

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал) КФУ



УТВЕРЖДАЮ

Директор
Елабужского института КФУ
 Е.Е. Мерзон.
" 8 " 06 2023 г.

Программа дисциплины (модуля)

Основы искусственного интеллекта и исследование операций

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2023

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. Анисимова Э.С. (Кафедра математики и прикладной информатики)

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.1	Знать принципы поиска информации, критического анализа и синтеза информации, методики системного подхода для решения поставленных задач
УК-1.2	Уметь осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.3	Владеть навыками поиска, критического анализа и синтеза информации; способностью применять системный подход для решения поставленных задач
ОПК-5	Способен осуществлять контроль и оценку формирования результатов образования обучающихся, выявлять и корректировать трудности в обучении
ОПК-5.2	Уметь применять технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

принципы поиска, критического анализа и синтеза информации, методики системного подхода для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций.

Должен уметь:

осуществлять поиск, критический анализ и синтез информации; применять системный подход для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций;

выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении основам искусственного интеллекта и исследования операций.

Должен владеть:

навыками поиска, критического анализа и синтеза информации; способностью применять системный подход для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.08.11 Дисциплины (модули)" основной профессиональной образовательной программы 44.03.05 "Педагогическое образование (с двумя профилями подготовки) (Математика и информатика)" и относится к обязательным дисциплинам. Осваивается на 4 курсе в 7 семестре.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 2 зачетных(ые) единиц(ы) на 72 часа(ов).

Контактная работа - 36 часа(ов), в том числе лекции - 16 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 20 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 36 часа(ов).

Контроль (зачёт / экзамен) - 0 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 7 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Линейное программирование	7	2	0	4	8
2.	Тема 2. Теория двойственности	7	2	0	2	6
3.	Тема 3. Элементы теории игр	7	2	0	2	8
4.	Тема 4. Введение в проектирование интеллектуальных систем	7	2	0	0	2
5.	Тема 5. Модели представления знаний в интеллектуальных системах	7	2	0	0	2
6.	Тема 6. Язык логического программирования Пролог	7	2	0	4	4
7.	Тема 7. Экспертные системы	7	2	0	4	2
8.	Тема 8. Нейронные сети	7	2	0	4	4
	Итого 72		16	0	20	36

4.2 Содержание дисциплины (модуля)

Тема 1. Линейное программирование

Общая постановка задачи оптимизации. Целевая функция. Допустимое множество. Допустимое решение. Оптимальное решение. Оптимальное множество. Постановка задачи математического программирования. Классификация задач математического программирования. Постановка задачи линейного программирования. Примеры задач линейного программирования (ЛП): задача о банке, задача о диете, задача об использовании ресурсов, транспортная задача. Общая постановка задачи ЛП и различные формы ее записи (числовая, матричная). Стандартная и каноническая формы задачи ЛП. Геометрия задачи ЛП. Выпуклая многогранная область в n R. Проектирование выпуклого многогранника на координатные плоскости. Теорема о проекциях. Теорема о существовании оптимального решения задачи ЛП в случае ограниченности целевой функции. Теорема о достижимости оптимального решения задачи ЛП в угловой точке (в случае ограниченности целевой функции). Строение множества оптимальных решений. Графический метод решения задач ЛП при малом числе неизвестных. Линия уровня целевой функции. Алгоритм решения задачи ЛП графическим методом. Сведение задач линейного программирования общего вида к задачам, допускающим решение графическим методом. Симплекс-метод решения задачи ЛП общего вида. Алгоритм решения задачи ЛП симплекс-методом. Геометрическая интерпретация симплекс-алгоритма. Теорема о конечности симплекс-алгоритма. Применение компьютерных программ для решения задач линейного программирования

Тема 2. Теория двойственности

Постановка взаимно двойственных задач ЛП. Симметричные взаимно двойственные задачи. Экономический смысл двойственности. Основное неравенство для двойственных задач. Основная теорема двойственности и ее следствия. Критерий оптимальности. Теорема равновесия. Условия дополняющей нежесткости. Двойственность в экономических задачах. Двойственные цены. Применение двойственности в однопродуктивной задаче. Транспортная задача ЛП. Открытая и закрытая модель транспортной задачи. Критерий разрешимости транспортной задачи. Методы построения начального опорного плана транспортной задачи (метод СЗ угла, метод минимального тарифа, метод Фогеля). Потенциалы, их экономический смысл. Решение транспортной задачи методом потенциалов. Двойственность в

транспортной задаче.

Тема 3. Элементы теории игр

Понятие об игровых моделях. Платежная матрица. Верхняя и нижняя цена игры. Седловая точка. Решение игр в смешанных стратегиях. Теорема Неймана. Матричная игра как задача линейного программирования. Принципы максимина и минимакса. Оптимальная стратегия и цена игры. Графическое решение игр вида $2 \times n$ и $m \times 2$. Решения игровых задач методами линейного программирования.

Тема 4. Введение в проектирование интеллектуальных систем

Мышление и интеллект, философские аспекты проблемы мышления. Интеллектуальные задачи. Искусственный интеллект. Основные направления исследований в области искусственного интеллекта. План имитации мышления Тьюринга. Японский проект ЭВМ пятого поколения. Этапы создания интеллектуальных компьютеров. Внутренняя и внешняя интеллектуализация.

Тема 5. Модели представления знаний в интеллектуальных системах

Признаки интеллектуальных информационных систем (ИИС), классификация ИИС. Данные и знания. Организация базы знаний. Предметное (фактуальное) и проблемное (операционное) знания. Декларативная и процедурная формы представления знаний. Модели представления знаний: продукционная модель, семантические сети, простые и сложные фреймы, формальные логические модели.

Тема 6. Язык логического программирования Пролог

Императивные и декларативные языки программирования. История возникновения и развития Пролога. Области использования, преимущества и недостатки языка Пролог. Логические основы Пролога. Основные понятия Пролога. Управление выполнением программ на Прологе. Операции над списками, множествами. Обработка строк и файлов. Работа с внутренними (динамическими) базами данных.

Тема 7. Экспертные системы

Понятие об экспертной системе (ЭС). Общая характеристика ЭС. Виды ЭС и типы решаемых задач. Структура и режимы использования ЭС. Этапы проектирования и разработки ЭС: идентификация, концептуализация, формализация, реализация, тестирование, опытная эксплуатация. Участники процесса проектирования: эксперты, инженеры по знаниям, конечные пользователи.

Тема 8. Нейронные сети

Нейрон и межнейронное взаимодействие. Модель технического нейрона. Искусственные нейронные сети и их архитектура. Многослойный перцептрон. Типичные задачи, решаемые с помощью нейронных сетей. Ограничения применения нейронных сетей. Обучение перцептрона. Алгоритм обучения перцептрона. Процедура обратного распространения. Обучающий алгоритм обратного распространения. Пример обучения. Область применения алгоритма и ограничения по использованию.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержденный приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года № 245)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- критерии оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;
- в печатном виде - в Научной библиотеке Елабужского института КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки Елабужского института КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Основы программирования на языке Пролог - <https://www.intuit.ru/studies/courses/44/44/info>

Основы теории нейронных сетей - <https://www.intuit.ru/studies/courses/88/88/info>

Проектирование систем искусственного интеллекта - <https://www.intuit.ru/studies/courses/1122/167/info>

Исследование операций Примеры и задачи - <http://math.nsc.ru/LBRT/k4/or/>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	Лекционные занятия проводятся с использованием интерактивных технологий и предполагают активное участие студентов. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них. Желательно выделять в используемой литературе постановки вопросов, на которые разными авторами могут быть даны различные ответы. На основании постановки таких вопросов следует собирать аргументы в пользу различных вариантов решения поставленных проблем.

Вид работ	Методические рекомендации
лабораторные работы	Лабораторные занятия - это одна из разновидностей практического занятия, являющаяся эффективной формой учебных занятий в организации высшего образования. Лабораторные занятия имеют выраженную специфику в зависимости от учебной дисциплины, углубляют и закрепляют теоретические знания. На этих занятиях студенты осваивают конкретные методы изучения дисциплины, обучаются экспериментальным способам анализа, умению работать с приборами и современным оборудованием. Лабораторные занятия дают наглядное представление об изучаемых явлениях и процессах, студенты осваивают постановку и ведение эксперимента, учатся умению наблюдать, оценивать полученные результаты, делать выводы и обобщения. Отчёт по итогам выполненных лабораторных работ выполняется на листах белой бумаги формата А4 в печатном или рукописном виде. При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху. При оформлении отчёта в печатном виде желательно соблюдать следующие требования. Для заголовков: полужирный шрифт, 14 пт, центрированный. Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине. Во всех случаях тип шрифта - Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал. Поля: левое - 3 см, правое - 1 см, верхнее и нижнее - 2 см. Отчет должен содержать следующие элементы: 1) Титульный лист с обязательным указанием варианта; 2) Цель работы; 3) Задание; 4) Основная часть; 5) Вывод.
самостоятельная работа	Самостоятельная работа студентов по дидактической сути представляет собой комплекс условий обучения, организуемых преподавателем и направленных на самоподготовку учащихся. Учебная деятельность протекает без непосредственного участия преподавателя и заключается в проработке лекционного материала, подготовке к лабораторным занятиям; изучении учебной литературы из основного и дополнительного списка.
зачет	Зачет является формой оценки качества освоения студентом образовательной программы по дисциплине. По результатам зачета студенту выставляется оценка "зачтено" или "не зачтено". Зачет может проводиться в форме устного опроса по билетам (вопросам) или без билетов, с предварительной подготовкой или без подготовки, по усмотрению кафедры. Преподаватель может проставить зачет без опроса или собеседования тем студентам, которые активно участвовали на лабораторных занятиях.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория №60 (423600, Республика Татарстан, г. Елабуга, ул. Казанская, д. 89) для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещение для самостоятельной работы. Комплект мебели (посадочных мест) 29 шт. Комплект мебели (посадочных мест) для преподавателя 1 шт. Компьютерный класс: Компьютеры intel core i5 15 шт. Мониторы ViewSonic 22d 15 шт. Проектор EPSON EB-535W 1 шт. Интерактивная доска IQBoard DVT TN082 1 шт. Трибуна 1 шт. Кондиционер 1 шт. Настенные полки 6 шт. Шкаф двухстворчатый с полками 1 шт. Веб-камера 1 шт. Выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Набор учебно-наглядных пособий: комплект презентаций в электронном формате по преподаваемой дисциплине

3-5

шт.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:
- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 44.03.05 "Педагогическое образование (с двумя профилями подготовки)" и профилю подготовки "Математика и информатика".

Приложение №1
к рабочей программе дисциплины (модуля)
Б1.О.08.11 Основы искусственного интеллекта и исследование операций

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал)

Фонд оценочных средств по дисциплине (модулю)
Б1.О.08.11 Основы искусственного интеллекта и исследование операций

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)
Профиль подготовки: Математика и информатика
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский
Год начала обучения по образовательной программе: 2022

СОДЕРЖАНИЕ

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)
2. Критерии оценивания сформированности компетенций
3. Распределение оценок за формы текущего контроля и промежуточную аттестацию
4. Оценочные средства, порядок их применения и критерии оценивания
 - 4.1. Оценочные средства текущего контроля
 - 4.1.1. Лабораторные работы
 - 4.1.1.1. Порядок проведения.
 - 4.1.1.2 Критерии оценивания
 - 4.1.1.3. Содержание оценочного средства
 - 4.1.2. Тестирование
 - 4.1.2.1. Порядок проведения.
 - 4.1.2.2 Критерии оценивания
 - 4.1.2.3. Содержание оценочного средства
 - 4.1.3. Эссе
 - 4.1.3.1. Порядок проведения.
 - 4.1.3.2 Критерии оценивания
 - 4.1.3.3. Содержание оценочного средства
 - 4.2. Оценочные средства промежуточной аттестации
 - 4.2.1. Устный или письменный ответ на вопрос
 - 4.2.1.1. Порядок проведения.
 - 4.2.1.2. Критерии оценивания.
 - 4.2.1.3. Оценочные средства.
 - 4.2.2. Решение задач
 - 4.2.2.1. Порядок проведения.
 - 4.2.2.2. Критерии оценивания.
 - 4.2.2.3. Оценочные средства.

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
УК-1 – Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Знать принципы поиска, критического анализа и синтеза информации, методики системного подхода для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций. Уметь осуществлять поиск, критический анализ и синтез информации; применять системный подход для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций; Владеть навыками поиска, критического анализа и синтеза информации; способностью применять системный подход для решения стандартных и нестандартных задач искусственного интеллекта и исследования операций.	Текущий контроль: Лабораторные работы по темам Тема 1. Линейное программирование Тема 2. Теория двойственности Тема 3. Элементы теории игр Тема 4. Введение в проектирование интеллектуальных систем Тема 5. Модели представления знаний в интеллектуальных системах Тема 6. Язык логического программирования Пролог Тема 7. Экспертные системы Тема 8. Нейронные сети Тестирование по темам Тема 1. Линейное программирование Тема 2. Теория двойственности Тема 3. Элементы теории игр Эссе по темам Тема 4. Введение в проектирование интеллектуальных систем Тема 5. Модели представления знаний в интеллектуальных системах Тема 6. Язык логического программирования Пролог Тема 7. Экспертные системы Тема 8. Нейронные сети Промежуточная аттестация: <i>Зачёт</i>
ОПК-5 – Способен осуществлять контроль и оценку формирования результатов образования обучающихся, выявлять и корректировать трудности в обучении	Уметь выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении основам искусственного интеллекта и исследованию операций.	Текущий контроль: Лабораторные работы по темам Тема 1. Линейное программирование Тема 2. Теория двойственности Тема 3. Элементы теории игр Тема 4. Введение в проектирование интеллектуальных систем Тема 5. Модели представления знаний в интеллектуальных системах Тема 6. Язык логического программирования Пролог Тема 7. Экспертные системы Тема 8. Нейронные сети Тестирование по темам Тема 1. Линейное программирование Тема 2. Теория двойственности Тема 3. Элементы теории игр Эссе по темам Тема 4. Введение в проектирование интеллектуальных систем Тема 5. Модели представления знаний в интеллектуальных системах

		Тема 6. Язык логического программирования Пролог Тема 7. Экспертные системы Тема 8. Нейронные сети Промежуточная аттестация: <i>Зачёт</i>
--	--	--

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	Средний уровень (хорошо) (71-85 баллов)	Низкий уровень (удовлетворительно) (56-70 баллов)	Ниже порогового уровня (неудовлетворительно) (0-55 баллов)
УК-1	Демонстрирует высокий уровень знаний методов исследования операций, необходимые для решения математических и финансово-экономических задач; основных понятий теории интеллектуальных информационных систем, особенности представления данных и знаний; этапов разработки и функционирования экспертных систем; методы разработки и создания экспертных систем и экспертных оболочек.	Знает основы методов исследования операций, необходимые для решения математических и финансово-экономических задач; основные понятия теории интеллектуальных информационных систем, особенности представления данных и знаний; этапы разработки и функционирования экспертных систем; методы разработки и создания экспертных систем и экспертных оболочек в базовом (стандартном) объеме	Демонстрирует частичные знания методов исследования операций, необходимые для решения математических и финансово-экономических задач; основных понятий теории интеллектуальных информационных систем, особенности представления данных и знаний; этапов разработки и функционирования экспертных систем; методов разработки и создания экспертных систем и экспертных оболочек, без грубых ошибок.	Демонстрирует частичные знания методов исследования операций, необходимые для решения математических и финансово-экономических задач; основных понятий теории интеллектуальных информационных систем, особенности представления данных и знаний; этапов разработки и функционирования экспертных систем; методов разработки и создания экспертных систем и экспертных оболочек, допуская грубые ошибки
	Демонстрирует высокий уровень умений применения методов исследования операций для решения экономических задач; использования прикладных экспертных систем для решения плохо формализуемых задач	Умеет применять методы исследования операций для решения экономических задач; использовать прикладные экспертные системы для решения плохо формализуемых задач в базовом (стандартном) объеме	Демонстрирует частичные умения применения методов исследования операций для решения экономических задач; использования прикладных экспертных систем для решения плохо формализуемых задач без грубых ошибок	Демонстрирует частичные умения применения методов исследования операций для решения экономических задач; использования прикладных экспертных систем для решения плохо формализуемых задач, допуская

				грубые ошибки
	Демонстрирует владение навыками применения современного математического инструментария для решения экономических задач; методикой построения, анализа и применения математических моделей для оценки состояния и прогноза развития экономических явлений и процессов (в части компетенций, соответствующих основным методам); навыками работы с инструментальными средствами в области интеллектуальных информационных систем; навыками логического проектирования баз данных предметной области; навыками логического (функционального) программирования на языке Пролог (Лисп) на высоком уровне	Владеет навыками применения современного математического инструментария для решения экономических задач; методикой построения, анализа и применения математических моделей для оценки состояния и прогноза развития экономических явлений и процессов (в части компетенций, соответствующих основным методам); навыками работы с инструментальными средствами в области интеллектуальных информационных систем; навыками логического проектирования баз данных предметной области; навыками логического (функционального) программирования на языке Пролог (Лисп) в базовом (стандартном) объеме	Демонстрирует частичные владения навыками применения современного математического инструментария для решения экономических задач; методикой построения, анализа и применения математических моделей для оценки состояния и прогноза развития экономических явлений и процессов (в части компетенций, соответствующих основным методам); навыками работы с инструментальными средствами в области интеллектуальных информационных систем; навыками логического проектирования баз данных предметной области; навыками логического (функционального) программирования на языке Пролог (Лисп)	Демонстрирует низкий уровень владения навыками применения современного математического инструментария для решения экономических задач; методикой построения, анализа и применения математических моделей для оценки состояния и прогноза развития экономических явлений и процессов (в части компетенций, соответствующих основным методам); навыками работы с инструментальными средствами в области интеллектуальных информационных систем; навыками логического проектирования баз данных предметной области; навыками логического (функционального) программирования на языке Пролог (Лисп)
ОПК-5	Демонстрирует высокий уровень умений применять методы исследования операций для решения задач; использовать нейронные сети; составлять программы на языке логического проектирования Пролог в базовом (стандартном) объеме	Умеет применять методы исследования операций для решения задач; использовать нейронные сети; составлять программы на языке логического проектирования Пролог в базовом (стандартном) объеме	Демонстрирует частичные умения применять методы исследования операций для решения задач; использовать нейронные сети; составлять программы на языке логического проектирования Пролог без грубых ошибок	Демонстрирует частичные умения применять методы исследования операций для решения задач; использовать нейронные сети; составлять программы на языке логического проектирования Пролог, допуская грубые ошибки

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

7 семестр:

Текущий контроль:

Лабораторные работы

Тема 1. Линейное программирование

Тема 2. Теория двойственности

Тема 3. Элементы теории игр

Тема 4. Введение в проектирование интеллектуальных систем

Тема 5. Модели представления знаний в интеллектуальных системах

Тема 6. Язык логического программирования Пролог

Тема 7. Экспертные системы

Тема 8. Нейронные сети

Максимальное количество баллов по БРС - 40.

Тестирование

Тема 1. Линейное программирование

Тема 2. Теория двойственности

Тема 3. Элементы теории игр

Максимальное количество баллов по БРС - 5.

Эссе

Тема 4. Введение в проектирование интеллектуальных систем

Тема 5. Модели представления знаний в интеллектуальных системах

Тема 6. Язык логического программирования Пролог

Тема 7. Экспертные системы

Тема 8. Нейронные сети

Максимальное количество баллов по БРС - 5.

Итого $40+5+5=50$ баллов

Промежуточная аттестация - зачет – 50 баллов.

Промежуточная аттестация проводится после завершения изучения дисциплины или ее части в форме, определяемой учебным планом образовательной программы с целью оценить работу обучающегося, степень усвоения теоретических знаний, уровень сформированности компетенций.

Преподаватель, принимающий зачет обеспечивает случайное распределение вариантов зачетных заданий между обучающимися с помощью билетов и/или с применением компьютерных технологий; вправе задавать обучающемуся дополнительные вопросы и давать дополнительные задания помимо тех, которые указаны в билете.

Зачет проводится по билетам. В каждом билете два оценочных средства: устный или письменный ответ на вопрос и решение задачи.

Устный или письменный ответ – 20 баллов.

Решение задачи – 30 баллов.

Итого $20+30=50$ баллов.

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: $50+50=100$ баллов.

Соответствие баллов и оценок:

Для зачета:

56-100 – зачтено

0-55 – не зачтено

4. Оценочные средства, порядок их применения и критерии оценивания

4.1. Оценочные средства текущего контроля

4.1.1. Лабораторные работы

Тема 1. Линейное программирование

Тема 2. Теория двойственности

Тема 3. Элементы теории игр

Тема 4. Введение в проектирование интеллектуальных систем
Тема 5. Модели представления знаний в интеллектуальных системах
Тема 6. Язык логического программирования Пролог
Тема 7. Экспертные системы
Тема 8. Нейронные сети

4.1.1.1. Порядок проведения.

Лабораторные работы проводятся в часы аудиторной работы.

Перед выполнением каждой работы студенты-бакалавры должны проработать соответствующий материал, используя конспекты теоретических занятий, периодические издания, учебно-методические пособия и учебники.

По окончании занятий студенты оформляют отчет по каждой работе, соблюдая следующую форму:

- Наименование темы;
- Цель работы;
- Задание и содержание выполненной работы,
- Письменные ответы на контрольные вопросы.
- Выводы по проделанной работе.
- Список использованных источников.

4.1.1.2 Критерии оценивания

35-40 баллов ставится, если обучающийся:

Правильно выполнил все задания. Продемонстрировал высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.

29-34 баллов ставится, если обучающийся:

Правильно выполнил большую часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий.

23-28 баллов ставится, если обучающийся:

Задания выполнил более чем наполовину. Присутствуют серьезные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.

0-22 баллов ставится, если обучающийся:

Задания выполнил менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.

4.1.1.3. Содержание оценочного средства

Темы 1-8

Лабораторная работа 1. Графическое решение задачи линейного программирования.

Цель работы: построение математической модели и решение оптимизационной задачи линейного программирования.

Требуется:

1. В соответствии с исходными данными записать математическую модель.
2. В соответствии с ограничениями построить область допустимых решений в пространстве параметров X_e, X_i .
3. В ОДР построить линии равных уровней целевой функции D и определить максимальное значение D , принадлежащее этой области.
4. Определить оптимальные значения суточных объемов производства X_e^* и X_i^* изделий E и I обеспечивающие максимум целевой функции.
5. Рассчитать значения целевой функции D в узлах ОДР и сравнить полученные значения целевой функции D и координат оптимального решения X_e^* и X_i^* с решениями, полученными графически.
6. Выяснить, какие ограничения мешают дальнейшему увеличению дохода и выдать рекомендации по ослаблению этих ограничений.

Исходные данные

Номер варианта	a_{11}	a_{12}	a_{21}	a_{22}	b_1	b_2	b_3	b_4	C_e	C_i
	1,2	2	2	1,2	5,9	7,9	0,8	2,1	4	3

Постановка задачи.

Нередко на практике возникают задачи оптимизации какого-либо процесса по выбранному критерию. В данной работе рассматривается задача построения математической модели и её оптимизация графическим методом линейного программирования.

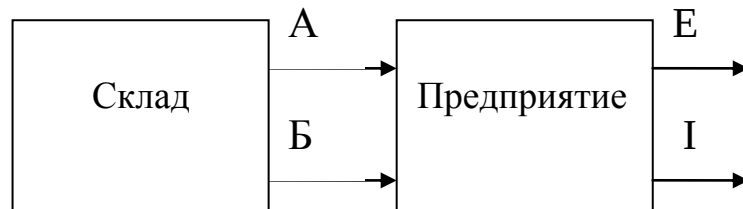


Рис.1

Небольшое предприятие (рис.1) выпускает два вида изделий Е и I. Продукция обоих видов поступает в продажу. Для производства требуется два исходных продукта – А и В. Максимально возможные суточные запасы этих продуктов на складе предприятия составляют b_1 и b_2 соответственно.

Расходы продуктов А и В на единицу соответствующего изделия приведены в табл.1.

Таблица 1

Исходный продукт	Расход исходного продукта на единицу изделия		Максимально возможный запас
	Е	I	
А	a_{11}	a_{12}	b_1
В	a_{21}	a_{22}	b_2

Цена одной единицы изделия Е равна C_e и C_i для I.

Изучение рынка сбыта показало, что суточный спрос на изделие I никогда не превышает спроса на изделие Е более чем на b_3 . Кроме того установлено, что спрос на изделие I никогда не превышает b_4 .

Каков должен быть объем производства каждого вида изделия в сутки, чтобы доход от реализации был максимальным?

Лабораторная работа 2. Симплексный метод решения задач линейного программирования.

