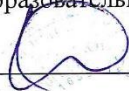


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Умаров Марат Файзуллаевич  
Должность: Директор  
Дата подписания: 18.02.2026 10:48:18  
Уникальный программный ключ:  
48505f11ec15acaa386f5219d3113d727fefda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования  
«Казанский (Приволжский) федеральный университет»  
Елабужский институт (филиал) КФУ



**УТВЕРЖДАЮ**  
Заместитель директора по  
образовательной деятельности  
  
С.Ю. Бахвалов  
« 19 » 05 2025 г.  
МП

**Программа дисциплины (модуля)**  
*Промышленное программирование*

Направление подготовки/специальность: 23.03.01 Технология транспортных процессов

Направленность (профиль) подготовки (специальности): Проектирование и управление интеллектуальными транспортными системами

Квалификация выпускника: бакалавр

Форма обучения: заочная

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

## 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) старший преподаватель, 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:

<b>Шифр компетенции</b>	<b>Расшифровка приобретаемой компетенции</b>
ПК-4.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ПК-4.1.	Знать технологии разработки алгоритмов и программ, пригодных для практического применения
ПК-4.2.	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения
ПК-4.3.	Владеть способностью разрабатывать алгоритмы и программы, пригодные для практического применения

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

принципы и основные элементы объектной модели разработки программного обеспечения.

Должен уметь:

проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID).

Должен владеть:

способностью разрабатывать пользовательские приложения в рамках объектного подхода.

**2. Место дисциплины (модуля) в структуре ОПОП ВО**

Данная дисциплина (модуль) включена в раздел " Б1.В.12.01 Дисциплины (модули)" основной профессиональной образовательной программы 23.03.01 Технология транспортных процессов и относится к части, формируемой участниками образовательных отношений. Осваивается на 5 курсе в 10 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 - 12 часа(ов), в том числе лекции - 4 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 8 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 56 часа(ов).

Контроль (зачёт / экзамен) - 4 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 10 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	С е м е с тр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Основные концепции C++.	10	1	0	2	14
2.	Тема 2. Базовые и продвинутое методы управления ресурсами на C++.	10	1	0	2	14
3.	Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ.	10	1	0	2	14
4.	Тема 4. Разработка и проектирование графического пользовательского интерфейса.	10	1	0	2	14
	Итого		4	0	8	56

#### 4.2 Содержание дисциплины (модуля)

##### Тема 1. Основные концепции C++.

Развитие и стандартизация языка C++. Типы данных. Наследование. Полиморфизм. Инкапсуляция. Специальные функции. Шаблоны. Стандартная библиотека.

##### Тема 2. Базовые и продвинутое методы управления ресурсами на C++

Концепции C++. Основные стратегии копирования-владения. Стратегия запрета копирования. Стратегия исключительного владения. Стратегия глубокого копирования. Стратегия совместного владения. Определение функции обмена состояниями для класса. Удаление промежуточных копий компилятором. Реализация семантики перемещения. Возможные варианты реализации стратегии совместного владения. Стратегия исключительного владения и семантика перемещения. Жизненный цикл ресурса и объекта-владельца ресурса

##### Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ

Шаблон C++. Обзор промышленных программных библиотек. Многопоточное программирование. Управление потоками. Разделение данных между потоками. Синхронизация параллельных операций. Проектирование параллельных программ. Продвинутое управление потоками. Тестирование и отладка многопоточных приложений.

##### Тема 4. Разработка и проектирование графического пользовательского интерфейса

Графический пользовательский интерфейс. Проектирование графического пользовательского интерфейса на C++. Инструментарий разработчика. Разбор существующего проекта с открытым исходным кодом.