У задачи линейного программирования в канонической форме все ограничения (кроме требования неотрицательности переменных) имеют вид равенств, т.е.

$$\begin{aligned} f(X) &= f(x_1, x_2, \dots, x_n) = \\ &= c_1 x_1 + c_2 x_2 + \dots + c_n x_n \rightarrow \max(\min) \end{aligned}$$
$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\ x_i \geq 0, \quad i = 1, 2, \dots, n \end{array} \right.$$
$$\text{или } \left\{ \begin{array}{l} AX = B \\ X \geq 0 \end{array} \right. , (1)$$
$$f(X) = CX \rightarrow \max(\min)$$

где $X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$ - вектор переменных, $B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$ - столбец свободных членов,

$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$ - матрица системы ограничений.

Задачей линейного программирования в симметричной форме записи называется задача вида (в матричной форме записи):

$$\begin{aligned} f(X) = CX \rightarrow \min & & f(X) = CX \rightarrow \max \\ \begin{cases} AX \geq B \\ X \geq 0 \end{cases} & \text{или} & \begin{cases} AX \leq B \\ X \geq 0 \end{cases} \end{aligned} \quad (2)$$

Симплекс-метод или метод улучшенного плана – один из универсальных методов решения задач линейного программирования. Это упорядоченный процесс перехода от одного опорного плана к другому, при котором (при решении задач на максимум) соответствующие значения целевой функции возрастают (или, по крайней мере, не убывают).

Пусть задача линейного программирования имеет канонический вид

$$\begin{aligned} f(X) = CX \rightarrow \max, \\ \begin{cases} AX = B \\ X \geq 0 \end{cases}, \end{aligned} \quad (3)$$

причем столбец свободных членов удовлетворяет условию $B \geq 0$. В системе ограничений m уравнений и n неизвестных, т.е. матрица A имеет размер $m \times n$, вектор-столбец $B - m \times 1$, вектор-строка коэффициентов целевой функции $C - 1 \times n$. Алгоритм решения задачи (3.1) симплекс-методом будем сопровождать решением конкретного примера, а именно задачи

$$\begin{aligned} f(X) = 5x_1 - x_2 + 3x_3 - 2x_4 \rightarrow \max, \\ \begin{cases} 3x_1 + 2x_2 + x_3 + x_4 = 7 \\ 5x_1 + 3x_2 + x_3 + 2x_4 = 11 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases} \end{aligned} \quad (4)$$

- 1) Получение начального опорного плана. Один из вариантов – преобразование расширенной матрицы системы ограничений к приведенному виду, выделение базисных и свободных переменных:

$$\left(\begin{array}{cccc|c} 3 & 2 & 1 & 1 & 7 \\ 5 & 3 & 1 & 2 & 11 \end{array} \right) \xrightarrow{C_2 = C_2 - C_1} \left(\begin{array}{cccc|c} 3 & 2 & 1 & 1 & 7 \\ 2 & 1 & 0 & 1 & 4 \end{array} \right) \xrightarrow{C_1 = C_1 - C_2} \left(\begin{array}{cccc|c} 1 & 1 & 1 & 0 & 3 \\ 2 & 1 & 0 & 1 & 4 \end{array} \right).$$

Здесь x_3, x_4 – базисные переменные, x_1, x_2 – свободные переменные. При преобразованиях необходимо следить за тем, чтобы столбец свободных членов оставался неотрицательным. Начальный опорный план получается, если присвоить свободным переменным значения, равные нулю; при этом базисные

переменные принимают значения, равные числам в соответствующей строке столбца свободных членов: $X_{оп1}=(0;0;3;4)$.

2) По преобразованной системе ограничений составляют симплекс-таблицу. В верхней строке (заголовки столбцов) располагаются свободные переменные, в крайнем левом столбце – базисные переменные; крайний правый столбец – это столбец свободных членов, а самая нижняя строка является строкой целевой функции (об определении чисел Δ_1 , Δ_2 , Δ_f в этой строке речь пойдет ниже). Остальное содержимое таблицы – столбцы преобразованной матрицы, отвечающие соответствующим столбцам свободных переменных (см. таблицу 1).

	x_1	x_2	B
x_3	1	1	3
x_4	2	1	4
f	Δ_1	Δ_2	Δ_f

Таблица 1.

	x_1	x_2	B
x_3	1	1	3
x_4	2	1	4
f	-6	2	1

Таблица 2

Для того чтобы найти значение Δ_i (в рассматриваемом примере $i=1,2$), воспользуемся правилом: вектор из коэффициентов при базисных переменных в целевой функции скалярно умножить на i -й столбец симплекс-таблицы и вычесть из найденного числа коэффициент целевой функции при соответствующем свободном переменном. Для Δ_f скалярно перемножаются вектор коэффициентов при базисных переменных целевой функции и столбец свободных членов:

$$\Delta_1 = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} - 5 = 3 \cdot 1 + (-2) \cdot 2 - 5 = -6;$$

$$\Delta_2 = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} - (-1) = 3 \cdot 1 + (-2) \cdot 1 + 1 = 2;$$

$$\Delta_f = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = 3 \cdot 3 + (-2) \cdot 4 = 1.$$

Итак, таблица 2 представляет собой окончательный вид первой симплекс-таблицы.

Замечание. Следует обратить внимание на то, что Δ_f – это значение целевой функции при найденном начальном опорном решении: $\Delta_f = f(0,0,3,4) = 1$. Числа же Δ_1 , Δ_2 – оценки, которые будут учитываться при проверке этого решения на оптимальность.

3) Построенное начальное опорное решение является оптимальным, если все оценки Δ_i неотрицательны (причем в случае положительности всех оценок решение единственно!). Если среди этих оценок есть отрицательная, но среди чисел в ее столбце нет положительных, то исходная задача не имеет решения в силу неограниченности целевой функции. Наконец, если в каждом столбце с отрицательной оценкой есть хотя бы один положительный элемент, то необходимо осуществить переход к новому опорному плану (который затем снова проверяется на оптимальность).

4) Переход к новому опорному плану проводится по следующей схеме.

- Выбирается ведущий столбец (столбец с отрицательной оценкой). Если отрицательных оценок несколько, то выбирается столбец с отрицательной оценкой, наибольшей по модулю. В рассматриваемом примере ведущим будет первый столбец ($\Delta_1 = -6$)

- Выбирается ведущая строка. Для этого определяется наименьшее из симплексных отношений (т.е. отношений свободных членов к соответствующим положительным элементам ведущего столбца). В

примере оба числа в первом столбце положительны, поэтому: $\min \left\{ \frac{3}{1}, \frac{4}{2} \right\} = 2$, ведущей будет вторая

строка.

- На пересечении ведущих строки и столбца определяется ведущий (разрешающий) элемент (в примере это 2, соответствующая ячейка таблицы 3.2 заштрихована).

- В заголовках меняются местами переменные, соответствующие ведущим строке и столбцу (в примере меняются места x_1 и x_4).

- Ведущий элемент заменяется значением, обратным этому элементу (в примере $\frac{1}{2}$).

- Все остальные элементы ведущей строки делятся на ведущий элемент, а все остальные элементы ведущего столбца делятся на ведущий элемент, взятый со знаком «-» (см. таблицу 3, в которую внесены описанные выше изменения, а не найденные пока числа заменены греческими буквами).

- Оставшиеся элементы находят с помощью «правила многоугольника» $X = X' - \frac{AC}{B}$ (здесь X – вычисляемое значение, X' – соответствующий элемент «старой» таблицы, B – ведущий элемент, A и C – оставшиеся вершины четырехугольника с диагональю $X'B$). Для разбираемого примера имеем:

$$\alpha = 1 - \frac{1 \cdot 1}{2} = \frac{1}{2}; \beta = 3 - \frac{4 \cdot 1}{2} = 1; \gamma = 2 - \frac{1 \cdot (-6)}{2} = 5; \delta = 1 - \frac{4 \cdot (-6)}{2} = 13.$$

Итак, получена новая симплекс-таблица (таблица 4), которая определяет новое опорное решение (свободные переменные x_2, x_4 ; их значения будут равны нулю; базисные переменные x_1, x_3 ; их значения – соответствующие числа из столбца свободных членов: $x_1=2, x_3=1$). Значение функции на этом опорном решении – в правом нижнем углу, т.е. $X_{оп2}=(2;0;1;0), f(2,0,1,0)=13$.

	x_4	x_2	B
x_3	-1/2	α	β
x_1	1/2	1/2	2
f	3	γ	δ

Таблица 3

	x_4	x_2	B
x_3	-1/2	1/2	1
x_1	1/2	1/2	2
f	3	5	13

Таблица 4

5) Построенное новое опорное решение требуется снова проверить на оптимальность и, если необходимо, повторить операцию перехода. В рассматриваемой задаче, однако, все оценки стали положительными, и, следовательно, $X_{оп2}=(2;0;1;0)=X_{оптим}$, $f_{max}=f(2,0,1,0)=13$.

Замечание 1. Задачи, в которых решается задача на минимум, легко свести к рассмотренному случаю. Для этого, сохранив систему ограничений, исследуем задачу с целевой функцией

$$f_1(X) = -f(X) \rightarrow \max.$$

Замечание 2. Бывают ситуации (например, когда система ограничений несовместна), когда начальное опорное решение построить невозможно; в этом случае исходная задача не имеет решения.

Решение ЗЛП в Excel

Для решения задач оптимизации в MS Excel используют надстройку **Поиск решения**, которая вызывается из пункта главного меню «Сервис» (рис. 1).

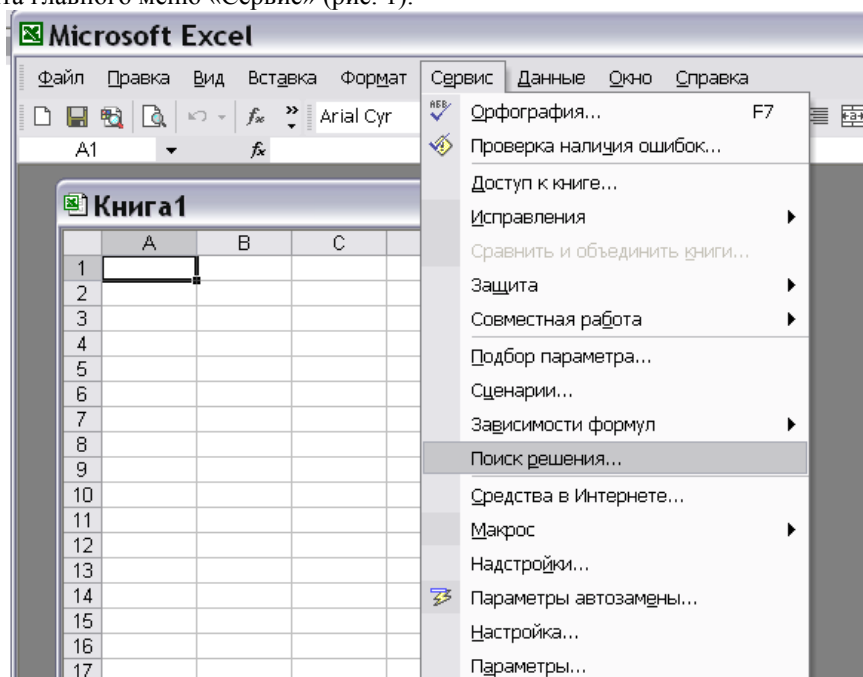


Рис. 1.

Если в версии Excel, установленной на Вашем компьютере, отсутствует данный подпункт меню «Сервис», необходимо вызвать пункт меню «Надстройки» и в предложенном списке дополнительных модулей выбрать «Поиск решения» (рис. 2).

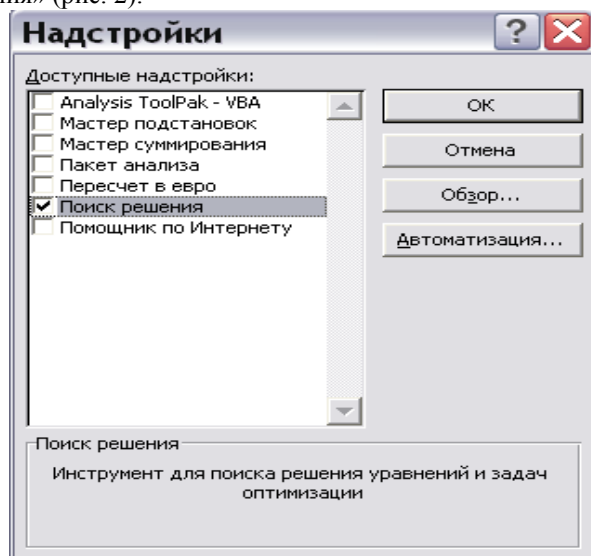


Рис. 2.

Рассмотрим на примере использование данной надстройки задачу.

$$\max_x f(x) = \max_x \{3x_1 + 2x_2\}$$

$$\begin{cases} x_1 + 2x_2 \leq 6; \\ 2x_1 + x_2 \leq 8; \\ x_1 + 0.8x_2 \leq 5; \\ -x_1 + x_2 \leq 1; \\ x_2 \leq 2; \\ x_1 \geq 0; \\ x_2 \geq 0. \end{cases}$$

Составим шаблон в редакторе Excel, как показано на рис. 3.

	A	B	C	D	E	F
1						
2		Переменные			Целевая функция	
3		Продукция П1	Продукция П2			
4	Значение					
5	Коэффициент в ЦФ					
6						
7		Ограничения				
8				Левая часть	Знак	Правая часть
9	Ресурс А					
10	Ресурс В					
11	Ресурс С					
12	Ограничение по спросу 1					
13	Ограничение по спросу 2					

Рис. 3. Шаблон оформления задачи.

Теперь занесём данную в задаче числовую информацию (рис.4).

Переменные			Целевая функция		
	Продукция П1	Продукция П2			
Значение					
Коэффициент в ЦФ	3	2			
Ограничения					
			Левая часть	Знак	Правая часть
Ресурс А	1	2		<=	6
Ресурс В	2	1		<=	8
Ресурс С	1	0.8		<=	5
Ограничение по спросу 1	-1	1		<=	1
Ограничение по спросу 2	0	1		<=	2

Рис.4. Исходные данные задачи

В выделенные пустые ячейки (значения целевой функции и левых частей неравенств) необходимо занести формулы, отображающие связи и отношения между числами на рабочем листе.

Ячейки В4 – С4 называются в Excel *изменяемыми* (в нашей модели это неизвестные переменные), т.е., изменяя их **Поиск решения** будет находить оптимальное значение целевой функции. Значения, которые первоначально вводят в эти ячейки, обычно нули (незаполненные клетки трактуются по умолчанию как содержащие нулевые значения).

Теперь необходимо ввести формулы. В нашей математической модели, целевая функция представляет собой произведение вектора коэффициентов на вектор неизвестных. Действительно, выражение $3x_1 + 2x_2$ можно рассматривать как произведение вектора (3,2) на вектор $3x_1 + 2x_2$.

В Excel существует функция СУММПРОИЗВ, которая позволяет найти скалярное произведение векторов. В ячейку Е4 необходимо вызвать данную функцию, а в качестве перемножаемых векторов задать адреса ячеек, содержащих коэффициенты уравнений (в данном случае, это В5:С5) и ячеек, в которые в результате решения будут помещены значения $(x_1, x_2)^T$ (ячейки В4:С4) (рис. 5).

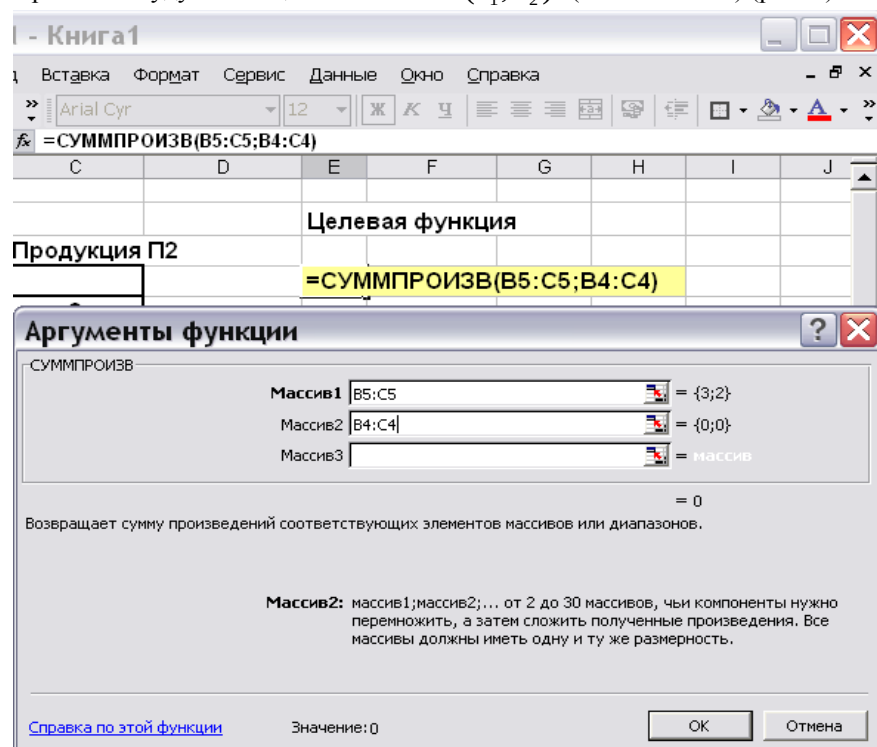


Рис. 5. Вызов функции СУММПРОИЗВ.

Каждая левая часть ограничения тоже представляет собой произведение двух векторов: соответствующей строки матрицы затрат и вектора неизвестных. То есть, выражение $x_1 + 2x_2$ (для первого ограничения $x_1 + 2x_2 \leq 6$) будем рассматривать как произведение вектора коэффициентов (1,2) и вектора пока переменных $(x_1, x_2)^T$.

В ячейке, отведенной для формулы левой части первого ограничения (D9), вызовем функцию СУММПРОИЗВ. В качестве адресов перемножаемых векторов занесем адрес строки коэффициентов B9:C9 и адрес значений переменных B4:C4 (рис.6).

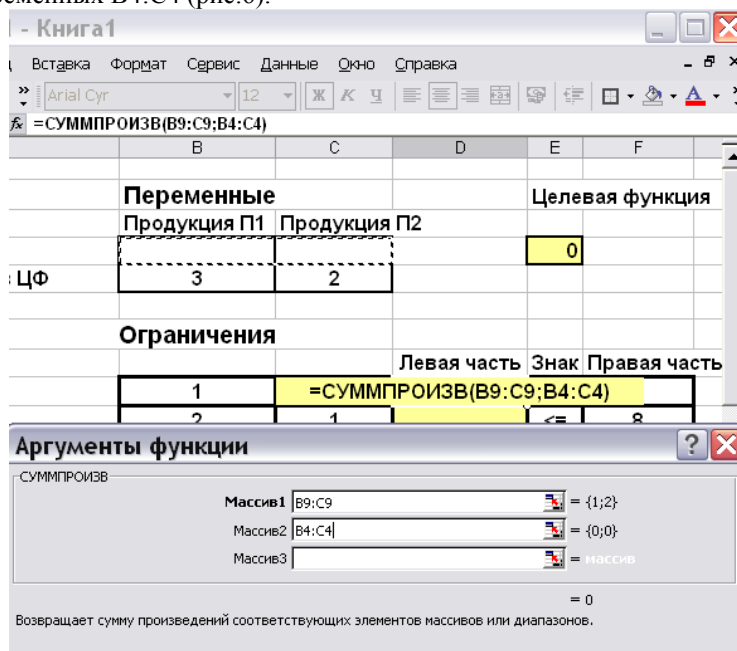


Рис. 6

В четыре оставшиеся ячейки графы «Левая часть» вводим аналогичные формулы, используя соответствующую строку матрицы затрат. Фрагмент экрана с введенными формулами показан на рис.7.

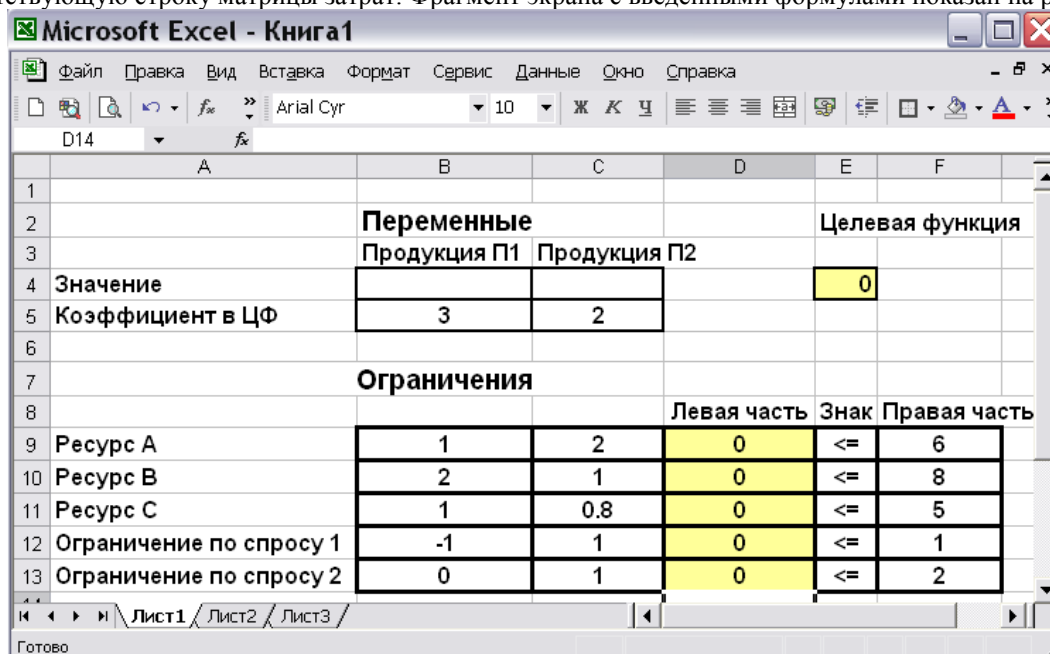


Рис. 7

Важно! К моменту вызова сервиса «Поиск решения» на рабочем листе с задачей должны быть занесены **формулы для левых частей ограничений и формула для значения целевой функции**.

В меню **Сервис** выбираем **Поиск решения**. В появившемся окне задаём следующую информацию:

1. в качестве целевой ячейки устанавливаем адрес ячейки для значения целевой функции E4;
2. «флажок» устанавливаем на вариант «максимальному значению», т.к. в данном случае, целевая функция дохода подлежит максимизации;
3. в качестве изменяемых ячеек заносится адрес строки значений переменных B4:C4;
4. справа от окна, предназначенного для занесения ограничений, нажимаем кнопку «Добавить», появится форма для занесения ограничения (рис. 8);

Рис.8. Форма для занесения одного ограничения ЗЛП.

5. в левой части формы «Ссылка на ячейку» заносится адрес формулы для левой части первого ограничения D9, выбирается требуемый знак неравенства (в нашем случае, <=), в поле «Ограничение» заносится ссылка на правую часть ограничения F9 (рис. 9).

Рис.9. Занесение первого ограничения задачи.

Аналогично заносятся все ограничения задачи, после чего нажимается кнопка «ОК».

Таким образом, окно «Поиск решения» с занесенной информацией выглядит следующим образом (рис.10):

Рис. 10.

Далее необходимо нажать кнопку **Параметры**, установить «флажки» **«Линейная модель»** и **«Неотрицательные значения»**, поскольку в данном случае задача является ЗЛП, а ограничение б) требует неотрицательности значений (рис.11).

Рис. 11. Установка параметров

Затем следует нажать «ОК», «Выполнить», после чего появляется окно результата решения (рис.12).

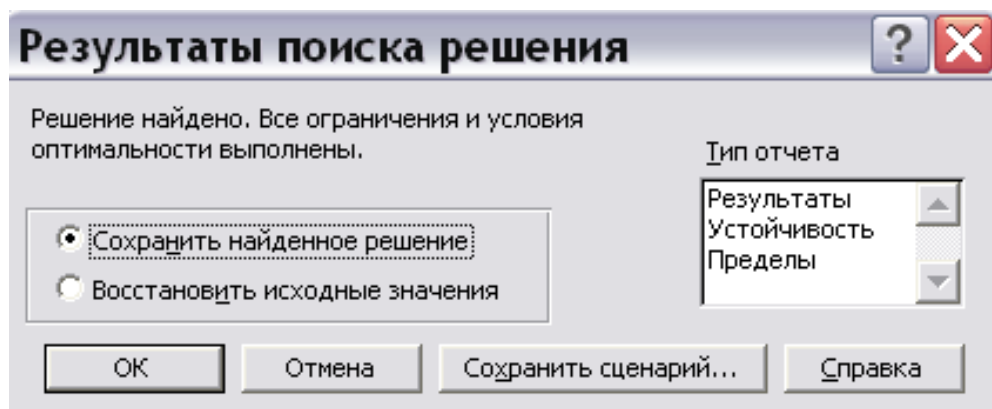


Рис. 12. Окно результата решения

Если в результате всех действий получено окно с сообщением «Решение найдено», то Вам предоставляется возможность получения трех типов отчета, которые полезны при анализе модели на чувствительность. В данном примере достаточно сохранить найденное решение, нажав «ОК». В результате получено решение задачи из примера 1. (рис.13).

Переменные		Целевая функция	
	Продукция П1	Продукция П2	
Значение	3,333333333	1,333333333	12,7
Коэффициент в ЦФ	3	2	
Ограничения		Левая часть	Знак Правая часть
Ресурс А	1	2	6 <= 6
Ресурс В	2	1	8 <= 8
Ресурс С	1	0,8	3,333333333 <= 5
Ограничение по спросу 1	-1	1	-2 <= 1
Ограничение по спросу 2	0	1	1,333333333 <= 2

Рис.13. Результат применения «Поиска решения»

Если в результате решения задачи выдано окно с сообщением о невозможности нахождения решения (рис.14), это означает, что при оформлении задачи была допущена ошибка (не заполнены формулы для ограничений, неправильно установлен «флажок» максимизации/минимизации и т.д.).

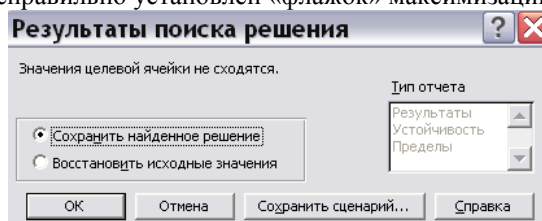


Рис.14. Сообщение об ошибке

В данном разделе рассмотрен общий формат решения задач оптимизации в Excel. В зависимости от математических моделей, выполняют его соответствующие модификации.

Например, можно установить условие на целочисленность некоторых переменных.

Задания для самостоятельной работы.

Решить задачи линейного программирования . в Excel.

$$f(X) = 11x_1 + 5x_2 - 8x_3 - 2x_4 \rightarrow \max, \quad f(X) = x_1 - 2x_2 + 3x_3 \rightarrow \max,$$

$$1) \begin{cases} x_1 + x_2 - x_3 = 4 \\ -2x_1 + 5x_3 + x_4 = 10 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 2) \begin{cases} 4x_1 + x_2 - 2x_3 \leq 3 \\ 2x_1 + x_2 + x_3 = 2 \\ 3x_1 + x_2 + 2x_3 \geq 3 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$f(X) = -7x_1 - 3x_2 - 3x_3 - 2x_4 \rightarrow \max, \quad f(X) = 2x_1 + 3x_2 + x_3 \rightarrow \max,$$

$$3) \begin{cases} -10x_1 - x_2 + 2x_3 + 3x_4 = -2 \\ 6x_1 + 2x_2 + 3x_3 + x_4 = 18 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 4) \begin{cases} x_1 + 3x_2 + 5x_3 \leq 15 \\ x_1 + x_2 + x_3 \leq 7 \\ 2x_1 + x_2 + 4x_3 \leq 12 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$f(X) = -3x_1 - 5x_2 + x_3 + x_4 \rightarrow \min, \quad f(X) = x_1 + 5x_2 + x_3 - x_4 \rightarrow \max,$$

$$5) \begin{cases} -2x_1 + 3x_2 + x_3 = 6 \\ -x_1 + 3x_2 - x_4 = -3 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 6) \begin{cases} x_1 + 2x_2 + x_3 = 3 \\ 2x_1 + x_2 + x_4 = 4 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$f(X) = x_1 - x_2 + 3x_3 - x_4 \rightarrow \max, \quad f(X) = 12x_1 + 8x_2 + 5x_3 + 4x_4 \rightarrow \min,$$

$$7) \begin{cases} -x_1 + 2x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + x_4 = 6 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 8) \begin{cases} -6x_1 + x_2 - x_3 + 2x_4 = -2 \\ 11x_1 - x_2 + 2x_3 - 3x_4 = 7 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$f(X) = -2x_1 - 3x_2 - 6x_3 + 18x_4 \rightarrow \max, \quad f(X) = x_1 + 2x_2 + x_3 \rightarrow \max,$$

$$9) \begin{cases} 4x_1 - 6x_2 + x_3 - 2x_4 = 8 \\ 4x_1 - 14x_2 + 2x_3 - 5x_4 = 12 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 10) \begin{cases} -2x_1 + x_2 + x_3 \leq 2 \\ -x_1 + x_2 + 3x_3 \leq 3 \\ x_1 - 3x_2 + x_3 \leq 1 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$f(X) = x_1 + x_2 + 3x_3 + x_4 \rightarrow \min, \quad f(X) = -4x_1 - 2x_2 + x_3 \rightarrow \min,$$

$$11) \begin{cases} 5x_1 - 6x_2 + x_3 - 2x_4 = 2 \\ 11x_1 - 14x_2 + 2x_3 - 5x_4 = 2 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad 12) \begin{cases} 3x_1 - 2x_2 + 4x_3 \leq 6 \\ 2x_1 + x_2 + 3x_3 = 18 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

Лабораторная работа 3. Решение задач линейного целочисленного программирования.

3.1 Методы решения целочисленных задач

Значительная часть задач по смыслу может иметь решения только в целых числах; например, число турбин, судов, животных может быть только целым числом. Такие задачи решаются методами целочисленного программирования. Общая постановка задачи линейного программирования дополняется требованием о том, чтобы найденные переменные в оптимальном плане были целыми.

Методы целочисленной оптимизации можно разделить на три основные группы: а) методы отсечения; б) комбинаторные методы; в) приближённые методы.

Методы отсечения используют оптимальные решения, найденные для задач линейного программирования. Сужая область допустимых планов до целочисленных границ, т. е. отсекая нецелочисленные допустимые планы, методами отсечения получают решения задач целочисленного программирования. Комбинаторные методы достигают решений задач целочисленного программирования, рассматривая возможные варианты целочисленных ограничений для задачи оптимизации. Приближённые методы опираются на приближённые методы нахождения экстремумов функций нескольких переменных и используют различные способы округления полученных нецелочисленных решений до целых значений. Особенно удобно применять приближённые методы в случае решения задачи целочисленного программирования относительно двух переменных.

3.2. Метод Гомори

Метод Гомори решения задач целочисленного программирования является методом отсечения. Сущность его состоит в том, что сначала задача решается как задача линейного программирования без учета условия целочисленности переменных. Если полученное решение задачи линейного программирования является целочисленным, задача целочисленного программирования также решена и найденное решение является оптимальным и для неё. Если же в найденном решении задачи линейного программирования одна или большее число переменных не целые, то для отыскания целочисленного решения задачи добавляется новое ограничение. Это ограничение линейное, и при продолжении решения дополненной задачи симплексным методом с учетом этого ограничения получается целочисленный план.

3.3. Метод ветвей и границ

Метод ветвей и границ – один из комбинаторных методов. Его суть заключается в упорядоченном переборе вариантов и рассмотрении лишь тех из них, которые оказываются по определённым признакам полезными для нахождения оптимального решения.

Метод ветвей и границ состоит в следующем: множество допустимых нецелочисленных решений (планов) некоторым способом разбивается на подмножества, для каждого из которых решается новая задача линейного программирования с целью получения целочисленного решения. При каждом ветвлении получается две новые задачи.

Очевидно, что возможен один из следующих четырёх случаев.

1. Одна из задач неразрешима, а другая имеет целочисленный оптимальный план. Тогда этот план и значение целевой функции на нём и дают решение исходной задачи.
2. Одна из задач неразрешима, а другая имеет оптимальный план, среди компонент которого есть дробные числа. Тогда рассматриваем вторую задачу и в её оптимальном плане выбираем одну из компонент, значение которой равно дробному числу, и строим две задачи на новых ограничениях по этой переменной, полученных делением её ближайших к решению целочисленных значений.
3. Обе задачи разрешимы. Одна из задач имеет оптимальный целочисленный план, а в оптимальном плане другой задачи есть дробные числа. Тогда вычисляем значения целевой функции на этих планах и сравниваем их между собой. Для определённости здесь и далее полагаем, что решается задача о максимуме целевой функции. Если на целочисленном оптимальном плане значение целевой функции больше или равно ее значению на плане, среди компонент которого есть дробные числа, то данный целочисленный план является оптимальным для исходной задачи и вместе со значением целевой функции на нём дает искомое решение. Если же значение целевой функции больше на плане, среди компонент которого есть дробные числа, то следует взять одно из таких чисел и для задачи, план которой рассматривается, произвести ветвление по дробной переменной и построить две новые задачи.
4. Обе задачи разрешимы, и среди оптимальных планов обеих задач есть дробные числа. Тогда вычисляем значение целевой функции на данных оптимальных планах и рассматриваем ту из задач, для которой значение целевой функции является наибольшим. В оптимальном плане этой задачи выбираем одну из компонент, значение которой является дробным числом, и производим ветвление на две новые задачи, разбивая область изменения этой переменной на две, ограниченные целыми числами справа и слева соответственно.

Такая последовательность действий при нахождении оптимального решения задачи целочисленного программирования нашла своё отражение в названии этого метода.

3.4 Решить целочисленные задачи в Excel

$$1) f(X) = x_1 - x_2 + x_3 + x_4 \rightarrow \max$$

$$\begin{cases} 5x_1 - 3x_2 + x_3 - 2x_4 = 1 \\ 9x_1 - 4x_2 + 2x_3 - 3x_4 = 6 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$2) f(X) = 5x_1 + x_2 - 7x_3 \rightarrow \max$$

$$\begin{cases} x_1 + x_2 - 2x_3 \leq 3 \\ 2x_1 + x_2 - 3x_3 \leq 7 \\ 3x_1 - 7x_3 \leq 10 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$1) f(X) = 2x_1 + x_2 - x_3 \rightarrow \min$$

$$\begin{cases} 2x_1 + x_2 - x_3 \geq 5 \\ x_1 + 2x_2 + x_3 \leq 7 \\ x_1 - x_2 + x_3 \geq 1 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$2) f(X) = 3x_1 + 2x_2 + x_3 \rightarrow \max$$

$$\begin{cases} x_1 + x_2 - 2x_3 \geq 2 \\ -3x_1 + 2x_2 + x_3 \leq 4 \\ x_1 + 2x_3 = 2 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$3) f(X) = 3x_1 + 2x_2 + x_4 \rightarrow \max$$

$$\begin{cases} x_1 + x_2 - 2x_3 + x_4 = 2 \\ -x_1 + x_3 + x_4 = 1 \\ -3x_1 + x_2 - 2x_3 + 4x_4 = 2 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$4) f(X) = x_1 + 4x_2 + x_3 \rightarrow \max$$

$$\begin{cases} -x_1 + 2x_2 + x_3 \leq 4 \\ 3x_1 + x_2 + 2x_3 \leq 9 \\ 2x_1 + 3x_2 + x_3 \geq 6 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$5) f(X) = 2x_1 + x_2 + 2x_3 \rightarrow \min$$

$$\begin{cases} x_1 + 2x_2 - x_3 \geq 2 \\ -2x_1 + x_2 + 2x_3 = 2 \\ 2x_1 + x_2 - 2x_3 \leq 6 \\ x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$6) f(X) = 2x_1 - x_2 + x_3 + x_4 \rightarrow \min$$

$$\begin{cases} 5x_1 - 3x_2 - x_3 - 2x_4 = 1 \\ -x_1 + 4x_2 + 2x_3 + x_4 = 6 \\ x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

Лабораторная работа 4. Двойственность в линейном программировании.

1. Составить двойственную задачу на основе исходной задач (см. лабораторную работу № 3).
2. Решить двойственную задачу симплекс-методом.
3. Решить двойственную задачу на компьютере.
4. Сравнить оптимальные решения, полученные на компьютере и симплекс - методом.
5. Сделать выводы.

Лабораторная работа 5. Классическая транспортная задача.

При формировании заказов автотранспортное предприятие учитывает наличие грузоотправителей и грузополучателей, их потребности по объемам перевозок, взаимное расположение для максимизации дохода АТП. Особое значение задача оптимизации доставки с учетом минимума транспортных расходов приобретает при определении наиболее выгодного пути доставки груза при консультировании грузоотправителей и грузополучателей по способу доставки груза, так как АТП может оказывать услуги по логистике клиентам. Причем задачу оптимизации осложняет и то, что спрос и предложение оказываются несбалансированными по объему. Отсюда возникает вопрос, как распределить товар от грузоотправителей до грузополучателей с минимумом транспортных расходов.

Математическая постановка задачи

В исследовании операций под *транспортной задачей* обычно понимают задачу выбора плана

перевозок некоторого товара (изделий, груза) от m **источников** (пунктов производства, поставщиков) к n **стокам** (станциям назначения, пунктам сбыта), обеспечивающего минимальные транспортные затраты. При этом предполагают, что: а) **мощность i -го источника** (объем поставок товара от i -го источника) равна $S_i > 0$, $i=1, \dots, m$; б) **мощность j -го стока** (объем поставок товара к j -му стоку) равна $D_j > 0$, $j=1, \dots, n$; в) стоимость перевозки единицы товара (в условных денежных единицах) от i -го источника к j -му стоку равна c_{ij} ; г) суммарная мощность всех источников равна суммарной мощности всех стоков, т.е.

$$\sum_{i=1}^m S_i = \sum_{j=1}^n D_j.$$

Далее под объемом товара будем понимать его количество в фиксированных единицах измерения.

Для математического описания транспортной задачи вводят переменные x_{ij} , обозначающие объемы поставок товара от i -го источника к j -му стоку. В этом случае $x_{i1} + x_{i2} + \dots + x_{in}$ — общий объем поставок товара от i -го источника, т.е. мощность этого источника; $x_{1j} + x_{2j} + \dots + x_{mj}$ — общий объем поставок товара к j -му стоку, т.е. мощность этого стока; $c_{11}x_{11} + c_{12}x_{12} + \dots + c_{mn}x_{mn}$ — суммарная стоимость перевозок товара от источников к стокам. С учетом этого рассматриваемая задача может быть представлена в следующем виде:

$$\left\{ \begin{array}{l} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min; \\ \sum_{j=1}^n x_{ij} = S_i, \\ \sum_{i=1}^m x_{ij} = D_j, \\ x_{ij} \geq 0, \\ x_{ij} \in N \cup \{0\}. \end{array} \right.$$

На рис. 1 показано представление транспортной задачи в виде сети с m пунктами отправления и n пунктами назначения, которые показаны в виде **узлов** сети. **Дуги**, соединяющие узлы сети, соответствуют маршрутам, связывающим пункты отправления и назначения. С дугой (i, j) , соединяющей пункт отправления i с пунктом назначения j , соотносятся два вида данных: стоимость c_{ij} перевозки единицы груза из пункта i в пункт j и количество перевозимого груза x_{ij} . Объем грузов в пункте отправления i равен S_i , а объем грузов в пункте назначения j равен D_j . Задача состоит в определении неизвестных величин x_{ij} , минимизирующих суммарные транспортные расходы и удовлетворяющих ограничениям, накладываемым на объемы грузов в пунктах отправления (предложение) и пунктах назначения (спрос).

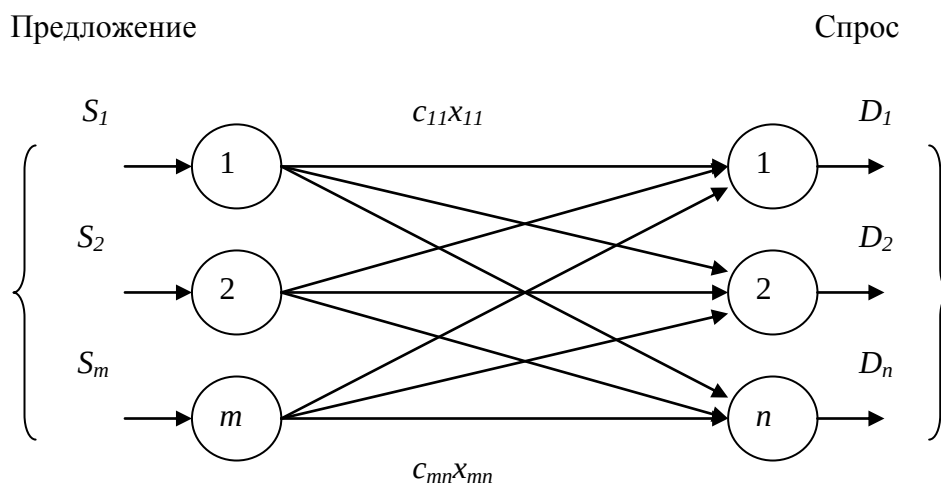


Рис. 1. Схема грузопотоков (грузоотправителей и грузополучателей)

Решение классической транспортной задачи в Excel

Задача. Для перевозки продовольственных грузов имеется три оптово-розничных склада № 1, № 2

и № 3 и два магазина № 1 и № 2. Объемы перевозки грузов составят соответственно 1000, 1500 и 1200 кг. Ежедневная потребность магазинов (распределительных центров) составляет 2300 и 1400 кг товара. Расстояние (в км) между заводами и распределительными центрами приведены в табл. 1.

Таблица 1

Мощность грузопотоков

Поставщики	Потребители	
	Магазин № 1	Магазин № 2
Склад № 1	10	26
Склад № 2	12	13
Склад № 3	14	8,5

Транспортная компания оценивает свои услуги в 1 копейку за перевозку одного кг груза на один км. В результате получаем представленную в табл. 2 стоимость перевозок (в рублях) по каждому маршруту.

Таблица 2

Стоимость перевозки товаров

Поставщики	Потребители	
	Магазин № 1	Магазин № 2
Склад № 1	0,1	0,26
Склад № 2	0,12	0,13
Склад № 3	0,14	0,085

Основываясь на данных из табл. 2, формулируем следующую задачу линейного программирования.

Минимизировать

$$F = 0,1x_{11} + 0,26x_{12} + 0,12x_{21} + 0,13x_{22} + 0,14x_{31} + 0,085x_{32} \Rightarrow \min$$

при ограничениях

$$x_{11} + x_{12} = 1000 \text{ (Склад № 1),}$$

$$x_{21} + x_{22} = 1500 \text{ (Склад № 2),}$$

$$x_{31} + x_{32} = 1200 \text{ (Склад № 3),}$$

$$x_{11} + x_{21} + x_{31} = 2300 \text{ (Магазин № 1),}$$

$$x_{12} + x_{22} + x_{32} = 1400 \text{ (Магазин № 2),}$$

$$x_{ij} \geq 0, i=1,2,3, j=1,2.$$

Эти ограничения выражены в виде равенств, поскольку общий объем перевезенных грузов со складов ($S=1000+1500+1200=3700$) равен суммарному спросу магазинов ($D=2300+1400=3700$).

Данную задачу можно решить с помощью так называемой **транспортной таблицы**. Решение данной задачи в Excel представлено на рис. 2 - 4.

Microsoft Excel - Лабораторная работа № 2

ФайлПравкаВидВставкаФорматСервисДанныеОкноСправка

Arial10ЖКЧ

F28fx

	A	B	C	D	E	F	G	H	I
1	Стоимость перевозки одного кг груза на один км. руб.								
2			Магазин № 1	Магазин № 2					
3		Склад № 1	0,1	0,26					
4		Склад № 2	0,12	0,13					
5		Склад № 3	0,14	0,085					
6									
7	Таблица-план оптимального закрепления								
8	Поставщики	Предложение	Потребители		Сумма				
9			Магазин № 1	Магазин № 2					
10			Спрос						
11			2300	1400		3700			
12	Склад № 1	1000	0	0	0				
13	Склад № 2	1500	0	0	0				
14	Склад № 3	1200	0	0	0				
15	Сумма	3700	0	0					
16									
17	Таблица-план расходов на перевозку								
18	Поставщики	Предложение	Потребители		Сумма				
19			Магазин № 1	Магазин № 2					
20			Спрос						
21			2300	1400		3700			
22	Склад № 1	1000	0	0	0				
23	Склад № 2	1500	0	0	0				
24	Склад № 3	1200	0	0	0				
25	Стоимость перевозок, руб.	3700	0	0	0				
26									
27									
28									
29									
30									

User:
Целевая ячейка

Рис. 2. Таблицы решения транспортной задачи

Исходные данные для решения классической транспортной задачи целесообразно представить в виде трех таблиц (см. рис. 2), в первой из которых представлены значения стоимости перевозок единицы товара c_{ij} от i -го поставщика к j -му потребителю.

Заполняем таблицу **Стоимость перевозки одного кг груза на один км, руб.** (рис. 1).

Во второй таблице представлены: значения S_i предложения каждого i -го поставщика; значения D_j спроса каждого j -го потребителя; переменные x_{ij} , первоначально принимающие нулевые значения; вспомогательная строка и вспомогательный столбец "Сумма".

Далее заполняем графы **Таблицы-плана оптимального закрепления** (рис. 2).

Ниже формируем **Таблицу-план расходов на перевозку**. Пересечения столбцов и строк представляют собой произведения соответствующих ячеек двух предыдущих таблиц.

Целевая ячейка **E25** должна содержать формулу, выражающую целевую функцию:
= СУММ(C22:D24)

Используя меню **Сервис**⇒**Поиск решения** открываем диалоговое окно **Поиск решения** (см. рис. 3), в котором устанавливаем целевую ячейку равной минимальному значению, определяем диапазон изменяемых ячеек и ограничения и запускаем процедуру вычисления, щелкнув по кнопке **Выполнить**.

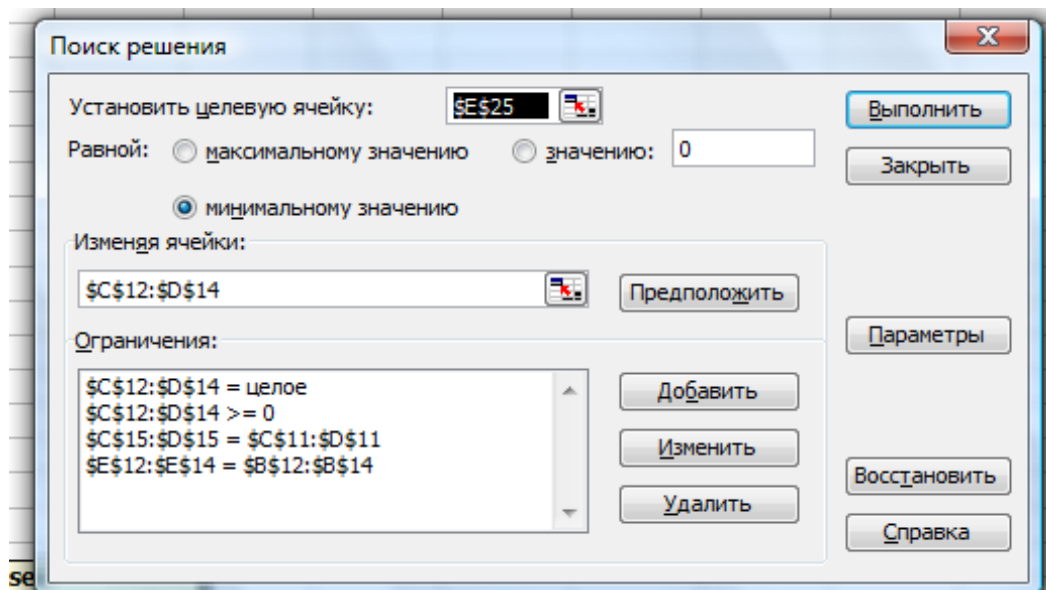


Рис. 3. Окно поиска решения

Оптимальное решение данной задачи представлено на рис. 4. Оно предполагает перевозку 1000 кг товара со склада №1 до магазина № 1, 1300 кг – со склада № 2 до магазина № 1, 200 кг товара со склада № 2 до магазина № 2 и 1200 кг товара со склада № 3 до магазина № 2. Минимальная стоимость перевозок составляет 384 рубля.

Microsoft Excel - Лабораторная работа № 2

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Arial 10 Ж К Ч

E25 =СУММ(C22:D24)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Стоимость перевозки одного кг груза на один км, руб.											
2			Магазин № 1	Магазин № 2								
3		Склад № 1	0,1	0,26								
4		Склад № 2	0,12	0,13								
5		Склад № 3	0,14	0,085								
6												
7	Таблица-план оптимального закрепления											
8			Потребители									
9	Поставщики	Предложение	Магазин № 1	Магазин № 2	Сумма							
10			Спрос									
11			2300	1400	3700							
12	Склад № 1	1000	1000	0	1000							
13	Склад № 2	1500	1300	200	1500							
14	Склад № 3	1200	0	1200	1200							
15	Сумма	3700	2300	1400								
16												
17	Таблица-план расходов на перевозку											
18			Потребители									
19	Поставщики	Предложение	Магазин № 1	Магазин № 2	Сумма							
20			Спрос									
21			2300	1400	3700							
22	Склад № 1	1000	100	0	100							
23	Склад № 2	1500	156	26	182							
24	Склад № 3	1200	0	102	102							
25	Стоимость перевозок, руб.	3700	256	128	384							
26												

Результаты поиска решения

Решение найдено. Все ограничения и условия оптимальности выполнены.

Тип отчета: Результаты, Устойчивость, Пределы

☒ Сохранить найденное решение

☐ Восстановить исходные значения

OK Отмена Сохранить сценарий... Справка

User: Целевая ячейка

Рис. 4. Оптимальное распределение товара по маршрутам

Когда суммарный объем предложений (грузов, имеющих в пунктах отправления) не равен общему объему спроса на товары (грузы), запрашиваемые пунктами назначения, транспортная задача называется **несбалансированной**. В этом случае, при решении классической транспортной задачи **методом потенциалов**, применяют прием, позволяющий несбалансированную транспортную задачу сделать сбалансированной. Для этого вводят **фиктивные** пункты назначения или отправления. Выполнение баланса транспортной задачи необходимо для того, чтобы иметь возможность применить алгоритм решения, построенный на использовании транспортных таблиц. В Excel несбалансированная транспортная задача решается путем изменения ограничений по спросу (если спрос превышает

предложение) или по предложению (если предложение превышает спрос), т.е. система ограничений будет иметь вид:

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = S_i, \\ \sum_{i=1}^m x_{ij} \leq D_j, \\ x_{ij} \geq 0, \\ x_{ij} \in N \cup \{0\}. \end{array} \right.$$

в первом случае, либо

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} \leq S_i, \\ \sum_{i=1}^m x_{ij} = D_j, \\ x_{ij} \geq 0, \\ x_{ij} \in N \cup \{0\}. \end{array} \right.$$

во втором случае.

При решение несбалансированной транспортной задачи в Excel, в отличии от вышеприведенного примера в окне поиска решения приведено ниже. Оно заключается в определении ограничений для поиска решения.

Например, уменьшив потребный объем перевозок для первого магазина до 2000 кг, мы получим результат решения, аналогичный приведенному на рисунке ниже. На этом же рисунке приведены ограничения, накладываемые для поиска решения.

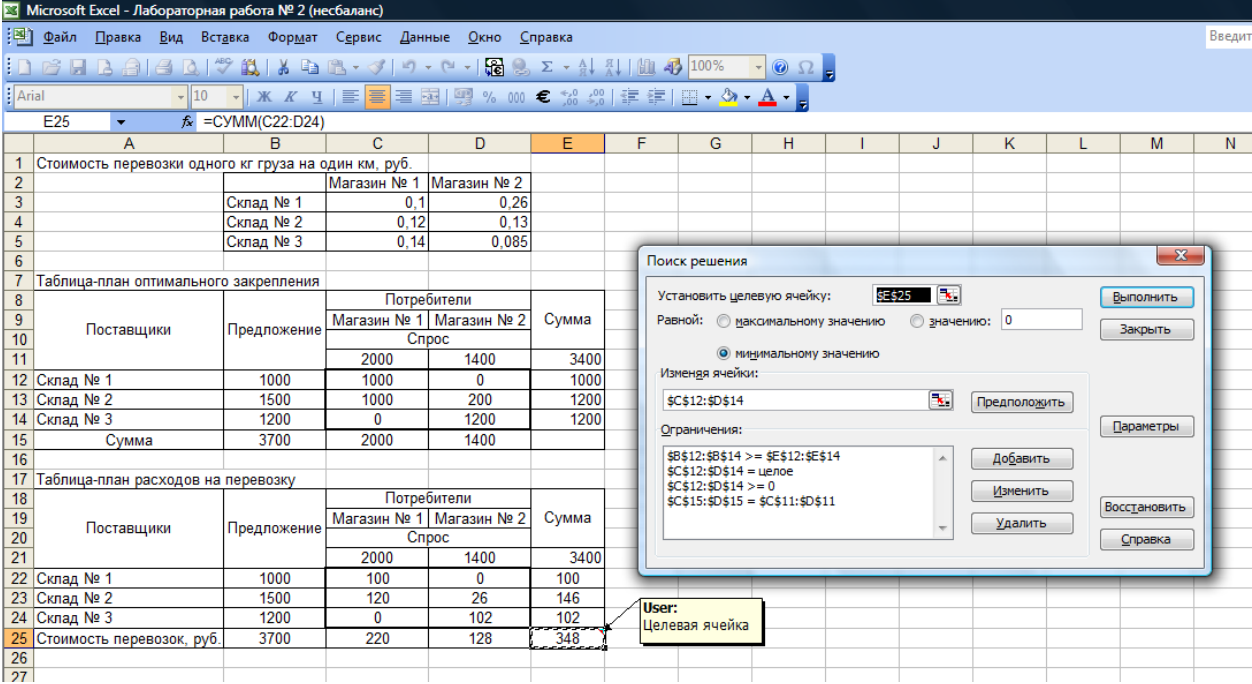


Рис. 5. Пример решения несбалансированной транспортной задачи

Задания к лабораторной работе
«Классическая транспортная задача»

Решите задачу 2, используя исходные данные о характеристике заказов, представленные в табл. 3.

Таблица 3

Варианты заданий для самостоятельного решения

Последняя цифра учебного шифра студента	Грузополучатели	Требуемый объем перевозок грузополучателям, т	Расстояние от грузоотправителя до грузополучателя, км			Возможный объем перевозок от грузоотправителей, т		
			1	2	3	1	2	3
1	1-ый	255	10	15	35	120	200	175
	2-ой	225	40	70	60			
2	1-ый	190	10	50	80	69	56	175
	2-ой	110	17	20	30			
3	1-ый	50	100	75	50	33	34	90
	2-ой	140	20	15	100			
4	1-ый	100	10	25	40	112	100	108
	2-ой	220	20	10	12			
5	1-ый	145	100	35	15	87	45	65
	2-ой	52	20	15	100			
6	1-ый	90	10	50	80	45	20	120
	2-ой	120	17	20	30			
7	1-ый	115	100	75	50	110	30	75
	2-ой	100	40	70	60			
8	1-ый	200	15	125	35	95	115	180
	2-ой	190	20	10	12			
9	1-ый	65	10	125	35	25	35	15
	2-ой	10	44	19	27			
10	1-ый	90	10	50	80	45	20	120
	2-ой	95	54	20	30			

Вариант подбирается по последней цифре шифра.

Порядок выполнения работы

- Изучить теоретические основы решения поставленной задачи.
- Получить вариант индивидуального задания у преподавателя (или выбрать по учебному шифру).
- Открыть Excel и выполнить операции по созданию табличной модели:
 - сформировать на рабочем листе таблицу **Стоимость перевозки одного кг груза на один км, руб.** Занести в нее данные о стоимости перевозок C_{ij} в зависимости от расстояния перевозки от поставщика до грузополучателя;
 - сформировать на рабочем листе **Таблица-план оптимального закрепления**. В ячейках будут определены объемы перевозок X_{ij} от i -го поставщика (в строках) до j -го потребителя (в столбцах). Под строкой «Спрос» необходимо подставить значения объемов потребления для j -го потребителя, а под строкой «Предложение» - значения объемов отправки для i -го поставщика. Остальные ячейки будут заполнены в результате поиска решения, для начала заполните их нулями;
 - сформировать на рабочем листе **Таблица-план расходов на перевозку**, представляющую суммарные эксплуатационные затраты, которые являются результатом перемножения соответствующих ячеек таблиц «Стоимость перевозки одного кг груза на один км, руб.» и «Таблица-план оптимального закрепления».
 - занести в итоговую ячейку целевую функцию $=\text{СУММ}(C_{ij} X_{ij})$.
- Оптимизировать модель, выбрав команду **Поиск решения** в меню **Сервис**.
В открывшемся диалоговом окне **Поиск решения** указать данные, необходимые для процесса оптимизации и ограничения, и нажать кнопку **Выполнить**.
- Сохранить расчетные данные для отчета о лабораторной работе, указав в качестве имени файла № группы и фамилии студентов, выполнивших работу.
- Формулировка выводов.
- Составление списка использованной литературы.
- Написание отчета.

Лабораторная работа 6. Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов.

Задание. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве **a1, a2 и a3** тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно **b1, b2, b3, b4, b5** тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

a1=200, a2=250, a3=200,

b1=190, b2=100, b3=120, b4=110, b5=130.

$$D = \begin{pmatrix} 28 & 27 & 18 & 27 & 24 \\ 18 & 26 & 27 & 32 & 21 \\ 27 & 33 & 23 & 31 & 34 \end{pmatrix}$$

Решение.

1. Построим начальный план двумя методами: методом северо-западного угла и методом наименьшей стоимости, и выберем тот план, который будет наилучшим, то есть получим минимальные затраты за перевозку однородного груза.

А) Строим начальный план методом северо-западного угла. Составим таблицу значений:

Потребители Поставщики	B1	B2	B3	B4	B5	Запасы
A1	28 190	27 10	18	27	24	200, 10, к
A2	18	26 90	27 120	32 40	21	250, 160, 40, к
A3	27	33	23	31 70	34 130	200, 130, к
Потребности	190, к	100, 90, к	120, к	110, 70, к	130, к	650=650

Число назначенных перевозок **m+n-1=3+5-1=7**, то есть начальный план

$$x_{11} = 190, \quad x_{12} = 10, \quad x_{22} = 90, \quad x_{23} = 120, \quad x_{24} = 40, \quad x_{34} = 70, \quad x_{35} = 130$$

невыврожденный.

При таком плане суммарные транспортные издержки равны:

$$F = 28 \cdot 190 + 27 \cdot 10 + 26 \cdot 90 + 27 \cdot 120 + 32 \cdot 40 + 31 \cdot 70 + 34 \cdot 130 = 5320 + 270 + 2340 + 3240 + 1280 + 2170 + 4420 = 19040(\text{единиц})$$

Б) Строим начальный план методом наименьшей стоимости. Составим таблицу значений:

Потребители Поставщики	B1	B2	B3	B4	B5	Запасы
A1	28	27 10	18 120	27	24 70	200, 80, 10, к
A2	18 190	26	27	32	21 60	250, 60, к
A3	27	33 90	23	31 110	34	200, 90, к
Потребности	190, к	100, 90, к	120, к	110, к	130, 70, к	650=650

Начальный план:

$$x_{12} = 10, \quad x_{13} = 120, \quad x_{15} = 70, \quad x_{21} = 190, \quad x_{25} = 60, \quad x_{32} = 90, \quad x_{34} = 110$$

При таком плане транспортные издержки

$$F = 27 \cdot 10 + 18 \cdot 120 + 24 \cdot 70 + 18 \cdot 190 + 21 \cdot 60 + 33 \cdot 90 + 31 \cdot 110 = 270 + 2160 + 1680 + 3420 + 1260 + 2970 + 3410 = 15170(\text{единиц})$$

Сравнивая транспортные издержки, видим, что план, построенный методом наименьшей стоимости, лучший.

2. Выбираем лучший план и находим потенциалы поставщиков и потребителей, используя первое условие оптимальности плана: $u_i + v_j = c_{ij}$

Потребители, v_j Поставщики, u_i		21	27	18	25	24
		B1	B2	B3	B4	B5
0	A1	28	27 10	18 120	27	24 70
-3	A2	18 190	26	27	32	21 60
6	A3	27	33 90	23 +1	31 110	34

Используя первое условие оптимальности плана, составим систему линейных уравнений для определения потенциалов:

$$\begin{cases} u_1 + v_2 = 27 \\ u_1 + v_3 = 18 \\ u_1 + v_5 = 24 \\ u_2 + v_1 = 18 \\ u_2 + v_5 = 21 \\ u_3 + v_2 = 33 \\ u_3 + v_4 = 31 \end{cases}$$

Система линейных уравнений содержит 7 уравнений и 8 неизвестных, т.е. она имеет множество решений. Так как нужно одно решение, то любой из неизвестных задаем значение и вычисляем остальные неизвестные.

Пусть $u_1 = 0$, тогда

$$u_2 = -3, \quad u_3 = 6, \quad v_1 = 21, \quad v_2 = 27, \quad v_3 = 18, \quad v_4 = 25, \quad v_5 = 24$$

1. Проверяем второе условие оптимальности плана для свободных клеток $u_i + v_j \leq c_{ij}$. Если есть нарушения, то заносим их со знаком «+». В результате проверки получили одну потенциальную клетку. Таким образом, начальный план не оптимален.
2. Улучшение плана. Выбираем клетку с максимальным нарушением и для нее строим замкнутый контур.

Потребители, v_j						
		B1	B2	B3	B4	B5
Поставщики, u_i	A1	28	27 + 10	18 - 120	27	24 70
	A2	18 190	26	27	32	21 60
	A3	27	33 - 90	23 - +1	31 110	34

Среди клеток, помеченных знаком «-», выбираем наименьшую перевозку:

$$q = \min(90, 120) = 90$$

На эту величину увеличиваем перевозки в клетках, помеченных знаком «+», и уменьшаем в клетках, помеченных знаком «-». В результате переназначения перевозок имеем план:

Потребители, v_j		21	27	18	26	24	
		B1	B2	B3	B4	B5	
Поставщики, u_i	0	A1	28	27 100	18 30	27	24 70
	-3	A2	18 190	26	27	32	21 60
	5	A3	27	33	23 90	31 110	34

Используя первое условие оптимальности плана, составим систему линейных уравнений для определения потенциалов:

$$\begin{cases} u_1 + v_2 = 27 \\ u_1 + v_3 = 18 \\ u_1 + v_5 = 24 \\ u_2 + v_1 = 18 \\ u_2 + v_5 = 21 \\ u_3 + v_3 = 23 \\ u_3 + v_4 = 31 \end{cases}$$

Система линейных уравнений содержит 7 уравнений и 8 неизвестных, т.е. она имеет множество решений. Так как нужно одно решение, то любой из неизвестных задаем значение и вычисляем остальные неизвестные.

Пусть $u_1 = 0$, тогда

$$u_2 = -3, \quad u_3 = 5, \quad v_1 = 21, \quad v_2 = 27, \quad v_3 = 18, \quad v_4 = 26, \quad v_5 = 24$$

Проверяем второе условие оптимальности плана для свободных клеток $u_i + v_j \leq c_{ij}$. Условие оптимальности выполнены, следовательно, план, соответствующий таблице, оптимален.

$$x_{12} = 100, \quad x_{13} = 30, \quad x_{15} = 70, \quad x_{21} = 190, \quad x_{25} = 60, \quad x_{33} = 90, \quad x_{34} = 110$$

$$F = 27 \cdot 100 + 18 \cdot 30 + 24 \cdot 70 + 18 \cdot 190 + 21 \cdot 60 + 23 \cdot 90 + 31 \cdot 110 = 2700 + 540 + 1680 + 3420 + 1260 + 2070 + 3410 = 15080 (\text{единиц})$$

Ответ: Сравнивая три метода нахождения оптимального плана, делаем вывод, что метод потенциалов находит оптимальный план решения транспортной задачи, так как получили минимальные транспортные издержки равные 15080 единиц.

Задания для самостоятельной работы

1 вариант.

Задача. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве **a1, a2 и a3** тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно **b1, b2, b3, b4, b5** тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

$$a1=200, \quad a2=350, \quad a3=300,$$

$$b1=270, \quad b2=130, \quad b3=190, \quad b4=150, \quad b5=110.$$

$$D = \begin{pmatrix} 24 & 50 & 55 & 27 & 16 \\ 50 & 47 & 23 & 17 & 21 \\ 35 & 59 & 55 & 27 & 41 \end{pmatrix}$$

2 вариант.

Задача. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве **a1, a2 и a3** тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно **b1, b2, b3, b4, b5** тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15

A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

$a1=200, a2=300, a3=250,$

$b1=210, b2=150, b3=120, b4=135, b5=135.$

$$D = \begin{pmatrix} 20 & 10 & 13 & 13 & 18 \\ 27 & 19 & 20 & 16 & 22 \\ 26 & 17 & 19 & 21 & 23 \end{pmatrix}$$

3 вариант.

Задача. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве $a1, a2$ и $a3$ тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно $b1, b2, b3, b4, b5$ тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

$a1=230, a2=250, a3=170,$

$b1=140, b2=90, b3=160, b4=110, b5=150.$

$$D = \begin{pmatrix} 40 & 19 & 25 & 25 & 35 \\ 49 & 26 & 27 & 18 & 38 \\ 46 & 27 & 36 & 40 & 45 \end{pmatrix}$$

4 вариант.

Задача. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве $a1, a2$ и $a3$ тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно $b1, b2, b3, b4, b5$ тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

$a1=350, a2=200, a3=300,$

$b1=170, b2=140, b3=200, b4=195, b5=145.$

$$D = \begin{pmatrix} 22 & 14 & 16 & 28 & 30 \\ 19 & 17 & 26 & 36 & 36 \\ 37 & 30 & 31 & 39 & 41 \end{pmatrix}$$

5 вариант.

Задача. Имеются три пункта поставки однородного груза **A1, A2, A3** и пять пунктов **B1, B2, B3, B4, B5** потребления этого груза. На пунктах **A1, A2 и A3** находится груз соответственно в количестве $a1, a2$ и $a3$ тонн. В пункты **B1, B2, B3, B4, B5** требуется доставить соответственно $b1, b2, b3, b4, b5$ тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5

A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

$a1=300, a2=250, a3=200,$

$b1=210, b2=150, b3=120, b4=135, b5=135.$

$$D = \begin{pmatrix} 4 & 8 & 13 & 2 & 7 \\ 9 & 4 & 11 & 9 & 17 \\ 3 & 16 & 10 & 1 & 4 \end{pmatrix}$$

Лабораторная работа 7. Общие сведения об языке логического программирования

Пример 1. Наша первая Пролог программа будет содержать информацию о военнослужащих некоторого воинского подразделения и их званиях: “Павлов генерал”, “Сабо полковник”, “Денисов капитан”, “Матвеев капитан”, “Кулёмин сержант”, “Николаев сержант”. Сформулировать на Прологе следующие вопросы: 1) Павлов генерал? 2) Кто является полковником? 3) Кем является Денисов? 4) В подразделение есть военный в звании сержанта? 5) В подразделение есть военный в звании подполковника? 6) Вывести военных, имеющих одинаковые звания.

В программе каждого военного мы представим предикатом `military` размерности 2, каждый компонентом, первый представляет фамилию, а второй – его звание.

Программа 3. База данных «Военная часть»

Domains

`s=symbol`

Predicates

`military(s,s)`

Clauses

`military(pavlov, general).`

`military(sabo, polkovnik).`

`military(denisov, kapitan).`

`military(matveev, kapitan).`

`military(kulemin, serzhant).`

`military(nikolaev, serzhant).`

Сформулируем запросы:

1) ? `military(pavlov, general)`

Ответ: yes

2) ? `military(X, polkovnik)`

Ответ: X= sabo

3) ? `military(denisov, X)`

Ответ: X= kapitan

4) ? `military(_, serzhant)`

Ответ: yes

5) ? `military(_, podpolkovnik)`

Ответ: no

6) ? `military(X, Y), military(Z, Y), X<>Z`

Ответ: X= denisov Z= matveev Y= kapitan

X= kulemin Z= nikolaev Y= serzhant

Пример 2. Данные о крупных реках России сведены в таблицу:

Таблица 5.

Данные о крупных реках России

Название реки	Длина, км	Годовой сток, км ³	Площадь бассейна, тыс. км ²	Истоки	Куда впадает
Амур	4416	350	1855	Яблоневый хребет	Татарский пролив
Лена	4400	488	2490	Байкальский хребет	Море Лаптевых
Обь	4070	400	2990	Предгорья Алтая	Карское море
Иртыш	4248	323	1643	Китай	Обь
Енисей	3487	600	2580	Восточный Саян	Карское море

Волга	3530	255	1360	Валдайская возвышенность	Каспийское море
Колыма	2129	44	643	Хребет Черского	Восточносибирское море
Урал	2428	54	231	Южный Урал	Каспийское море
Дон	2200	45	504	Среднерусская возвышенность	Азовское море
Кама	1805	130	507	Верхне — Камская возвышенность	Волга
Печора	1809	130	322	Северный Урал	Баренцево море
Ангара	1779	62	1039	Байкал	Енисей
Селенга	1024	14	447	Монголия	Байкал
Кубань	870	11	58	Кавказ	Азовское море

Составить базу данных и ответить на следующие вопросы:

- 1) Определить реки, впадающие в Азовское море.
- 2) Определить реки, исток которых находится на Валдайской возвышенности?
- 3) Какие реки короче Камы?
- 4) Какие реки длиннее Иртыша?
- 5) Как задать вопрос, определяющий все данные о реке Кама?

Программа 4. База данных «Реки России»

Domains

S=symbol

N=integer

Predicates

reka(S,N,N,N,S,S)

Clauses

reka(амур, 4416, 350, 1855, yablonevi_hrebet, tatar_proliv).

reka(лена, 4400, 488, 2490, baikal_hrebet, more_laptevih).

reka(об, 4070, 400, 2990, altai, more_karskoe).

reka(иртиш, 4248, 323, 1643, kitai, ob).

reka(енисей, 3487, 600, 2580, vost_cain, more_karskoe).

reka(волга, 3530, 255, 1360, valdais_vozvishennost, more_kaspi).

reka(колыма, 2129, 44, 643, hrebet_cherskogo, vost_sibir_more).

reka(урал, 2428, 54, 231, yuzhni_ural, more_kaspi).

reka(дон, 2200, 45, 504, sredn_rus_vozvishennost, more_azov).

reka(кама, 1805, 130, 507, verhne_kamsk_vozvishennost, volga).

reka(печора, 1809, 130, 322, sever_ural, barenzevo_more).

reka(ангара, 1779, 62, 1039, baikal, enisei).

reka(селенга, 1024, 14, 447, mongolia, baikal).

reka(кубань, 870, 11, 58, kavkaz, more_azov).

Запросы:

reka(X, _, _, _, more_azov)

reka(X, _, _, valdais_vozvishennost, _)

reka(X, Y, _, _, _), reka(лена, Z, _, _, _), Y<Z

reka(X, Y, _, _, _), reka(иртиш, Z, _, _, _), Y>Z

reka(кама, A,B,C,D,E)

Пример 3. Известно, что Лене нравится теннис, Денису нравится футбол, Борису – бейсбол, Эдику – плавание, Марку нравится теннис, а Фёдору то, что нравится Борису. Записать факты на Прологе и ответить на вопросы: 1) Кому нравится теннис? 2) Что нравится Федору? 3) Кто занимается одинаковыми видами спорта?

Программа 5. База знаний «Предпочтения»

Predicates

likes(symbol,symbol)

Clauses

```

likes(lena, tennis).
likes(denis, football).
likes(boris, baseball).
likes(edic, swimming).
likes(mark, tennis).
likes(fedor, Activity):- likes(boris, Activity).
/* Activity играет роль переменной*/

```

Запросы

1) ? likes(X, tennis)

Ответ:

X= lena

X= mark

2) ? likes(fedor, X)

Ответ:

X= baseball

3) ? likes(X, T), likes(Y, T)

Ответ:

X= lena Y= mark T= tennis

X= mark Y= lena T= tennis

X= boris Y= fedor T= baseball

X= fedor Y= boris T= baseball

Пример 4. Лена, Анна, Денис и Борис-люди, лада и нисан - автомобили, Лене нравится лада, Анне - пицца, Денису - футбол, а Борис - Мерседес, Васке - рыбка. Пицца, лада, мерседес продаются. Человек может купить машину, если она продается, и она ему нравится. Сформулировать на прологе вопросы:
 1)Какую машину может купить Лена? 2) Кто-нибудь может купить мерседес? 3) какие машины продаются, и ответить на них.

Программа 6. База знаний «Предпочтения и возможности»

Domains

s=symbol

Predicates

human(s)

car(s)

likes(s,s)

can_by(s,s)

cells(s)

Clauses

human(lena).

human(anna).

human(denis).

human(boris).

car(lada).

car(nissan).

likes(lena, lada).

likes(anna, pizza).

likes(denis, football).

likes(boris, mercedes).

likes(vasya, ribka).

cells(pizza).

cells(lada).

cells(mercedes).

can_by(X,Y):-human(X), car(Y), likes(X,Y), cells(Y).

Запросы:

1) can_by(lena, X)

X=lada

2) can_by(_,mercedes)

no

3) car(X),cells(X)

X=lada

Пример 5. Программа иллюстрирует различные способы ввода данных.

Программа 7. Ввод данных

```
Predicates
  vvod_int
  vvod_ch
  vvod_s
Clauses
  vvod_int:- readint(N1), readint(N2), N=N1+N2, write(N).
  vvod_ch:- readchar(N1), readchar(N2), N=N1+N2, write(N).
  vvod_s:- readln(N1), readln(N2), concat(N1,N2,N), write(N).
```

Пример 6. Вывести в каждой строке сообщения: Леонард отец Катерины, Карл отец Джейсона, Карл отец Марины.

Программа 8. База знаний «Семья»

```
Domains
  name = symbol
Predicates
  father(name, name)
  everybody
clauses
  father(leonard, katherine).
  father(carl, jason).
  father(carl, marina).
  everybody :-
    father(X, Y),
    write(X, " is ", Y, "s father\n"),
    fail.
```

В некоторых случаях может быть необходимым продолжение поиска дополнительных решений, для этого можно использовать встроенный предикат **fail**. Он не имеет аргументов, всегда считается ложным.

Проверьте работу программы с и без использования предиката fail.

Отрицание задается с помощью предиката **not**.

Пример 7. У нас есть информация о странах-партнерах Европы, имеющих общую границу.

Предположим, нас интересуют какие страны-партнеры не имеют общей границы.

Программа 9. База знаний «Страны Европы»

```
Domains
  country=symbol
Predicates
  euro_pair(country, country)
  border(country, country)
  find_non_border_pair
Clauses
  euro_pair("France", "Germany").
  euro_pair("France", "Spain").
  euro_pair("France", "Italy").
  euro_pair("Germany", "Spain").
  euro_pair("Germany", "Italy").
  euro_pair("Spain", "Italy").
  border("France", "Germany").
  border("France", "Spain").
  border("France", "Italy").
  find_non_border_pair:-
    euro_pair(X,Y),
    not(border(X,Y)),
    write(X," - ",Y),fail,nl.
Goal
  find_non_border_pair
```

Результатом запроса будет ответ

Germany - Spain. Germany - Italy. Spain - Italy.

Пример 8. Использование составных объектов

Программа 10. База данных «Коллекция»(Вариант 1)

Domains

personal_library=book(title, author, publisher, year)
 collector, title, author, publisher=symbol
 year=integer

Predicates

collection(collector, personal_library)

Clauses

collection(ivanov, book("Zolushka", "Denis Tarakanov", "Dinamo", 2003)).
 collection(petrov, book("Maslenniza", "Anna Zimina", "Zenit", 2005)).
 collection(petrov, book("Repka", "Irina Larina", "Mir", 1999)).

/*Демонстрация двухуровневого составного объекта*/

Программа 11. База данных «Коллекция»(Вариант 2)

Domains

personal_library=book(title, author, publication)
 publication= publication (publisher, year)
 collector, title, author, publisher =symbol
 year=integer

Predicates

collection(collector, personal_library)

Clauses

collection(ivanov,book("Zolushka","Den Taran", publication ("Dina", 2003))).
 collection(petrov,book("Maslenniza","Anna Zima",publication("Zenit",2005))).
 collection(petrov, book("Repka","Irina Larina",publication ("Mir", 1999))).

/*Демонстрация использования конструкций альтернативных доменов*/

Программа 12. База знаний «Клуб по интересам»

Domains

thing=misc_ thing (whatever);
 book(author, title);
 record(artist, album, type)
 person, whatever, title, author,artist, album, type =symbol

Predicates

owns(person, thing)
 show_misc_things
 show_books
 show_records

Clauses

owns ("Ivanov", misc_things("sports car")).
 owns ("Petrov", misc_things("motor cycle")).
 owns ("Smirnov", misc_things("piano")).
 owns ("Ivanov", book ("James A.Mishener", "Space")).
 owns ("Petrov", book ("Frank Herbert", "Dune")).
 owns ("Sidorov", book ("J.R.R. Tolkein", "Return of the Ring")).
 owns ("Ivanov", record ("Elton John", "Ice Fair", "popular")).
 owns ("Petrov", record("Michael Jackson", "We are the World", "popular")).
 owns ("Sidorov", record ("Madonna", "Madonna", "popular")).

show_misc_things:- owns (X, misc_things(Y)),write(X, " ",Y), nl, fail.

show_books:- owns (X, book (_,Y)),write(X, " ",Y), nl, fail.

show_records:- owns (X, record (_,Y,_)),write(X, " ",Y), nl, fail.

Задания для самостоятельной работы

1) Дана база данных "Родители и дети":

родитель(полина, борис), родитель(анатолий, борис), родитель(анатолий, лиза), родитель(борис, катя), родитель(борис, валентина), родитель(полина, евгений).

Сформулировать вопросы на Прологе: Кто является родителем Кати? Есть ли у Лизы ребенок? Кто дети Бориса? Кто чей родитель?

2) Дана база данных “Теремок”:

живет(муха, горюха), живет(комар, пискун), живет(мышка, погрызуха), живет(лягушка, квакушка), живет(заюнок, кривоног),
живет(лиса, краса), живет(волк, хватыш), не_живет(медведь, пригнетыш).

Указать ответы на следующие вопросы:

?-живет(мышка, погрызуха). ? -живет(волк, X).

?-живет(X, кривоног). ?-не_живет(M,P).

Сформулировать вопросы на Прологе: Живет ли лягушка в теремке? Какое прозвище у лисы? Кто имеет прозвище горюха? Какой следует задать вопрос, чтобы узнать обитателей теремка (без прозвищ)?

3) База данных “Рождение и хобби друзей”:

рождение(иванова, лена, 22, июнь, 1971), рождение(петров, сергей, 25, октябрь, 1973),
рождение(сидорова, оля, 1, декабрь, 1974), любит(иванова, лена, книги), любит(иванова, лена, танцы), любит(петров, сергей, видео), любит(сидорова, оля, кино).

Сформулировать вопросы на Прологе: Кто родился в 1971 году? Кто родился в октябре? Кто любит книги? Кто любит и книги и танцы?

4) База данных “Колобок”: ушел(колобок, дедушка), ушел(колобок, бабушка),
ушел(колобок, заяц), ушел(колобок, волк), ушел(колобок, медведь), не_ушел(колобок, лиса).

Сформулировать вопросы на Прологе: Кто ушел от волка? Кто не ушел от лисы? Кто ушел от волка и от бабушки? Какой следует задать вопрос, чтобы узнать всех персонажей сказки?

5) База данных “Распорядок дня”:

занятие(0, 7, сон), занятие(7, 8, завтрак), занятие(8, 13, школа), занятие(13, 14, обед),
занятие(14, 19, свобода), занятие(19, 20, ужин), занятие(20, 23, отдых), занятие(23, 24, сон).

Сформулировать вопросы на Прологе: Когда бывает обед? Что бывает между 14 и 19 часами? Когда бывает сон? (сколько будет решений?)

6) Построить базу данных “Важнейшие события Древнего Мира” на основе установленных фактов, произошедших с 31 по 6 век до нашей эры.

Каждый факт приводить в виде событие(X, Y, Z), где X — название государства, где произошло событие, Y - в каком веке произошло событие, Z — какое произошло событие.

В 31-м веке до нашей эры возникли первые города-государства. Единое государство в Египте образовалось в 30 веке до нашей эры. В 27 веке до нашей эры в Индии появились первые древнейшие города, а в Египте построена пирамида Хеопса. Первые греческие государства появились в 18 веке до нашей эры. В этом же веке в Египте произошло крупное восстание бедняков и рабов. В 15 веке до нашей эры появились первые государства в Китае. Тутмос III правил в Египте в 15 веке до нашей эры. Греция вела троянскую войну в 13 веке до нашей эры. Вторжение борийских племен в Грецию произошло в 11 веке до нашей эры. В 8 веке до нашей эры был основан город Рим. Олимпийские игры стали проводиться в Греции в 8 веке до нашей эры. В 6 веке до нашей эры в Риме была установлена республика, а в Греции произошли реформы Солона. В этом же веке персы взяли Вавилон в Междуречье и завоевали Египет.

1. Составить 3 запроса к этой базе данных.
2. Какие события произошли в период с 15 до 7 в. до н.э.

7) В таблице даны некоторые характеристики движения планет Солнечной системы(числовые величины округлены):

Таблица 6.

Характеристики движения планет солнечной системы

Планета	Расстояние до Солнца (у.е.)	Период обращения	Средние солнечные сутки
Меркурий	39	88 суток	176 суток
Венера	72	225 суток	117 суток
Земля	100	365 суток	24 часа
Марс	152	687 суток	25 часов
Юпитер	520	12 лет	10 часов
Сатурн	954	29 лет	10 часов
Уран	1920	84 года	24 часа
Нептун	3010	165 лет	22 часа
Плутон	3950	247 лет	6 суток

Составить базу данных, учитывая измерение по некоторым параметрам в разных единицах.

Ответить на вопросы: Какие планеты ближе к Солнцу, чем Земля? Какие планеты дальше от Солнца, чем Земля? На каких планетах солнечные сутки меньше, чем земные? На каких планетах период обращения измеряется в годах?

8) Построить базу знаний “Рабочая смена”:

Мария работает в дневную смену. Сергей работает в вечернюю смену. Борис работает в вечернюю смену. Валентина работает в вечернюю смену. Два служащих знают друг друга, если они работают в одну смену. Определить: Знает ли Сергей Бориса? Кого знает Валентина? Кого знает Мария?

9) Даны результаты сдачи экзаменов для группы из пяти учеников:

Таблица 7.

Успеваемость

Фамилия	Алгебра	Геометрия	История
Бобров	5	3	2
Вяткин	5	5	5
Кротов	2	3	3
Соснин	4	4	4
Вавилов	4	2	1

Построить базу знаний о результатах экзаменов, определив в ней следующие правила:

отличник - человек, у которого по всем предметам пятерки;

двоечник - есть хотя бы одна двойка;

математик - по алгебре и по геометрии учится на 4 и 5;

введите предикаты алгебра, геометрия, история для определения оценки Y для ученика X .

Получить ответы на следующие вопросы: Является ли Вяткин отличником? Определить всех отличников. Является ли Соснин математиком? Определить всех неуспевающих по истории.

10) Сформировать базу знаний “Квартет” из следующих фактов и правил:

Марышка играет на скрипке. Осел играет на альте. Козел играет на виолончели. Мишка играет на контрабасе. Четверо музыкантов X, Y, Z и W могут образовать квартет, если один из них играет на скрипке, другой — на альте, третий — на виолончели и четвертый — на контрабасе.

Ответить на вопросы: Кто играет на альте? На чем играет мартышка? Образуют ли квартет Мартышка, Осел, Козел и Мишка? Кто из музыкантов данной базы знаний может образовать квартет?

11) База знаний "Воинская служба":

возраст(борис, 18), возраст(андрей, 17), возраст(михаил, 18), возраст(анна, 18), возраст(юлия, 17), мужчина(андрей), мужчина(борис), мужчина(михаил), женщина(анна), женщина(юлия).

Определить правило подлжит призыву, не_подлжит_призыву.

Сформулировать вопросы: Кто подлжит призыву? Подлжит ли призыву Анна?

12) База знаний "Семья":

мать(екатерина, юлия), мать(екатерина, мария), мать(анна, екатерина), отец(петр, юлия), отец(виктор, петр), отец(андрей, екатерина).

Дополните базу данных предикатом мужчина, женщина. Определите правила дед, бабушка, внук, внучка, тетя, дядя.

Сформулировать вопросы на Прологе:

Кто является ребенком Екатерины и Петра? Кто является дедом Юлии? Кто является бабушкой Юлии?

13) Построить базу знаний и сформулировать к ней вопросы, основываясь на следующих утверждениях: Резвый - это собака. Рекс-это собака. Белка – это кошка. Быстрая - это лошадь. Резвый - черная. Белка - белая. Рекс -рыжая. Быстрая - белая. Домашние животные - это собака или кошка. Животные – это либо лошадь, либо домашние животные. Том владеет тем, кто собака и не черного цвета. Кейт владеет тем, кто либо черного цвета, либо лошадь.

14) Построить базу знаний. Муська — коричневая кошка, Стрелка — черная кошка, Мурка — рыжая кошка. Рекс, Дружок и Мухтар — собаки. Дружок — рыжая, Мухтар — белая. Все животные, которыми владеют Анатолий и Николай, имеют родословные. Анатолий владеет всеми черными и коричневыми животными, а Николай владеет всеми собаками небелого цвета, которые не являются собственностью Анатолия. Иван владеет Муркой, если Николай не владеет Муськой и если Мухтар не имеет родословной. Рекс — пятнистая собака. Определить, какие животные не имеют хозяев.

15) Составить базу знаний "Знакомства" из следующих фактов и правил:

Мери прелестна. Джон добрый. Джон мужественный. Джон сильный. Некто счастлив, если богатый или нравится женщинам. Мужчина нравится женщине, если женщина нравится мужчине и он добрый, либо мужчина добрый и сильный. Мужчине нравится женщина, если она прелестна.

Сформулировать вопрос: счастлив ли Джон? Найти мужчин, которые могут нравиться женщинам.

16) Ответьте на вопрос "Что ест моя кошка?", если:

Птицам нравятся червяки. Кошкам нравятся рыбы. Друзья нравятся друг другу. Моя кошка мой друг. Моя кошка ест все, что ей нравится.

17) Запрограммируйте утверждения.

Число четное. Число не четное. Ни одно число не является четным и нечетным одновременно. Число не четное, если следующее за ним четное. Число, следующее за данным числом нечетное, если данное число четное, число, следующее за данным числом четное, если данное число нечетное.

Лабораторная работа 8. Арифметика. Управление логическим выводом в программах

Пример 1. Описать предикаты для вычисления суммы, разности, произведения, частного двух чисел, возведения числа в квадрат, вывода остатка при делении на 3, вывод случайного числа из интервала [1,100].

Программа 13. Арифметика

Domains

N=integer

R=real

Predicates

add(N,N)

sub(N,N,N)

multi(N,N,N)

division(N,N,R)

kvadrat(N,N)

ostat(N,N)

vivod(N)

Clauses

add(X,Y):-S=X+Y, write("Sum= ",S),nl.

sub(X,Y,S):-S=X-Y.

multi(X,Y,P):-P=X*Y.

division(X,Y,R):-Y<>0, R=X/Y.

kvadrat(X,N):-N=X*X.

ostat(X,N):-N=X mod 3.

vivod(N):-random(Y), N=1+Y*100.

Пример 2. Программирование взаимоисключающих утверждений. Процедуру нахождения

наибольшего из двух чисел можно записать в виде отношения

$\text{max}(X, Y, X):-X \geq Y.$

$\text{max}(X, Y, Y):-X < Y.$

Эти правила являются взаимоисключающими. Возможна более экономная формулировка: если $X \geq Y$, то максимум= X , иначе $=Y$. На Прологе это запишется следующим образом:

$\text{max}(X, Y, X):-X \geq Y, !.$

$\text{max}(_, Y, Y).$

Программа 14. Максимум

Domains

$N = \text{integer}$

Predicates

$\text{max}(N, N, N)$

Clauses

$\text{max}(X, Y, X):-X > Y, !.$

$\text{max}(_, Y, Y).$

Пример 3. Рассмотрим различные способы записи предиката `different`, определяющего различны ли числа, использующие сочетание встроенных предикатов `!` и `fail`.

$\text{different}(X, X):-!, \text{fail}.$

$\text{different}(_, _).$

или

$\text{different}(X, Y):-X = Y, !, \text{fail}.$

$\text{different}(_, _).$

или

$\text{different}(X, Y):-X = Y, !, \text{fail}; \text{true}.$

/ true –встроенный предикат, который всегда истинен*/*

или

$\text{different}(X, Y):-\text{not}(X = Y).$

Задания для самостоятельной работы

№1. Описать предикаты:

- 1) Найти квадрат числа X ; куб числа X .
- 2) Найти значение функций $y = a * x + b$, где a , b и x — целые числа.
- 3) Найти периметр треугольника, если все его стороны известны.
- 4) Найти площадь прямоугольного треугольника по двум его катетам.
- 5) Найти площадь трапеции с основаниями A и B и высотой H .
- 6) Найти квадрат гипотенузы в прямоугольном треугольнике по двум его катетам.
- 7) Найти объем прямоугольного параллелепипеда со сторонами A , B и C .
- 8) Зная скорость V и время T , определите путь.
- 9) Найти последнюю цифру в записи натурального числа.
- 10) Найти цифры в десятичной записи двузначного натурального числа.
- 11) Найти первую цифру в десятичной записи трехзначного натурального числа.
- 12) Найти сумму цифр в десятичной записи трехзначного натурального числа.

№2.

- 1) Найти

А. Наименьшее значение из двух чисел;

Б. Наименьшее значение из трех чисел на основе первой задачи;

В. Наименьшее значение из шести чисел на основе второй задачи.

- 2) Определить, удовлетворяют ли длины трех отрезков условию прямоугольного треугольника.
- 3) Определить, удовлетворяют ли длины трех отрезков условию треугольника.
- 4) Найти модуль числа X .
- 5) Описать предикат для вычисления функции, заданной соотношением:

$$f(x) = \begin{cases} x^2, & x < -1 \\ x + 1, & -1 < x < 1 \\ x^2, & x > 1 \end{cases}$$

Лабораторная работа 9. Повторение и рекурсия

Пример 1. Выводить на экран квадрат числа, вводимого пользователем, пока не будет введен 0.

```

Domains
  x=integer
Predicates
  go
  repeat
  check(x)
Clauses
  repeat.
  repeat:- repeat.
  go:- repeat, write ("Введите число пожалуйста или 0-для выхода "),
        readint(A), check(A).
  check(0):-nl,write("ok"),!.
  check(A):-B=A*A, write(B),nl, fail.
Или
  go:- repeat, write ("Введите число "), readint(A),
        B=A*A, write(A, "^2= ",B),nl,
        write("прервать да/нет (y/n) "), readchar(C), C= "n",!.

```

Пример 2. Организовать меню для выбора арифметической операции.

Программа 16. Меню

```

Domains
  x=integer
Predicates
  menu
  actions(x)
  repeat
Clauses
  repeat.
  repeat:- repeat.
  menu:- repeat,
        write("Введите 1 для +, 2 для *, 3-для -, 0-для выхода\n"),
        readint(N),
        actions(N).
  actions(0):-!.
  actions(1):-write("Введите первое число"), nl, readint(A),
        write("Введите второе число") nl, readint(B),
        C=A+B,write(C),nl,fail.
  actions(2):- write("Введите первое число"), nl, readint(A),
        write("Введите второе число") nl, readint(B),
        C=A*B,write(C),nl,fail.
  actions(3):- write("Введите первое число"), nl, readint(A),
        write("Введите второе число") nl, readint(B),
        C=A-B,write(C),nl,fail.

```

Пример 3. Вычислить n-ый член последовательности Фибоначчи. N-ый член последовательности Фибоначчи, начиная с третьего, определяется суммой 2-х предыдущих, а 1-ый и 2-ой члены равняются единице.

Введем двуместный предикат fib, первый аргумент будет определять порядковый номер члена, а второй будет записываться для записи ответа. Введем два факта, первый - первый член последовательности Фибоначчи равен 1, второй - второй член последовательности равен 1, а для определения n-го члена запишем правило. Действительно, чтобы определить n-ый член, мы должны определить значения двух предыдущих и сложить их.

```

fib(1,1).
fib(2,1).
fib(N,F):- N1=N-1, fib(N1,F1), N2=N-2, fib(N2,F2), F=F1+F2.
? fib(1,F)

```

Ответом будет F=1, и Пролог сделает попытку сопоставить с запросом второй факт и потерпит неудачу. Однако сопоставление головы третьего утверждения с запросом происходит успешно и осуществляется попытка доказать цель fib(-1,F1), что в свою очередь, приводит к цели fib(-2, ...) и так далее, т.е. образуется бесконечный цикл. Эту ситуацию можно устранить, используя отсечение и тем самым, указывая Прологу, что не существует других решений в случае успешного согласования граничного условия.

Программа 17. Последовательность Фибоначчи

Domains

x=integer

Predicates

fib(x,x)

Clauses

fib(1,1):-!.

fib(2,1):-!.

fib(N,F):- N1=N-1, fib(N1,F1), N2=N-2, fib(N2,F2), F=F1+F2.

Задания для самостоятельной работы

- 1) Вычислить $N!$.
- 2) Вычислить n -ый член последовательности Фибоначчи.
- 3) Вывести все числа от n до 1.
- 4) Вывести все числа от 1 до n .
- 5) Вычислить сумму чисел от 1 до n .
- 6) Определите x^n , $n > 0$.
- 7) Определите 2^n , $n > 0$.
- 8) Определите N^5 , $n > 0$.
- 9) Вычислите сумму четных чисел от 1 до n .
- 10) Вычислите сумму квадратов нечетных чисел от 1 до n .
- 11) Вычислите сумму a_k , где $a_k = 1/(1+k)$.
- 12) Вычислить $\sum_{i=1}^N x^i$.
- 13) Вычислить $\sum_{i=1}^N i^i$.
- 14) Определите корень уравнения методом половинного деления.
- 15) Найти наибольший общий делитель двух чисел, трех чисел.
- 16) Определить число сочетаний C_n^m .
- 17) Вычислить $N! + (N-1)! + \dots + 2! + 1!$.
- 18) Вычислить количество четных элементов на заданном интервале.
- 19) Перевести число из десятичной системы счисления в систему с основанием N , где $N < 10$, $N > 1$.

Лабораторная работа 10. Применение рекурсии для обработки списков

Пример 1. Определить количество элементов в списке.

Количество элементов пустого списка равно 0, в противном случае, нужно определить количество элементов в хвосте и найденное значение увеличить на единицу.

Программа 18. Определение длины списка

Domains

list = integer*

Predicates

length_of(list, integer)

Clauses

length_of([], 0):-!.

length_of([_|T], N) :- length_of(T, N1), N = N1 + 1.

?- length ([2, 12, 45]), X).

Ответ: X=3.

Пример 2. Найти сумму элементов в списке.

Сумма элементов пустого списка равна 0, сумма элементов непустого списка определяется сложением головы списка с суммой элементов хвоста.

sum ([], 0):-!.

sum ([H|T], S) :- sum (T, S1), S = S1+H.

?-sum ([2, 12, 45]), X).

Ответ: X=59

Пример 3. Определить принадлежность элемента списку.

Программа 19. Определение принадлежности элемента списку

```
Domains
  type=integer
  list=type*
Predicates
  member(type,list)
Clauses
  member(H,[H|_]):-!.
  member(H,[_|T]):-member(H,T).
```

Запрос:
member (4,[1,3,4,9]).

Ответ: Yes.

Данная программа имеет очень простой декларативный смысл: элемент принадлежит списку, если он является его головой или принадлежит хвосту списка.

Пример 4. Объединить два списка.

Эту задачу можно описать с помощью следующих предикатов:

- а) если к пустому списку [] добавить список P, то в результате получится P;
- б) чтобы добавить список P к концу списка [X|Y], нужно P добавить к хвосту Y и затем присоединить к голове X, при этом получается список [X|T].

Программа 20. Объединение двух списков

```
Domains
  type=integer
  list=type*
Predicates
  append(list,list,list)
Clauses
  append([],L,L):-!.
  append([H|T],P,[H|Y]):-append(T,P,Y).
```

? append ([3, 5, 7], [12,5],K).

Ответ: K=[3,5,7,12,5]

Процедуру append можно использовать:

- 1) для разбиения списка на подсписки:

? append (X,Y,[1,2])

Ответ: X=[] Y=[1,2]

X=[1] Y=[2]

X=[1,2] Y=[]

- 2) Для вывода последнего элемента списка:

last(X,L):-append(_,[X],L)

Пример 5. Удаление элементов из списка.

Первый аргумент указывает удаляемый элемент, второй аргумент - исходный список и третий - список-результат.

Программа 21. Удаление элементов из списка

```
Domains
  type=integer
  list=type*
Predicates
  delete(type,list,list)
Clauses
  delete(X,[X|T],T):-!.
  delete(X,[Y|T],[Y|T1]):-delete(X,T,T1).
```

Смысл: результат удаления произвольного элемента из пустого списка, есть пустой список, если удаляемый элемент совпадает с головой списка, то результатом программы является хвост списка, иначе удаление производится из хвоста списка.

Данная программа удаляет первое вхождение элемента в список. Знак отсечения "!" в конце правила предотвращает последующий поиск и вывод лишних вариантов ответов после выполнения ограничительного факта.

Для удаления всех вхождений элемента X программу надо дополнить:

```
delete(, [], []) :-!.
delete(X, [X | T], W) :- delete(X, T, W),!.
delete(X, [Y | T], [Y|W]) :- delete(X, T, W) .
```

Смысл программы таков: пока список не пуст, удалить элемент, если он совпадает с головой списка, а затем удалять его из хвоста, иначе надо сразу удалять элемент из хвоста.

Пример 6. Индексация элементов списка.

Определить номер элемента X.

```
nomer([X | _], 1, X):-!.
nomer( [_ | T], N, X) :- nomer(T, N1, X), N=N1+1.
```

Пример 7. Вывести максимальный элемент.

```
max ( [X] , X) .
max ([X | T], X) :- max (T, M), X>M, !.
max ( [_ | T] , M) :- max (T, M) .
```

Смысл программы: если в списке один элемент - он и является максимальным, если более одного, то это голова списка, если она больше максимального элемента хвоста, или максимальный элемент хвоста.

Пример 8. Обращение списка.

Обратить список из одного элемента - означает оставить список без изменения. Обратить более длинный список - обратить его хвост, а потом справа добавить к нему голову исходного списка.

```
reverse([X], [X]):-!.
reverse ([X | T], Z) :- reverse (T, W), append(W, [X], Z).
```

Пример 9. Иногда бывает полезно представить в виде списка информацию, содержащуюся в известных фактах. В Прологе для этой цели служит предикат **findall**. Полный формат предиката: **findall**(Переменная, Атом, Список), здесь Атом- отношения, которое определяет список элементов, конкретизирующих переменную в качестве аргумента предиката атом. Рассмотрим предикат на примере вычисления общей суммы окладов служащих

Программа 22. Сумма окладов служащих

```
Domains
    i=integer
    sp=i*
    fam,im,dolgnost=symbol
Predicates
    sum(sp,i)
    go
    spisok(fam,im,dolgnost,i)
Clauses
    sum([],0):-!.
    sum([H|T],S):-sum(T,S1),S=S1+H.
    spisok(ivanov, ivan, slesar,400).
    spisok(petrov, petr,tokar,200).
    spisok(sidorov, denis,plotnik,100).
    go:-findall(X,spisok(_,_,_),X), sum(L,A).
```

Задания для самостоятельной работы

- 1) Определите количество элементов в списке.
- 2) Определите сумму элементов списка
- 3) Определите количество нечетных элементов в списке.
- 4) Определите, принадлежит ли заданный элемент списку.
- 5) Определите, сколько раз заданный элемент входит в список.
- 6) Выведите максимальный элемент.
- 7) Выведите голову списка.
- 8) Выведите последний элемент.
- 9) Замените голову списка.
- 10) Определите номер элемента X.
- 11) Выведите элемент под номером N.
- 12) Удалите из списка все вхождения заданного элемента.

- 13) Объедините два списка.
- 14) Перепишите список в обратном порядке.
- 15) Объедините два списка без дублирования элементов.
- 16) Удалите первое вхождение заданного элемента.
- 17) Сложить поэлементно 2 списка.
- 18) Сложить два списка следующим образом: $a_1+b_n, a_2+b_{n-1}, \dots, a_{n-1}+b_2, a_n+b_1$.
- 19) Найти количество элементов, предшествующих первому(последнему) максимальному.
- 20) Переместите голову списка в конец списка.
- 21) Найти сумму максимального и минимального элементов списка.
- 22) Поменяйте местами элементы с нечетными индексами с элементами с четными индексами.
- 23) Составить список из цифр заданного числа в обратном порядке. Например, 127645: [5,4,6,7,2,1].
- 24) Выполните N последовательных перестановок головы в конец списка.
- 25) Увеличьте каждый элемент списка на заданный элемент.
- 26) Увеличьте элемент с заданным номером на заданное число.
- 27) Все вхождения заданного элемента уменьшите на заданное число.
- 28) Удалите элемент с заданным номером N.
- 29) Создать список, элементами которого являются $2^n, n!$, члены последовательности Фибоначчи.
- 30) Определите среднее элементов списка.
- 31) Замените четные элементы списка нулем.
- 32) Определите сумму элементов, больше заданного N.
- 33) Отсортируйте список методом пузырька.
- 34) Отсортируйте список методом вставками.
- 35) Отсортируйте список быстрым методом сортировки.
- 36) Используя предикат findall, решите следующие задачи:
 1. Вывести самых молодых жильцов дома и номера квартир, в которых они живут.
 2. Вывести фамилии студентов и их возраст с максимальным размером стипендии.
 3. Вывести фамилии сотрудников предприятия и их оклады, оклады которых меньше среднего.
 4. Вывести студентов с заданной фамилией и посчитать их количество.

Лабораторная работа 11. Решение логических задач.

Пример 1. Беседует трое друзей: Белокуров, Рыжов, Чернов. Брюнет сказал Белокурову: “Любопытно, что один из нас блондин, другой брюнет, третий - рыжий, но ни у кого цвет волос не соответствует фамилии”. Какой цвет волос у каждого из друзей?

Для решения построим вспомогательную таблицу:

Таблица 8.

Вспомогательная таблица соответствия

Цвет \ Фамилия	Белокуров	Рыжов	Чернов
блондин	-		
рыжий		-	
брюнет	-		-

Выводы:

- 1) Белокуров не брюнет и не блондин;
- 2) Чернов не черный, цвет волос Чернова и Белокурова не совпадают;
- 3) Рыжов не рыжий, цвет волос у Рыжова и Белокурова, Рыжова и Чернова не совпадают.

Программа 23. Логическая задача на соответствие

Predicates

```
fam(symbol)
color(symbol)
cootvet(symbol, symbol)
```

Clauses

```
fam(belokurov).
fam(ryzov).
fam(chernov).
color(ryziy).
color(cherniy).
color(beliy).
cootvet(X,Y):- fam(X), color(Y), X=belokurov,
               not(Y=cherniy), not(Y=beliy).
cootvet(X,Y):- fam(X), color(Y), X= chernov,
```

```

not(Y=cherniy),not(cootvet(belokurov,Y).
cootvet(X,Y):- fam(X), color(Y), X= ryzov,
not(cootvet(belokurov, Y), not(cootvet(chernov, Y).

```

Пример 2. На скамейке сидели Петя, Боря, Коля. Петя справа от Бори, Боря справа от Коли. Кто сидел посередине? Кто сидел с правого(левого) края? Кто сидел между указанными объектами(увеличьте число объектов)?

Программа 24. Логическая задача на выяснение порядка

Predicates

```

rayd(symbol, symbol, symbol)
sprava(symbol, symbol)
seredina(symbol
kr_cl(symbol)
kr_cpr(symbol)

```

Clauses

```

sprava(kolya, boray). /*Справа от Коли Боря*/
sprava(boray, petay).
rayd(X,Y,Z):- sprava(X,Y), sprava(Y, Z).
seredina(X):- rayd(_,X,_).
kr_cl(X):- rayd(X,_,_).
kr_cpr(X):- rayd(_,_,X).

```

Задания для самостоятельной работы

- 1) Коля и Саша носят фамилии Шилов и Гвоздев. Какую фамилию носит каждый из них, если Саша с Шиловым живут в разных домах.
- 2) В соревнованиях по бегу Юра, Гриша и Толя заняли три первых места. Какое место занял каждый ребенок, если Гриша занял не второе и не третье место, а Толя не третье?
- 3) Три подружки вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфель совпадали. Ни туфли, ни платье Вали не были белыми. Наташа была в зеленых туфлях. Определить цвета платья и туфель на каждой из подруг.
- 4) На заводе работали три друга: слесарь, токарь и сварщик. Их фамилии Борисов, Иванов и Семенов. У слесаря нет ни братьев, ни сестер. Он самый младший из друзей. Семенов, женатый на сестре Борисова, старше токаря. Назвать фамилии слесаря, токаря и сварщика.
- 5) В бутылке, стакане, кувшине и банке находятся молоко, лимонад, квас и вода. Известно, что вода и молоко не в бутылке, сосуд с лимонадом находится между кувшином и сосудом с квасом, в банке - не лимонад и не вода. Стакан находится около банки и сосуда с молоком. Как распределены эти жидкости по сосудам.
- 6) Воронов, Павлов, Левицкий и Сахаров – четыре талантливых молодых человека. Один из них танцор, другой художник, третий-певец, а четвертый-писатель. О них известно следующее: Воронов и Левицкий сидели в зале консерватории в тот вечер, когда певец дебютировал в сольном концерте. Павлов и писатель вместе позировали художнику. Писатель написал биографическую повесть о Сахарове и собирается написать о Воронове. Воронов никогда не слышал о Левицком. Кто чем занимается?
- 7) Три друга заняли первое, второе, третье места в соревнованиях универсиады. Друзья разной национальности, зовут их по-разному, и любят они разные виды спорта. Майкл предпочитает баскетбол и играет лучше, чем американец. Израильянин Саймон играет лучше теннисиста. Игрок в крикет занял первое место. Кто является австралийцем? Каким спортом увлекается Ричард?
- 8) Три девочки Маша, Рита, Лена пошли гулять. На улице было жарко, и они купили мороженое «Белка», «Стрелка», «Гагара». Какое мороженое купила каждая из девочек, если Лена купила не «Белку» и не «Гагару», а Рита – не «Гагару».
- 9) В комнате находятся Коля, Света, Оля. Каждый из них сидит на отдельной мебели (кровать, стул, диван). Известно, что Коля сидит не на стуле и не на кровати. Света не сидит на стуле. Кто где сидит?
- 10) На столе лежат ручка, карандаш, фломастер, красного, синего и зеленого цвета. Известно, что ручка лежит между предметом красного и зеленого цвета. Карандаш либо зеленый, либо синий.
- 11) Однажды в Артеке за круглым столом оказалось пятеро ребят из Москвы, Санкт-Петербурга, Новгорода, Перми, Томска: Денис, Игорь, Иван, Алеша, Сергей. Москвич сидел между томичем и Сергеем, Санкт-петербуржец - между Денисом и Игорем, а напротив него сидел пермяк и Иван. Алеша ни разу не был в Санкт-Петербурге, а Денис не бывал в Москве и Томске, а томич с Игорем регулярно переписываются. Определите, кто в каком городе живет каждый из ребят.
- 12) На улице, встав в кружок, беседует четыре девочки: Аня, Валя, Надя, Галя. Девочка в зеленом платье – не Аня и не Валя - стоит между девочкой в голубом платье и Галей. Девочка в белом платье стоит между девочкой в розовом платье и Валей. Какого цвета платье у каждой из девочек?
- 13) Трое юношей: Коля, Дима и Юра влюблены в трех девушек: Аню, Лену, Вику. Но эта любовь без взаимности. Коля любит девушку, влюбленную в юношу, который любит Лену. Дима любит девушку, влюбленную в юношу, который любит Вику. Лена не любит Юру.

14) Составить базу знаний по сказке “Репка”. Фактами в этой базе должны быть утверждения типа тянет(X,Y). Составить правила, определяющие: кто первый тянет репку, кто последний тянет реку, кто тянет после бабки, кто тянет на четвертом месте.

15) Даны числа X, Y, Z, T. X меньше Y и меньше T; Y больше Z и больше T; Z больше X и меньше T. В каком порядке расположены эти числа.

Лабораторная работа 12. Головоломки. Игровые программы.

Пример 1. Нарисовать конверт, не отрывая карандаша от бумаги и не проводя два раза по одной и той же линии. Введем обозначения, ребра – латинскими буквами, вершины - цифрами.

Знания о структуре графа можно представить следующим образом: rebro(a,1,2), rebro(a,2,1), rebro(b,2,3), rebro(b,3,2), rebro(c,1,3), rebro(c,1,2), rebro(c,3,1), rebro(d,2,4), rebro(d,4,2), rebro(e,2,5), rebro(e,5,2), rebro(f,3,4), rebro(f,4,3), rebro(g,3,5), rebro(g,5,3), rebro(h,4,5), rebro(h,5,4).

Решением задачи должен явиться список пройденных ребер графа, причем длина его должна быть равна 8, и в нем не должно быть повторяющихся ребер, что можно описать так:

```
prohod(T,P):-length(P,8),write_list(P),!.
```

```
prohod(T,P):-rebro(R,T,H), not(member(R,P), prohod(H,[R|P]).
```

T-текущая вершина графа, а P-список пройденных ребер. Первое правило означает, что если длина списка P пройденных вершин становится равной 8, то список P выводится на печать. Это правило ограничивает рекурсивный перебор вершин и ребер, проводимый вторым правилом. Второе правило является генератором перебора, оно перебирает предикаты “rebro()” и находит такое ребро R, из текущей вершины T в новую H, чтобы оно не принадлежало списку P, зато это ребро добавляется в качестве головы к списку P, и поиск дальнейшего пути производится уже из новой вершины H.

Пример 2. Программа “Угадай число”.

Программа 25. Логическая игра «Угадай число»

Predicates

```
play_the_game
give_info
play_it
generate_rand(integer)
play_it_sam(integer)
test_and_tell(integer,integer)
say_you_got_it_right(integer)
say_too_big
say_too_small
```

Clauses

```
play_the_game:-give_info, play_it.
/*информация для пользователя*/
give_info:-makewindow(1,7,7,"",0,0,25,80),
makewindow(2,7,7,"Угадай число", 2,20,22,45),
nl, write("Это игра Угадай число"),
nl, write("Я загадаю число из"),
nl, write("интервала [1,100]"),
nl, write("Я буду подсказывать больше"),
nl, write("или меньше задуманное число"),nl,
nl, write("Кода будешь готов- нажми space bar"),
nl,readchar(_),
clearwindow.
/*выполнение игры*/
play_it:-generate_rand(A),
play_it_sam(A).
/*генерация случайного числа*/
generate_rand(X):-
random(R),
X=1+R*100,
nl,write("Я загадал число"),
nl,write("Теперь дело за Вами"),nl.
/*запрос к пользователю */
play_it_sam(X):-
nl, write ("Введите свой вариант"),
nl, readint(G),
test_and_tell(X,G),
play_it_sam(X).
```

```

/* проверка и выдача результата*/
test_and_tell(X,X):- say_you_got_it_right(X),!.
test_and_tell(X,Y):- X>Y, say_too_big,!.
test_and_tell(_,_):- say_too_small.
/*выдача сообщений*/
say_too_big:-nl, write("Загаданное число больше").
say_too_small:-nl, write("Загаданное число меньше").
say_you_got_it_right(X):-nl, write("Вы правы"),
                        nl,write("Я загадал ",X),
                        nl, write("До новых встреч!"),
                        nl, write("Нажмите space bar"),
                        nl,readchar(_),
                        exit.

```

Goal

```
play_the_game.
```

Пример 3. Игра Баше – 23 спички. На столе 23 спички, 2 игрока по очереди берут от 1 до 3 спичек, проигравшим считается тот, кто возьмет последнюю спичку.

Программа 26. Логическая игра «Игра Баше – 23 спички»

Predicates

```

play_the_game
do_windows
play(integer,integer,integer)
play_it_again(integer,integer,integer)
real_int(real,integer)

```

Clauses

```

play_the_game:-do_windows,play(23,0,0).
/*правило для образования окон*/
do_windows:-makewindow(1,7,7,"",0,0,25,80),
             makewindow(2,7,7,"Игра 23 спички", 1,5,22,40),
             nl, write("Добро пожаловать!"),
             M=23, H=0, C=0,
             nl, write("Всего ",M," спички" ),
             nl, write("По очереди мы будем перемещать " ),
             nl, write("спички в свои кучки"),
             nl, write("За раз можно взять 1, 2, 3 спички"),nl,
             nl, write("Проиграет, тот кто заберет последнюю спичку"),
             nl, write("Итак, начнем, всего ", M, "сп."),nl,
             nl, write("я взял ",C," сп. Вы взяли ",H, " сп."),nl.
play(M,H,C):-play_it_again(M,H,C).
/*правило определения победителя*/
play_it_again(M,_,_):-
    M<=0,
    nl, write("Вы выиграли!"),
    nl, write("Нажмите любую клавишу для выхода"),
    readchar(_), clearwindow,!,exit.
play_it_again(1,_,_):-
    nl, write("Я выиграл!"),
    nl, write("Нажмите любую клавишу для выхода"),
    readchar(_), clearwindow,!,exit.
play_it_again(M,_,_):-
    nl, write("Ваш ход"),
    nl, write("Сколько спичек вы хотите взять?"),
    readint(Hn),
    M2=M-Hn, H2=Hn,
    write("Осталось ",M2,"сп."),
    nl,write("Мой ход"),
    random(F), Rea=1+3*F,
    real_int(Rea,Rint),
    M3=M2-Rint,
    nl, write("Я беру ",Rint," сп."),
    nl, write("Осталось ",M3, " сп."),nl,

```

```

M7=M3,H7=H2,C7=Rint,
play_it_again(M7,H7,C7).
/*вспомогательное правило*/
real_int(Re,In):-In=trunc(Re).
Goal
play_the_game

```

Задания для самостоятельной работы

- 1) Отгадать число, загаданное компьютером, максимум за 3 попытки.
- 2) 23 спички. Разработать выигрышную стратегию для компьютера.
- 3) Имеется два сосуда – на 3 и 5 литров. Как отмерить с их помощью 4 литра воды?
- 4) Решите задачу о перевозке фермером через реку волка, козы, капусты.
- 5) Раскрасить плоскую карту так, чтобы никакие две смежные области на ней не были раскрашены в одинаковый цвет. В наборе 4 цвета.

Таблица 9.

Поле для раскраски		
a		
b	c	d
e	f	

- 6) Лабиринт задан матрицей соединений, в которой для каждой пары комнат указано, соединены ли они коридором. Построить путь перехода из комнаты “a” в комнату “g”:

Таблица 10.

Поле для лабиринта							
	A	B	C	D	E	F	G
A	-	1	0	0	0	0	0
B	1	-	1	0	1	0	0
C	0	1	-	1	0	0	0
D	0	0	1	-	1	0	0
E	0	1	0	1	-	0	0
F	0	0	0	0	1	0	0
G	0	0	0	0	1	0	-

- 7) Имеется n городов. Некоторые из них соединены дорогами известной длины. Вся система дорог задана квадратной матрицей $n \times n$, элемент $a_{ij} < 0$, если между городами дороги нет, в противном случае равен расстоянию между городами.

- 8) Задан лабиринт. Напишите программу для нахождения маршрутов для выхода из этого лабиринта. Начало маршрута точка O. Закрашенная клетка – стена.

Таблица 11.

		O		

- 9) Быки и коровы. Игрок A выбирает секретный код, представляющий последовательность из N различных десятичных цифр ($N=4$). Игрок пытается угадать задуманный код и спрашивает игрока A о числе “быков” (“быки” - количество совпадающих цифр в одинаковых позициях предполагаемого и задуманного кодов; число “коров” - количество совпадающих цифр, входящих в предполагаемый и задуманный код, но находящихся на разных позициях).

Лабораторная работа 13. Обработка файлов. Предикаты для работы с файлами

deletefile(DOS_filename)-удаление файла

save(DOS_filename) –сохранение файла

renamefile(old_DOS_filename, new_DOS_filename)-переименовывание файла

existfile(DOS_filename)-проверка на наличие файла с таким именем

flush(file_domain)-сброс данных из внутреннего буфера, отведенного для данного устройства записи

disk(Path) –выбор дискового и пути доступа

dir(Path, File_spec, File_name) –Переменной Path должен быть присвоен корректный путь доступа, переменная File_spec задает расширение представляющей интерес группы файлов. Данный предикат выдает каталог имен файлов с заданным расширением; вы можете выбрать среди них нужный и нажать Enter. Имя файла будет присвоено переменной File_name.

При описании файловых доменов тип домена записывается по левую сторону от знака равенства, а имя домена по правую.

Пример:

file=datafile

file=datafile1;datafile2

openwrite(datafile,filename) - открытие файла для записи или создание, где datafile- введенный пользователем файловый домен, filename-имя файла в DOS, теперь ссылки на datafile будут означать обращение к filename.

writedevise(datafile) -назначение файла в качестве устройства записи

openread(datafile,filename) - открытие файла для чтения.

readdevice(datafile) -назначение файла устройством чтения

openmodify(datafile,filename) - открытие файла для редактирования, указатель помещается в начало файла, сместить указатель можно при помощи предиката fileros.

openappend(datafile,filename) - открытие файла для добавления данных в конец файла.

closefile(datafile) -закрытие файла

Рассмотрим примеры работы с файлами.

Пример 1. Вывести информацию на экран дисплея и в файл на диске.

Программа 27. Запись данных в файл

Domains

str = string

file = datafile

Predicates

data(str)

write_lines

Goal

openwrite(datafile,"file1.dat"),

write_lines,

closefile(datafile).

Clauses

data("Старому году оставьте печали,!").

data("Забудьте обиду, беду.").

data("Только успехов, здоровья и счастья,").

data("Мы Вам желаем в Новом году!").

write_lines:-

data(Line),

write(" ",Line),nl,

writedevise(datafile),

write(" ",Line),nl,

writedevise(screen), /*для вывода данных на экран*/

fail, write_lines.

Пример 2. Вывести данные файла на экран.

Программа 28. Чтение данных из файла

Domains

str = string

file = datafile

Predicates

read_write_lines

Goal

openread(datafile,"file1.dat"),

read_write_lines,

closefile(datafile).

Clauses

read_write_lines :-

readdevice(datafile),

not(eof(datafile)),

readln(Line),

writedevise(screen),

write(" ",Line),nl,

read_write_lines.

Пример 3. Записать в файл данные, вводимые с клавиатуры

Программа 29. Запись в файл данных, вводимых с клавиатуры

Domains

file = datafile


```

dstring, cstring = string
Predicates
  readin(dstring,cstring)
  create_a_file
Goal
  create_a_file.
Clauses
  create_a_file :-
    nl,nl,
    write("Пожалуйста, введите полное имя файла."),
    nl,nl,
    readln(Filename), openwrite(datafile,Filename),
    writedev(device,datafile),
    readln(Dstr),
    concat(Dstr,"\\13\\10",Cstr),
    readin(Dstr,Cstr),
    closefile(datafile).
  readin("done",_) :- !.
/*ввод данных завершится после ввода слова "done"*/
  readin(_,Cstr) :-
    write(Cstr),
    readln(Dstr1),
    concat(Dstr1, "\\13\\10",Cstr1),
    writedev(device,datafile),
    readin(Dstr1,Cstr1).

```

Предикаты для работы с файлами прямого доступа

Openmodify(fn,filename)-связывает логическое имя файла fn с именем файла

Filepos(fn, pos, mode)-устанавливает указатель файла в заданную позицию

Таблица 12.

Действие системы при операциях с файлами прямого доступа

Режим mode	Действия системы
0	Смещение берется относительно начала файла
1	Смещение берется относительно текущей позиции
2	Смещение берется относительно конца файла

Пример 4. Данные, вводимые с клавиатуры, записать в файл прямого доступа.

Программа 30. Запись данных в файл прямого доступа

```

Domains
  file = datafile
Predicates
  create_a_random_access_file
  write_read_more(real, string)
  pad_str (strIng,string,integer)
Goal
  create_a_random_access_file.
Clauses
  create_a_random_access_file :-
    write("Please enter filename:").nl,
    readln(Filename),
    openwrite(datafile,Filename),
    closefile(datafile),
    openmodify(datafile,Filename),
    write("Введите строку").nl,
    readln(Dstr),
    write_read_more(0,Dstr),
    closefile(datafile).
  write_read_more(_, "done") :-
    nl, write(" Press the space bar."),
    readchar(_),exit.
  write_read_more(Index,Dstr) :-
    writedev(device,datafile),
    filepos(datafile,Index,0),

```

```

pad_str(Dstr,Padstr,38),
concat(Padstr, "\10\13", Cstr),
write(Cstr),
writedev(device(screen),
write("Введите строку"),nl,
readln(Dstr1),
Index1 = Index + 40,
write_read_more (Index1, Dstr1).
pad_str (Instr,Instr,Length) :-
str_len(Instr,Testlength),
Testlength >= Length,!.
pad_str (Instr,Padstr,Length) :-
concat(Instr,"-",Newstr),
pad_str(Newstr,Padstr,Length).

```

Пример 5. Вывести на экран заданную строку файла прямого доступа и выдача их на экран

Программа 31. Вывод данных из файла прямого доступа

```

Domains
file = datafile
Predicates
read_a_random_access_file
Goal
read_a_random_access_file.
Clauses
read_a_random_access_file:-
write("Please enter filename: "), nl, readln(Filename),
openread(datafile,Filename),
write("Type 1n record number: "),nl,
readreal(Record),
Index = (Record - 1) * 40,
readdevice(datafile),
filepos(datafile,Index,0),
readln(Cstr),
write(Cstr), nl,nl,
write("Press the space bar."),nl,
readdevice(keyboard),
readchar(_),
closefile(datafile),
exit.

```

Задание для самостоятельной работы

- 1) В файле задана последовательность целых чисел, найти сумму чисел, предшествующих первому отрицательному.
- 2) Переписать данные файла в обратном порядке.
- 3) Вывести на экран данные файла в неубывающем порядке.
- 4) Записать результат сложения чисел, содержащихся в файлах в третий.
- 5) Переписать числа из файла в другой, дописав за каждым его квадрат.

Лабораторная работа 14. Создание динамической базы данных. Предикаты для работы с базой данных

Для описания предикатов динамической базой данных предназначен раздел **database**. База данных (БД) называется динамической потому, что во время работы можно добавлять, удалять содержащиеся в них утверждения. Другая важная особенность динамических баз данных состоит в том, что она может быть записана на диск и считана с диска в оперативную память.

Иногда предпочтительно иметь часть информации БД в виде утверждений статической БД. Эти данные заносятся в динамическую сразу после активизации программы. В общем случае, предикаты статической БД имеют другое имя, но ту же самую форму представления данных, что и предикаты динамической.

Предикаты для работы с динамической БД:

Asserta заносит новый факт в БД, новый факт помещается перед всеми уже внесенными утверждениями.

Assertz помещает новый факт в БД за всеми имеющимися утверждениями.

Retract удаляет утверждение из БД.

Save сохраняет находящуюся в оперативной памяти БД в текстовом файле. Синтаксис этого предиката таков **Save(DOS_file_name)**

consult(DOS_file_name) считывает в память файл БД

readterm(Domain,Term) используется для чтения из файла объектов, относящихся к определенному в программе домену.

Предикат findall позволяет собрать все имеющиеся в базе данные в список, который может быть полезен при дальнейшей работе.

В качестве примера рассмотрим БД по игрокам футбольных команд, БД допускает следующие операции: добавление, удаление и просмотр данных. Эта программа создает БД и содержит её в оперативной памяти. Для работы с ней использовался предикат player с аргументами *p_name*-имя игрока, *k_name*-название клуба, *p_number*-номер игрока, *pos*-позиция игрока, *height* -рост, *weight*-вес, *nfl_exp*-стаж выступлений, *college*-учебное заведение)

Программа 32. Динамическая база данных «Футбольная команда»

Domains

p_name,k_name, pos, college = string
p_number, height, weight, nfl_exp = integer

Database

dplayer(p_name, k_name,p_number,pos, height,weight,nfl_exp, college)

Predicates

repeat
do_mbase
assert_database
menu
process(integer)
clear_database
player(p_name, k_name,p_number,pos, height,weight,nfl_exp, college)
error

Goal

do_mbase.

Clauses

repeat.
repeat:-repeat.

/*База данных футбол*/

player("Иванов Иван","Спартак",13,"з", 205,90,3, "ГГПИ").
player("Петров Петр","Динамо",96,"пз", 185,78,4, "ГТК").
player("Сидоров Денис","Локомотив",69,"в", 190,88,2, "ГТУ").
player("Васечкин Илья","Торпедо",5,"в", 195,80,5, "ГГПИ").
player("Алексеев Дима","ЦСКА",1,"н", 165,67,2, "ГТК").
player("Карпов Павел","Зенит",12,"н",170,74,1, "ГКК").

/*конец начальных данных*/

assert_database:-

player(P_name,K_name,P_number,Pos,Height,Weight,Nfl_exp,College),
assertz(dplayer(P_name,K_name,P_number,Pos,Height,Weight,Nfl_exp,
College)),

fail.

assert_database:-!.

clear_database:-

retract(dplayer(_,_,_,_,_,_)),
fail.

clear_database:-!.

do_mbase :-

assert_database,
makewindow(1,7,7," FOOTBALL DATABASE ",0,0,25,80),
menu,
clear_database.

menu :-

repeat, clearwindow,
nl,
write(" ***** "),nl,
write(" * 1. Добавление нового игрока в БД * "),nl,
write(" * 2. Удаление игрока из БД * "),nl,
write(" * 3. Просмотр данных * "),nl,
write(" * 4. Выход из программы * "),nl,
write(" ***** "),nl,
write(" Пожалуйста, сделайте свой выбор 1, 2, 3 or 4 : "),

```

        readint(Vibor),nl,process(Vibor),Vibor = 4,!.
        /* Добавление информации об игроке в БД */
process(1) :-
    makewindow(2,7,7,"Добавление данных",2,20,18,58),shiftwindow(2),
    write("Введите, пожалуйста: "),nl,
    write("Имя игрока: "), readln(P_name),
    write("Название клуба: "), readln(K_name),
    write("Номер игрока: "), readint(P_num),
    write("Позицию: "), readln(Pos),
    write("Рост: "), readint(Ht),
    write("Вес: "), readint(Wt),
    write("Стаж выступлений: "), readint(Exp),
    write("Название учебного заведения: "), readln(College),
    assertz(dplayer(P_name, K_name, P_num,Pos,Ht,Wt,Exp, College)),
    write(P_name," добавлен в БД"), nl,!,
    write("Press space bar. "), readchar(_),
    removewindow, shiftwindow(1).
        /* Удаление */
process(2) :-
    makewindow(3,7,7,"Удаление данных",10,30,7,40),shiftwindow(3),
    write("Введите имя удаляемого игрока: "), readln(P_name),
    retract(dplayer(P_name,_,_,_,_,_,_)),
    write(P_name," удален из БД "), nl,!,
    write("Press space bar."), readchar(_), removewindow,
    shiftwindow(1).
        /* Просмотр данных об игроке*/
process(3) :-
    makewindow(4,7,7," Просмотр ", 7,30,16,47), shiftwindow(4),
    write("Введите имя для просмотра: "), readln(P_name),
    dplayer(P_name,T_name,P_number,Pos,Ht,Wt,Exp,College),nl,
    write(" Имя игрока      : ",P_name),nl,
    write(" Название клуба   : ",T_name),nl,
    write(" Номер игрока      : ",P_number),nl,
    write(" Позиция           : ",Pos),nl,
    write(" Рост              : ",Ht), nl,
    write(" Вес               : ",Wt),nl,
    write(" Стаж выступлений   : ",Exp), nl,
    write(" Учебное заведение : ",College),nl, nl,!,
    write("Press space bar"), readchar(_),
    removewindow, shiftwindow(1).
process(3) :-
    makewindow(5,7,7," Неудача ",14,7,5,60), shiftwindow(5),
    write("К сожалению, данных нет."),nl,
    write("Извините, пока!"),nl,!,
    write("Press space bar."),readchar(_),
    removewindow,shiftwindow(1).
        /* Выход */
process(4) :-
    write("До новых встреч! "),readchar(_),exit.
        /*Обработка ошибки*/
process(Vibor):-
    Vibor<1, error; Vibor>5, error.
error:-
    write("Пожалуйста выберите число от 1 до 4"),
    write("(Press the spase bar to continue"),readchar(_).

```

Задание для самостоятельной работы

Модифицируйте программу, добавив в меню пункты:

- 1) Вывод списка игроков.
- 2) Сохранение данных в файл.
- 3) Выборка данных по 1 из трех критериев.

Лабораторная работа 15. Применение языка для решения задач ИИ. Создание экспертных систем

Пример 1.Рассмотрим пример ЭС для идентификации породы собак. Она помогает потенциальному хозяину выбрать породу собаки в соответствии с определенными критериями.

В данной ЭС используются следующие характеристики:

1. Короткая шерсть;
2. Длинная шерсть;
3. Рост меньше 30 дюймов;
4. Рост меньше 22 дюймов;
5. Низкопосаженный хвост;
6. Длинные уши;
7. Хороший характер
8. Вес больше 100 фунтов.

Каждая характеристика для конкретной породы либо верна, либо не верна. Для каждой породы справедливы следующие характеристики:

Таблица 13.

Характеристики собак	
Порода	Характеристики
Английский бульдог	1,4,5,7
Гончая	1,4,6,7
Дог	1,3,6,7,8
Амер.гончая	1,5,6,7
Кокер-спаниэль	2,4,5,6,7
Ирландский сеттер	2,3,6
Колли	2,3,5,7
Сенбернар	2,5,7,8

Программа 33. «Эксперт по породам собак»

**Эксперт по породам собак*/*

*/*Назначение: Демонстрация работы ЭС*/*

Domains

n=integer

list=n*

dog=symbol

Predicates

rule(n,dog,list)

cond(n,string)

do_expert

show_menu

do_consulting

process(n)

test(n,list)

topic

repeat

Goal

do_expert.

Clauses

rule(1,"английский бульдог",[1,4,5,7]).

rule(2,"гончая",[1,4,6,7]).

rule(3,"дог",[1,3,6,7,8]).

rule(4,"американская гончая",[1,5,6,7]).

rule(5,"кокер-спаниель",[2,4,5,6,7]).

rule(6,"ирландский сеттер",[2,3,6]).

rule(7,"колли",[2,3,5,7]).

rule(8,"сенбернар",[2,5,7,8]).

*/*Характеристики*/*

cond(1,"короткошерстная").

cond(2,"длинношерстная").

cond(3,"рост ниже 30 дюймов").

cond(4,"рост ниже 22 дюймов").

cond(5,"низкопосаженный хвост").

```

cond(6,"большие уши").
cond(7,"хороший характер").
cond(8,"вес более 100 фунтов").

do_expert:-
    makewindow(1,7,5,"ЭКСПЕРТНАЯ СИСТЕМА",0,0,25,80),
    show_menu.

repeat.
repeat:-repeat.

/*Вывод меню*/
show_menu:-
    repeat,
    write("*****"),nl,
    write("*****Добро пожаловать!*****"),nl,
    write("*****"),nl,
    write("*****1-консультация*****"),nl,
    write("*****2-список*****"),nl,
    write("*****3-выход*****"),nl,
    write("*****"),nl,
    write("*****Сделайте свой выбор*****"),nl,
    readint(X),
    process(X),fail.

/*Обработка 1 пункта меню "Консультация"*/
process(1):-
    do_consulting,
    readchar(_),
    shiftwindow(1),
    clearwindow.

/*Обработка 2 пункта меню "Вывод списка"*/
process(2):-
    makewindow(2,7,7,"",5,20,12,25),
    topic,
    readchar(_),
    shiftwindow(1),
    clearwindow.

/* Обработка 3 пункта меню "Выход"*/
process(3):-
    removewindow,
    exit.

/*Вывод пород собак*/
topic:-
    rule(X,Y,_),
    write(X," ",Y),
    nl,fail.

topic.

/*Консультация*/
do_consulting:-
    test(1,List),
    rule(_,X,List),
    write("Ваш выбор:" ,X),!.

do_consulting:-
    write("Мне жаль, что не смог Вам помочь.").

/*Тестирование*/
test(9,[]):-!.
test(1,[N|List]):-
    cond(N,Text),
    makewindow(2,7,7,"",5,20,10,35),
    write("Вопрос:-",Text,"?"),nl,
    write("1-да"),nl,
    write("0-нет"),nl,

```

```

        readint(R),R=1,!,test(3,List).
test(1,List):- test(2,List),!.
test(N,[N|List]):-
    cond(N,Text),
    makewindow(2,7,7,"",5,20,10,35),
    write("Вопрос:-",Text,"?"),nl,
    write("1-да"), nl,
    write("0-нет"), nl,
    readint(R),M=N+1,
    R=1,!,test(M,List).
test(N,List):-M=N+1,test(M,List).

```

Задания для самостоятельной работы

Разработать экспертную систему, тему выбрать самостоятельно. Отчет должен содержать следующие пункты:

1. Тема ЭС.
2. Назначение, возможности программы.
3. Разработать структурно-функциональную схему.
4. Определить базу знаний, разработать механизм вывода, интерфейс программы.
5. По каким параметрам программу можно отнести к классу ЭС.

Лабораторная работа 16. Применение нейросетевого прогнозирования при подборе кадров

Основные возможности программы NeuroPro

- Создание нейропроекта;
- Подключение к нейропроекту файла (базы) данных;
- Редактирование файла данных;
- Добавление в нейропроект нейронной сети слоистой архитектуры с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100 (число нейронов для каждого слоя сети может задаваться отдельно);
- Выбор алгоритма обучения, назначение требуемой точности прогноза, настройка параметров нейронной сети;
- Обучение нейронной сети решению задачи предсказания или классификации;
- Тестирование нейронной сети на файле данных;
- Вычисление показателей значимости каждого из входных сигналов для решения задачи, сохранение вычисленных показателей значимости в файле на диске;
- Упрощение нейронной сети;
- Генерация и визуализация вербального описания нейронной сети. Редактирование и сохранение вербального описания нейронной сети в файле на диске;
- Сохранение нейропроекта.

Форматы файлов

В качестве файлов данных (содержащих обучающую выборку для нейронных сетей) используются файлы форматов DBF(форматы пакетов Dbase, FoxBase, FoxPro, Clipper) и DB (Paradox). Возможно чтение и редактирование этих файлов и сохранение измененных файлов на диске. Программа не накладывает ограничений на число записей (строк) в файле данных.

Интерфейс программы

Для того, чтобы сделать активными пункты меню программы, необходимо открыть файл, содержащий входную информацию для создания нейросети. Таким файлом является файл формата dbf. Его можно создать самостоятельно в программе Excel 2003 или открыть уже созданный в формате dbf файл.

Щелкните по кнопке Создать, в новом окне нажмите Открыть файл данных. Откройте файл.

Открытый файл отображается в собственном окне, где предоставляется возможность его редактирования. При подключенном файле данных можно проводить операции создания новых сетей, их обучения, тестирования и упрощения.

Для создания новой нейронной сети необходимо нажать кнопку “Новая сеть” в окне нейропроекта и заполнить окно для создания нейронной сети: подтвердите статусы входных полей, выходного поля, установите Тип поля - количественный, на вкладке Структура сети укажите число слоев нейронов и число нейронов в слое. Нажмите кнопку Создать.

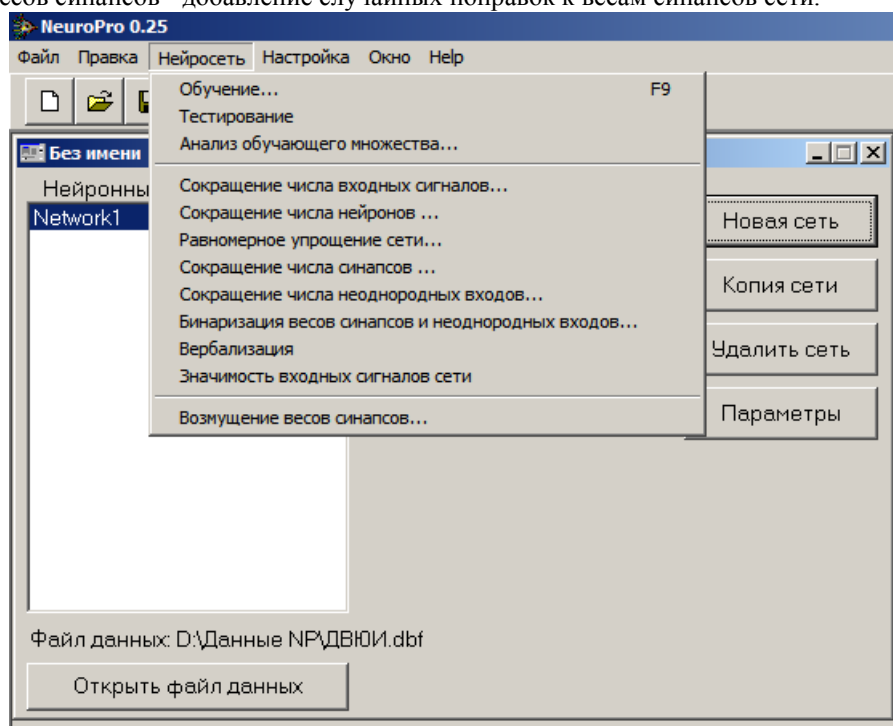
Далее рассмотрим ставшие активными пункты меню программы.

Меню программы содержит следующие пункты, относящиеся к нейронным сетям и работе с ними:

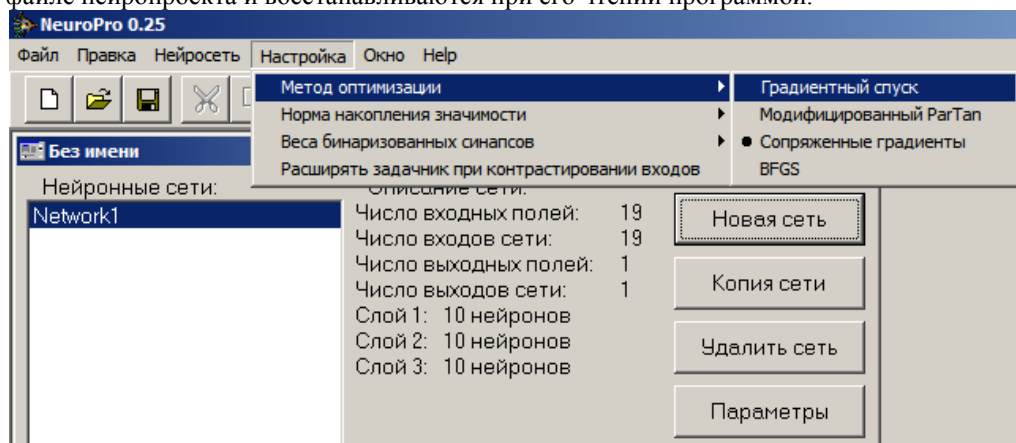
Файл - базовые операции с файлами, включая стандартные опции: создать, открыть, сохранить, сохранить как, выход.

Нейросеть - операции с нейронными сетями. Операция выполняется над активной в данный момент в нейропроекте нейронной сетью и включает в себя следующие опции:

- Обучение - обучение нейронной сети;
- Тестирование - тестирование нейронной сети;
- Сокращение числа входных сигналов - удаление наименее значимых входных сигналов;
- Сокращение числа синапсов - удаление наименее значимых синапсов сети;
- Сокращение числа неоднородных входов - удаление наименее значимых неоднородных входов нейронов сети;
- Равномерное упрощение сети - сокращение числа приходящих на нейроны сети сигналов до заданного пользователем;
- Бинаризация синапсов сети - приведение значений весов синапсов и неоднородных входов нейронов к значениям -1 и 1;
- Вербализация - генерация вербального описания нейронной сети;
- Значимость входов - подсчет и отображение значимости входных сигналов нейронной сети;
- Возмущение весов синапсов - добавление случайных поправок к весам синапсов сети.



Настройка - операции по настройке нейронной сети. Настройки действуют в пределах нейропроекта, сохраняются в файле нейропроекта и восстанавливаются при его чтении программой.



Опции данного пункта меню:

- Метод оптимизации - выбор метода оптимизации для обучения сети. Из трех реализованных в настоящее время в программе методов (градиентный спуск, модифицированный партан-метод (ParTap) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается ParTap.

• **Норма накопления значимости** - выбор нормы накопления градиента при подсчете показателей значимости, иначе говоря, показатель совокупной ошибки сети. При создании нейропроекта автоматически выбирается норма в виде суммы модулей.

Сокращение числа неоднородных входов - удаление наименее значимых неоднородных входов нейронов сети;

Равномерное упрощение сети - сокращение числа приходящих на нейроны сети сигналов до заданного пользователем;

Бинаризация синапсов сети - приведение значений весов синапсов и неоднородных входов нейронов к значениям -1 и 1;

Вербализация - генерация вербального описания нейронной сети;

Значимость входов - подсчет и отображение значимости входных сигналов нейронной сети;

Возмущение весов синапсов - добавление случайных поправок к весам синапсов сети.

Настройка - операции по настройке нейронной сети. Настройки действуют в пределах нейропроекта, сохраняются в файле нейропроекта и восстанавливаются при его чтении программой.

Опции данного пункта меню:

Метод оптимизации - выбор метода оптимизации для обучения сети. Из трех реализованных в настоящее время в программе методов (градиентный спуск, модифицированный партан-метод (PaTан) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается PaTан.

Норма накопления значимости - выбор нормы накопления градиента при подсчете показателей значимости, иначе говоря, показатель совокупной ошибки сети. При создании нейропроекта автоматически выбирается норма в виде суммы модулей.

Постановка задачи

При приеме на службу в органы внутренних дел (ОВД), поступлении в высшее юридическое учебное заведение МВД РФ каждый кандидат проходит обследование с помощью различных методик, направленных на оперативное и удобное обследование психологической сферы индивида (мотивы, познавательные процессы, самооценка, личностные особенности, социально-психологические качества и др.).

Однако данные, полученные в результате обследования, в дальнейшем используются неэффективно. В этой связи возникает необходимость разработки методики, применение которой давало бы возможность сопоставлять результаты тестирования слушателей высших учебных заведений МВД РФ с состоянием их успеваемости, дисциплины, а также эффективностью дальнейшей работы в ОВД и осуществлять поддержку обоснованных решений при приеме лиц на службу в ОВД или на учебу в высшие учебные заведения МВД РФ. Именно такая методика позволит более качественно осуществлять подбор сотрудников в ОВД и слушателей для высших юридических и пожарно-технических учебных заведений МВД РФ.

Для построения нейросетевой модели и ее практического применения использовались результаты тестирования лиц, поступающих на учебу в Дальневосточный юридический институт МВД РФ. Обучающая выборка состоит из результатов тестирования 270 абитуриентов ДВЮИ МВД РФ личностным опросником Кеттелла и оценок состояния их дисциплины, успеваемости, а также оценок эффективности их последующей работы в ОВД. Расшифровка оценочных шкал Кеттелла приведена в табл. 1.

Таблица 1 – Интерпретация оценочных факторов Кеттелла

Факторы	Интерпретация факторов
A	«замкнутость — общительность»
B	«интеллект»
C	«эмоциональная неустойчивость — эмоциональная устойчивость»
E	«подчиненность — доминантность»
F	«сдержанность — экспрессивность»
G	«подверженность чувствам — высокая нормативность поведения»
H	«робость — смелость»
I	«жесткость — чувствительность»
L	«доверчивость — подозрительность»
M	«практичность — развитое воображение»
N	«прямолинейность — дипломатичность»
O	«уверенность в себе — тревожность»
Q1	«консерватизм — радикализм»
Q2	«конформизм — неконформизм»
Q3	«низкий самоконтроль — высокий самоконтроль»
Q4	«расслабленность — напряженность»

Обученные соответствующим образом искусственные нейронные сети (НС) смогут прогнозировать для поступающего в ДВЮИ МВД РФ оценку состояния дисциплины и успеваемости с горизонтом прогнозирования 4 года и эффективность его последующей работы с горизонтом прогнозирования 7 лет. Обучение НС целесообразно производить на нейроимитаторе Neuropro 0.25, который успешно применяется для решения различных нерегулярных задач.

Последовательность выполнения работы

1. Запуск программного нейроимитатора «NEUROPRO 0.25».
«ПУСК» – «ПРОГРАММЫ» – «NEUROPRO» – «NEUROPRO 0.25» (Или ярлык на рабочем столе)
 2. Начало работы в нейроимитаторе.
«ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»
Открыть файл данных ДВЮИ.dbf, содержащий результаты тестирования и экспертные оценки прогнозируемых параметров, необходимые для обучения сети.
Щелкнуть левой кнопкой мыши (л.к.м.) на предыдущем окне (без имени)
Нажать кнопку «НОВАЯ СЕТЬ».
Установить входы и выходы сети:
поле «NOMER» не используется;
поле «DISCIP» не используется;
поле «USPEVA» не используется;
поле «РАБОТА» выходное, тип поля количественный, точность 0,1.
 3. Задать структуру сети:
число слоев – 1;
число нейронов – 16;
характеристика – 0,1.
Нажать кнопку «СОЗДАТЬ».
В верхнем меню нажать «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
Определить минимальное количество нейронов для решения данной задачи:
 4. Нажать «КОПИЯ СЕТИ».
 5. В «СТРУКТУРЕ СЕТИ» изменить количество нейронов.
Далее с п. 3 по п.4 включительно.
- ПРОГНОЗИРОВАНИЕ**
6. Щелкнуть л.к.м. на таблице с данными.
 7. Выбрать следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
Создать новый пункт (№ 271) и заполнить его произвольными числами, в диапазоне от 1 до 10 (перемещение между столбцами кнопкой «ТАВ» или «стрелка вправо»), кроме колонки прогнозируемого параметра (РАБОТА).
Щелкнуть л.к.м. на предыдущем окне (Без имени).
Из верхнего меню выбрать «НЕЙРОСЕТЬ» – «ТЕСТИРОВАНИЕ».
Переместиться в конец таблицы и посмотреть значение прогнозируемого параметра.
Закрыть окно тестирования.
 8. Оценка сетью значимости входных сигналов (в нашем случае оценочных шкал Кеттелла, за каждой из которых скрывается черта характера).
«НЕЙРОСЕТЬ» – «ЗНАЧИМОСТЬ ВХОДНЫХ СИГНАЛОВ».
Выпишите самые важные, с точки зрения сети, оценочные шкалы и закройте окно.
 9. Повторить с п.2, используя в качестве прогнозируемого параметра «ДИСЦИПЛИНУ» и «УСПЕВАЕМОСТЬ» (по очереди).
Поэкспериментировать с упрощением сети «НЕЙРОСЕТЬ» – «УПРОЩЕНИЕ».
 10. Пользуясь полученными диаграммами значимости входных сигналов сформулируйте в качестве вывода, какие качества личности, по Кеттелу, влияют на дисциплину, успеваемость и эффективность работы выпускников высших учебных заведений МВД РФ.
 11. Оформить отчет.

Лабораторная работа 17. Нейросетевое прогнозирование колебаний котировок валют

Курс доллара США к российскому рублю на сегодня 6 февраля 2015, составил **68.6113**.



Постановка задачи

Прогнозирование каких-либо нестабильных экономических показателей сопряжено с множеством трудностей, обусловленных различными факторами. Однако краткосрочный прогноз возможен и эффективно применяется в аналитической работе различных организаций.

1. Составьте в MS Excel таблицу, содержащую котировки валют с 01.01.2015 г. по 06.02.2015 г. (названия столбцов пишите английскими буквами).

Таблица 2 – Официальные Курсы ЦБ РФ (USD,EUR) на январь, февраль 2015 г.

Data	EUR/RUR	USD/RUR
01.01.2015	68,3681	56,2376
02.01.2015	68,3681	56,2376
03.01.2015	68,3681	56,2376
04.01.2015	68,3681	56,2376
05.01.2015	68,3681	56,2376
06.01.2015	68,3681	56,2376
07.01.2015	68,3681	56,2376
08.01.2015	68,3681	56,2376
09.01.2015	68,3681	56,2376
10.01.2015	68,3681	56,2376
11.01.2015	68,3681	56,2376
12.01.2015	68,3681	56,2376
13.01.2015	74,3551	62,7363
14.01.2015	76,7735	64,8425
15.01.2015	77,9629	66,0983
16.01.2015	76,3352	64,8337
17.01.2015	75,8623	65,1738
18.01.2015	75,8623	65,1738
19.01.2015	75,8623	65,1738
20.01.2015	75,174	64,9732
21.01.2015	75,2735	64,9862
22.01.2015	75,8218	65,5558
23.01.2015	75,7724	65,4
24.01.2015	71,9067	63,393
25.01.2015	71,9067	63,393
26.01.2015	71,9067	63,393
27.01.2015	73,5633	65,5937

28.01.2015	76,2922	67,8153
29.01.2015	76,2629	67,1506
30.01.2015	77,569	68,7303
31.01.2015	78,1105	68,9291
01.02.2015	78,1105	68,9291
02.02.2015	78,1105	68,9291
03.02.2015	78,79	69,664
04.02.2015	76,8271	67,7727
05.02.2015	75,0415	65,447
06.02.2015	77,9356	68,6113

Сохраните таблицу в формате Excel (для дальнейшего редактирования) и в формате DBF4 (для работы в нейромимитаторе). Сохранение в формате DBF4 возможно только в Excel 2003.

2. Создайте, обучите, протестируйте, упростите и вербализуйте нейронную сеть.
3. По обученной нейронной сети выполните прогноз курса доллара США с горизонтом прогнозирования 3 дня (с 07.02.2015 по 09.02.2015).
4. Постройте в табличном редакторе EXCEL линию тренда с уравнением в виде полинома 2 степени и прямой) и выполните прогноз вперед. Проведите корреляционный анализ показателей.
5. Сравните результаты прогнозируемых параметров, полученных с использованием нейронных сетей и Excel.
6. Дополните самостоятельно и отредактируйте в MS Excel таблицу, котировками других валют (не менее 5). Курсы валют приведены в приложении. Например, австралийский доллар AUD/RUR, английский фунт стерлингов GBP/RUR, датские кроны 10 DKK/RUR, шведские кроны 10 SEK/RUR, украинские гривны 10 UAN/RUR, турецкие лиры TRY/RUR, швейцарские франки CHF/RUR, японские йены 100 JPY/RUR (23,2498) и др. Последняя колонка – доллар. Сохраните таблицу в формате DBF 4, который поддерживает программа нейромимитатор. Выполните прогноз котировок валют с горизонтом прогнозирования 3 дня.
7. Сделайте выводы и оформите отчет по выполненной лабораторной работе.

Лабораторная работа 18. Нейросетевое распознавание сложных образов (текста)

Часть 1

1. Запустите Microsoft Excel.
2. Оставьте первую строку пустой.
3. Используя цифровую кодировку букв (А-1, Б-2, В-3, Г-4...), запишите друг под другом слева направо три слова САЛЬДО, ДЕБЕТ, КРЕДИТ (код каждой буквы в отдельной ячейке), пронумеровав строки в последнем столбце.
4. Выделите полученную таблицу (в т.ч. и оставленную пустой первую строку) и сохраните в формате DBF
4. Откройте файл через программу NEURO PRO 0.25 («ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»).
5. Нажмите кнопку «НОВАЯ СЕТЬ» и задайте выходным сигналом столбец с нумерацией строк.
6. Установите точность 0,01.
7. Нажмите кнопку «СОЗДАТЬ».
8. В верхнем меню нажмите «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
9. Щелкните л.к.м. на таблице с данными.
10. Выберите следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
11. Создайте новый пункт, написав цифрами любое слово с ошибкой.
12. Спрогнозируйте, используя «Нейросеть» – «Тестирование».

Прогнозируемый параметр должен быть близким к порядковому номеру слова.

Часть 2

1. Запустите Microsoft Excel.
2. Оставьте первую строку пустую.
3. Используя цифровую кодировку букв (А-1, Б-2, В-3, Г-4...), напишите в строку три определения (например, пороговая, линейная, сигмоидальная). В следующей строке напишите другую вариацию понятий, перебрав, таким образом, все возможные варианты (должно получиться 6 строк с разными сочетаниями цифр). Пронумеруйте полученные строки цифрами от 1 до 6.

Таблица 3. – Кодировка алфавита по порядковому номеру букв

А	Б	В	Г	Д	Е	Ё	Ж	З	И
1	2	3	4	5	6	7	8	9	10
Й	К	Л	М	Н	О	П	Р	С	Т
11	12	13	14	15	16	17	18	19	20
У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
21	22	23	24	25	26	27	28	29	30
Э	Ю	Я							
31	32	33							

4. Выделите полученную таблицу (в т.ч. и оставленную пустой первую строку) и сохраните в формате DBF
4. Откройте файл через программу NEURO PRO 0.25 («ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»).
5. Нажмите кнопку «НОВАЯ СЕТЬ» и задайте выходным сигналом столбец с нумерацией строк.
6. Установите точность 0,01.
7. Нажмите кнопку «СОЗДАТЬ».
8. В верхнем меню нажмите «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
9. Щелкните л.к.м. на таблице с данными.
10. Выберите следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
11. Добавьте запись, написав три определения, в произвольном порядке допустив не более 5 ошибок (можно просто вставить 5 нулей). Оставьте последнюю колонку пустой.
12. Спрогнозируйте, используя «Нейросеть/Тестирование».
13. Прогнозируемый параметр должен быть близким к порядковому номеру записи, вариацию которой вы выбрали.
14. Сформулируйте выводы и оформите ход работы на ПК.

4.1.2. Тестирование

Тема 1. Линейное программирование

Тема 2. Теория двойственности

Тема 3. Элементы теории игр

4.1.2.1. Порядок проведения.

Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий. Тестирование проводится по вариантам. В каждом варианте – 10 тестовых заданий. Ниже приведены примерные задания. Полный банк тестовых заданий хранится на кафедре.

4.1.2.2 Критерии оценивания

5 баллов ставится, если обучающийся:

86% правильных ответов и более.

4 баллов ставится, если обучающийся:

От 71% до 85 % правильных ответов.

3 баллов ставится, если обучающийся:

От 56% до 70% правильных ответов.

0--2 баллов ставится, если обучающийся:

55% правильных ответов и менее.

4.1.2.3. Содержание оценочного средства

Темы 1-3

1. Модель – это
 - 1) аналог (образ) оригинала, но построенный средствами и методами
 - 2) отличными от оригинала
 - 3) подобие оригинала
 - 4) копия оригинала
2. Экономико-математическая модель – это
 - 1) математическое представление экономической системы (объектов, задачи, явлений, процессов и т. п.)
 - 2) качественный анализ и интуитивное представление объектов, задач, явлений, 3) процессов экономической

системы и ее параметров

4) эвристическое описание экономической системы (объектов, задачи, явлений, процессов и т. п.)

3. Метод – это

1) подходы, пути и способы постановки и решения той или иной задачи в

2) различных областях человеческой деятельности

3) описание особенностей задачи (проблемы) и условий ее решения

4) требования к условиям решения той или иной задачи

4. Выберите неверное утверждение

1) ЭММ позволяют сделать вывод о поведении объекта в будущем

2) ЭММ позволяют управлять объектом

3) ЭММ позволяют выявить оптимальный способ действия

4) ЭММ позволяют выявить и формально описать связи между переменными, которые характеризуют исследования

5. Задача, включающая целевую функцию f и функции Φ , входящие в ограничения, является задачей линейного программирования, если

1) все Φ и f являются линейными функциями относительно своих аргументов

2) все Φ являются линейными функциями относительно своих аргументов, а функция f – нелинейна

3) функция f является линейной относительно своих аргументов, а функции Φ – нелинейны

4) только часть функций Φ и функция f являются линейными относительно своих аргументов

6. Множество всех допустимых решений системы задачи линейного программирования является

1) выпуклым

2) вогнутым

3) одновременно выпуклым и вогнутым

7. Если задача линейного программирования имеет оптимальное решение, то целевая функция достигает нужного экстремального значения в одной из

1) вершин многоугольника (многогранника) допустимых решений

2) внутренних точек многоугольника (многогранника) допустимых решений

3) точек многоугольника (многогранника) допустимых решений

8. В задачах линейного программирования решаемых симплекс-методом искомые переменные должны быть

1) неотрицательными

2) положительными

3) свободными от ограничений

4) любыми

9. Симплексный метод решения задач линейного программирования включает определение одного из допустимых базисных решений поставленной задачи (опорного плана)

1) определение правила перехода к не худшему решению

2) проверку оптимальности найденного решения

3) определение одного из допустимых базисных решений поставленной задачи

4) (опорного плана), определение правила перехода к не худшему решению, проверка оптимальности найденного решения

10. Если целевая функция и все ограничения выражаются с помощью линейных уравнений, то рассматриваемая задача является задачей

1) динамического программирования

2) линейного программирования

3) целочисленного программирования

4) нелинейного программирования

11. Модель задачи линейного программирования, в которой целевая функция исследуется на максимум и система ограничений задачи является системой уравнений, называется

1) стандартной

2) канонической

3) общей

4) основной

5) нормальной

12. Модель задачи линейного программирования, в которой целевая функция исследуется на максимум и система ограничений задачи является системой неравенств, называется

1) стандартной

2) канонической

3) общей

4) основной

5) нормальной

13. Задача линейного программирования может достигать максимального значения

1) только в одной точке

- 2) в двух точках
- 3) во множестве точек
- 4) в одной или двух точках
- 5) в одной или во множестве точек
- 14. Какой вид имеет целевая функция задачи линейного программирования?

1) $f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n +$

2) $Ax \leq (=, \geq) B$

3) $x \leq 0$

4) $x \geq 0$

15. Выполнение какого условия является признаком оптимальности решения двойственной задачи линейного программирования?

- 1) в симплекс-таблице элементы строки целевой функции прямой задачи положительны
- 2) в симплекс-таблице элементы столбца целевой функции двойственной задачи положительны
- 3) в симплекс-таблице элементы строки целевой функции прямой задачи и элементы столбца целевой функции двойственной задачи положительны

16. Если исходная задача линейного программирования имеет оптимальное решение, то задача двойственная к ней

...

- 1) имеет оптимальное решение
- 2) может не иметь решения
- 3) может не иметь смысла
- 17. Если исходная задача линейного программирования не имеет смысла, то задача двойственная к ней ...

- 1) имеет оптимальное решение
- 2) не имеет решения
- 3) не имеет смысла

18. Если ресурс образует «узкое место производства», то это означает

- 1) ресурс избыточен
- 2) ресурс использован полностью
- 3) двойственная оценка ресурса равна нулю

19. Транспортная задача является задачей Программирования

- 1) динамического
- 2) нелинейного
- 3) линейного
- 4) целочисленного
- 5) параметрического

20. Если в транспортной задаче объем спроса равен объему предложения, то такая задача называется

- 1) замкнутой
- 2) закрытой
- 3) сбалансированной
- 4) открытой
- 5) незамкнутой

21. Если в транспортной задаче объем запасов превышает объем потребностей, в рассмотрение вводят

- 1) фиктивный пункт производства
- 2) фиктивный пункт потребления
- 3) изменения структуры не требуются

22. Свободными членами системы ограничений двойственной задачи являются

- 1) неизвестные исходной задачи
- 2) коэффициенты при неизвестных исходной задачи
- 3) свободные члены исходной задачи
- 4) коэффициенты целевой функции исходной задачи

23. Если исходная ЗЛП была на максимум целевой функции, то двойственная задача будет

- 1) тоже на максимум
- 2) либо на максимум, либо на минимум
- 3) и на максимум, и на минимум
- 4) на минимум

24. Число переменных в двойственной задаче равно числу

- 1) переменных в исходной задаче
- 2) отличных от нуля правых частей исходной задачи
- 3) ограничений исходной задачи
- 4) ненулевых коэффициентов целевой функции исходной задачи

25. План модели транспортной задачи удобнее представлять

- 1) вектором
- 2) матрицей
- 3) числом
- 4) функцией

Ответы:

1.	2	6.	1	11.	2	16.	1	21.	2
2.	1	7.	1	12.	3	17.	3	22.	4
3.	2	8.	1	13.	3	18.	2	23.	4
4.	2	9.	4	14.	1	19.	3	24.	1
5.	1	10.	2	15.	1	20.	3	25.	2

4.1.3. Эссе

Тема 4. Введение в проектирование интеллектуальных систем

Тема 5. Модели представления знаний в интеллектуальных системах

Тема 6. Язык логического программирования Пролог

Тема 7. Экспертные системы

Тема 8. Нейронные сети

4.1.3.1. Порядок проведения.

Обучающиеся пишут на заданную тему сочинение, выражающее размышления и индивидуальную позицию автора по определённому вопросу, допускающему неоднозначное толкование. Оцениваются эрудиция автора по теме работы, логичность, обоснованность, оригинальность выводов.

4.1.3.2 Критерии оценивания

5 баллов ставится, если обучающийся:

Тему полностью раскрыл. Превосходное владение материалом. Высокий уровень самостоятельности, логичности, аргументированности. Превосходный стиль изложения

4 баллов ставится, если обучающийся:

Тему в основном раскрыл. Хорошее владение материалом. Средний уровень самостоятельности, логичности, аргументированности. Хороший стиль изложения.

3 баллов ставится, если обучающийся:

Тему частично раскрыл. Удовлетворительное владение материалом. Низкий уровень самостоятельности, логичности, аргументированности. Удовлетворительный стиль изложения.

0--2 баллов ставится, если обучающийся:

Тему не раскрыл. Неудовлетворительное владение материалом. Недостаточный уровень самостоятельности, логичности, аргументированности. Неудовлетворительный стиль изложения.

4.1.3.3. Содержание оценочного средства

Темы 4-8

1. Искусственный интеллект или что ждет нас в будущем
2. Искусственный интеллект - возможности и ограничения, перспективы и последствия.
3. Искусственный интеллект - фундаментальная угроза для всего человечества.
4. Искусственный интеллект - тот случай, когда нужно быть достаточно дальновидными в вопросах регулирования, иначе может оказаться слишком поздно.
5. Создание искусственного интеллекта может стать последним технологическим достижением человечества, если мы не научимся контролировать риски.
6. Возможно ли создание мыслящего компьютера

4.2. Оценочные средства промежуточной аттестации

По дисциплине предусмотрен зачет в 7 семестре. Зачет проходит по билетам. В каждом билете один теоретический вопрос и одна задача. Зачет проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.

4.2.1. Устный или письменный ответ на вопрос

4.2.1.1. Порядок проведения.

Устный или письменный ответ на вопрос направлен на проверку знаний основных разделов по дисциплине «Основы искусственного интеллекта и исследование операций».

4.2.1.2. Критерии оценивания.

17-20 баллов ставится, если обучающийся:

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.

14-16 баллов ставится, если обучающийся:

Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.

11-13 баллов ставится, если обучающийся:

Тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

0--10 баллов ставится, если обучающийся:

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.

4.2.1.3. Оценочные средства.

Вопросы для устного или письменного ответа

1. Общая постановка задачи оптимизации.
2. Целевая функция. Допустимое множество. Допустимое решение.
3. Оптимальное решение. Оптимальное множество.
4. Постановка задачи математического программирования. Классификация задач математического программирования.
5. Постановка задачи линейного программирования.
6. Стандартная и каноническая формы задачи ЛП.
7. Геометрия задачи ЛП. Выпуклая многогранная область в n R . Проектирование выпуклого многогранника на координатные плоскости.
8. Теорема о проекциях.
9. Теорема о существовании оптимального решения задачи ЛП в случае ограниченности целевой функции.
10. Теорема о достижимости оптимального решения задачи ЛП в угловой точке (в случае ограниченности целевой функции).
11. Графический метод решения задач ЛП при малом числе неизвестных. Линия уровня целевой функции.
12. Алгоритм решения задачи ЛП графическим методом. Сведение задач линейного программирования общего вида к задачам, допускающим решение графическим методом.
13. Симплекс-метод решения задачи ЛП общего вида. Алгоритм решения задачи ЛП симплекс-методом.
14. Геометрическая интерпретация симплекс-алгоритма.
15. Теорема о конечности симплекс-алгоритма.
16. Постановка взаимно двойственных задач ЛП.
17. Симметричные взаимно двойственные задачи.
18. Экономический смысл двойственности. Основное неравенство для двойственных задач.
19. Основная теорема двойственности и ее следствия.
20. Критерий оптимальности. Теорема равновесия.
21. Условия дополняющей нежесткости.
22. Двойственность в экономических задачах.
23. Двойственные цены.
24. Транспортная задача ЛП. Открытая и закрытая модель транспортной задачи.
25. Критерий разрешимости транспортной задачи.
26. Методы построения начального опорного плана транспортной задачи (метод СЗ угла, метод минимального тарифа, метод Фогеля).

4.2.2. Решение задач

4.2.2.1. Порядок проведения.

Предлагаются задачи по дисциплине «Основы искусственного интеллекта и исследование операций».

4.2.2.2. Критерии оценивания.

26-30 баллов ставится, если обучающимся:

Задание выполнено полностью и правильно.

21-25 баллов ставится, если обучающимся:

Задание выполнено полностью, но нет достаточного обоснования. Или при верном решении допущена вычислительная ошибка или недочет, не влияющий на правильную последовательность рассуждений.

17-20 баллов ставится, если обучающимся:

Задание выполнено частично или с фактическими и вычислительными ошибками.

0-16 баллов ставится, если обучающимся:

Задание не выполнено или выполнено с большим количеством фактических и вычислительных ошибок.

4.2.2.3. Оценочные средства.

1. Цех изготавливает изделия А и Б. Расход сырья, его запас и прибыль от реализации каждого изделия указаны в таблице.

Вид сырья	Расход на изделие		Запас
	А	Б	
C_1	48	12	600
C_2	24	21	840
C_3	15	27	1350
Прибыль	12	18	

- Найти план производства изделий, обеспечивающий предприятию максимальную прибыль от их реализации. Решить задачу графическим методом.
2. Для изготовления трех видов рубашек используются нитки, пуговицы и ткань. Запасы ниток, пуговиц и ткани, нормы их расхода на пошив одной рубашки указаны в таблице. Найти максимальную прибыль и оптимальный план выпуска изделий ее обеспечивающий.

	рубашка 1	рубашка 2	рубашка 3	Запасы
нитки (м.)	1	9	3	96
пуговицы (шт.)	20	10	30	640
ткань (м ²)	1	2	2	44
Прибыль (р.)	2	5	4	

3. Поставщики товара - оптовые коммерческие предприятия имеют запасы товаров соответственно в количестве и розничные торговые предприятия - подали заявки на закупку товаров в объемах соответственно. Тарифы перевозок единицы груза с каждого из пунктов поставки в соответствующие пункты потребления заданы в виде матрицы. Найти такой план перевозки груза от поставщиков к потребителям, чтобы совокупные затраты на перевозку были минимальными.

Поставщики \ Потребители	B_1	B_2	B_3	B_4	Запасы товаров, a_j
A_1	7	20	3	15	225
A_2	3	14	10	20	250
A_3	15	25	11	19	125
A_4	11	12	18	6	100
Заявки на закупку товаров, b_i	120	150	110	235	

4. Построить математическую модель задачи:

В банк за кредитом обратилось m клиентов, каждый из которых просит сумму a_i , $i = 1, \dots, m$. От сделки с i -м клиентом банк ожидает получить доход c_i . Банк имеет сумму A , которую он может потратить на кредиты и должен обслуживать не менее n и не более N клиентов. Требуется определить, каким клиентам следует дать кредит, чтобы максимизировать ожидаемую прибыль.

5. Компания производит полки для ванных комнат двух размеров - А и В. Агенты по продаже считают, что в неделю на рынке может быть реализовано до 550 полок. Для каждой полки типа А требуется 2 м² материала, а для полки типа В - 3 м² материала. Компания может получить до 1200 м² материала в неделю. Для изготовления одной полки типа А требуется 12 мин машинного времени, а для изготовления одной полки типа В - 30 мин; машину можно использовать 160 час в неделю. Если прибыль от продажи полок

типа А составляет 3 денежных единицы, а от полок типа В - 4 ден. ед., то сколько полок каждого типа следует выпускать в неделю?

6. Решить задачу линейного программирования симплекс-методом.

$$f = 2X_1 + X_2 - 2X_3 \rightarrow \min$$

$$\begin{cases} X_1 + X_2 - X_3 \geq 8; \\ X_1 - X_2 + 2X_3 \geq 2; \\ -2X_1 - 8X_2 + 3X_3 \geq 1; \\ X_i \geq 0 (i = 1, 2, 3). \end{cases}$$

7. Предприятие производит 3 вида продукции: А1, А2, А3, используя сырьё двух типов. Известны затраты сырья каждого типа на единицу продукции, запасы сырья на планируемый период, а также прибыль от единицы продукции каждого вида.

Сырьё	Затраты сырья на единицу продукции			Запас сырья
	А1	А2	А3	
I	3,5	7	4,2	1400
II	4	5		2000
Прибыль от ед. прод.	1	3	3	

- Сколько изделий каждого вида необходимо произвести, чтобы получить максимум прибыли?
 - Определить статус каждого вида сырья и его удельную ценность.
 - Определить максимальный интервал изменения запасов каждого вида сырья, в пределах которого структура оптимального плана, т.е. номенклатура выпуска, не изменится.
 - Определить количество выпускаемой продукции и прибыль от выпуска при увеличении запаса одного из дефицитных видов сырья до максимально возможной (в пределах данной номенклатуры выпуска) величины.
 - Определить интервалы изменения прибыли от единицы продукции каждого вида, при которых полученный оптимальный план не изменится.
8. Решить задачу линейного программирования симплексным методом:

$$F = -3x_1 + x_2 + 4x_3 \rightarrow \max,$$

$$\begin{cases} -x_2 + x_3 + x_4 = 1 \\ -5x_1 + x_2 + x_3 = 2 \\ -8x_1 + x_2 + 2x_3 - x_5 = 3 \end{cases}$$

$$x_1 \dots x_5 \geq 0.$$

9. Решить задачу линейного программирования симплекс-методом:

$$F = 2x_1 + 3x_2 - x_4 \rightarrow \max,$$

$$2x_1 - x_2 - 2x_4 + x_5 = 16,$$

$$3x_1 + 2x_2 + x_3 - 3x_4 = 18,$$

$$-x_1 + 3x_2 + 4x_4 + x_6 = 24,$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$$

10. Решить задачу симплекс-методом, рассматривая в качестве начального опорного плана, план, приведенный в условии:

$$x_1 + 2x_2 - x_3 + x_4 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 - 2x_3 + 3x_4 = 1, \\ 2x_1 - x_2 - x_3 + 3x_4 = 2, \end{cases}$$

$$x_i \geq 0, i = 1, 2, 3, 4,$$

$$X = (0, 0, 1, 1)$$

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2022

Основная литература:

1. Масленникова, О.Е. Основы искусственного интеллекта : учеб. пособие / О.Е. Масленникова, И.В. Гаврилова. — 3-е изд., стер. — Москва : ФЛИНТА, 2019. — 283 с. - ISBN 978-5-9765-1602-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1034902>
2. Пенькова, Т. Г. Модели и методы искусственного интеллекта : учебное пособие / Т. Г. Пенькова, Ю. В. Вайнштейн. — Красноярск : СФУ, 2019. — 116 с. — ISBN 978-5-7638-4043-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/157579> .
3. Ясницкий, Л. Н. Интеллектуальные системы: учебник / Л. Н. Ясницкий. - Москва : Лаборатория знаний, 2016. - 224 с. - ISBN 978-5-00101-417-1. - Текст: электронный // Лань: электронно-библиотечная система. - URL: <https://e.lanbook.com/reader/book/90254/#1> ..
4. Ржевский, С. В. Исследование операций : учебное пособие / С. В. Ржевский. — Санкт-Петербург : Лань, 2022. — 480 с. — ISBN 978-5-8114-1480-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/213248>
5. Шапкин, А. С. Математические методы и модели исследования операций : учебник / А. С. Шапкин, В. А. Шапкин. — 7-е изд. — Москва : Издательско-торговая корпорация «Дашков и К°», 2019. - 398 с - ISBN 978-5-394-02736-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1091193>.
6. Лемешко, Б. Ю. Теория игр и исследование операций / Лемешко Б.Ю. - Новосибирск :НГТУ, 2013. - 167 с.: ISBN 978-5-7782-2198-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/558878>.

**Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая
перечень программного обеспечения и информационных справочных систем**

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2022

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Office Professional Plus 2010

GIMP

Inkscape

Notepad ++

Python

Lazarus

Электронная библиотечная система «ZNANIUM.COM»

Электронная библиотечная система Издательства «Лань»

Электронная библиотечная система «Консультант студента»