#### 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 (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-99бин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

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 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)**

Многопоточное программирование - <http://ders.stml.net/cpp/mtprog/mtprog.html>

Руководство. Управление ресурсами (C++) - <https://docs.microsoft.com/ru-ru/cpp/windows/how-to-copy-resources?view=msvc-160>

Уроки программирования на языке C++ | Ravesli - <https://ravesli.com/uroki-cpp/>

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

Вид работ	Методические рекомендации
лекции	При изучении дисциплины сначала необходимо по каждой теме лекции прочитать рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.
лабораторные работы	При подготовке к лабораторным занятиям необходимо заранее изучить методические рекомендации по его проведению. Обратит внимание на цель работы, на основные вопросы для подготовки к работе, на содержание темы работы. Лабораторное занятие проходит в виде диалога, разбора основных вопросов темы. Также лабораторное занятие может проходить в виде показа презентаций, демонстративного материала (в частности плакатов, слайдов), которые сопровождаются беседой преподавателя со студентами. Студент может сдавать лабораторную работу в виде написания реферата, подготовки слайдов, презентаций и последующей защиты его, либо может написать конспект в тетради, ответив на вопросы по заданной теме. Ответы на вопросы можно сопровождать рисунками, схемами и т.д. с привлечением дополнительной литературы, которую следует указать.
самостоятельная работа	Обучающийся самостоятельно определяет режим своей самостоятельной работы и меру труда, затрачиваемого на овладение знаниями и умениями по дисциплине, выполняет внеаудиторную работу по индивидуальному плану, в зависимости от собственной подготовки, бюджета времени и других условий. Ежедневно обучающийся должен уделять выполнению самостоятельной работы в среднем не менее 3 часов. При выполнении самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания.
зачет	Для контроля усвоения данной дисциплины предусмотрен зачёт, на котором студентам необходимо ответить на вопросы зачётных билетов. При ответе на зачёте необходимо: продумать и четко изложить материал; дать определение основных понятий; дать краткое описание явлений; привести примеры. Ответ следует иллюстрировать схемами, рисунками и графиками.

## 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 № 61 (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) 30 шт. Компьютерный класс: компьютеры Lenovo Thinkcentre m920x 18 шт. Мониторы Thinkcentre TIO24GEN3 18 шт. Интерактивная панель 1 шт. Меловая доска настенная 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 23.03.01 "Технология транспортных процессов" и профилю подготовки "Проектирование и управление интеллектуальными транспортными системами".

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. В.ДВ.12.01 Промышленное программирование*

Направление подготовки: 23.03.01 Технология транспортных процессов  
Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами  
Квалификация выпускника: бакалавр  
Форма обучения: заочное  
Язык обучения: русский  
Год начала обучения по образовательной программе: 2025

## 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. Компьютерная программа
      - 4.1.2.1. Порядок проведения.
      - 4.1.2.2 Критерии оценивания
      - 4.1.2.3. Содержание оценочного средства
  - 4.2. Оценочные средства промежуточной аттестации (зачет)
    - 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. Практическое задание
      - 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
ПК-4 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<p>Знать принципы и основные элементы объектной модели разработки программного обеспечения</p> <p>Уметь проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID)</p> <p>Владеть способностью разрабатывать пользовательские приложения в рамках объектного подхода</p>	<p>Компьютерная программа по темам Тема 1. Основные концепции C++. Тема 2. Базовые и продвинутые методы управления ресурсами на C++. Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ. Тема 4. Разработка и проектирование графического пользовательского интерфейса.</p> <p>Лабораторные работы по темам Тема 1. Основные концепции C++. Тема 2. Базовые и продвинутые методы управления ресурсами на C++. Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ. Тема 4. Разработка и проектирование графического пользовательского интерфейса.</p> <p>Промежуточная аттестация: <i>Зачет</i></p>

## 2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично)	Средний уровень (хорошо)	Низкий уровень (удовлетворительно)	Ниже порогового уровня (неудовлетворительно)
ПК-4	Знает принципы и основные элементы объектной модели разработки программного обеспечения	Знает принципы и основные элементы объектной модели разработки программного обеспечения. Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	Знает некоторые принципы и основные элементы объектной модели разработки программного обеспечения. Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	Не знает принципы и основные элементы объектной модели разработки программного обеспечения
	Умеет проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID)	Умеет проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID). Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	Умеет проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID). Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	Не умеет проектировать и разрабатывать прототипы программных компонентов, учитывая принципы объектного подхода (G4, SOLID)

	Владеет способностью разрабатывать пользовательские приложения в рамках объектного подхода	Владеет способностью разрабатывать пользовательские приложения в рамках объектного подхода. Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	Владеет способностью разрабатывать пользовательские приложения в рамках объектного подхода. Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	Не владеет способностью разрабатывать пользовательские приложения в рамках объектного подхода
--	--	--	--	---

### 3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

#### Текущий контроль:

Лабораторные работы по темам

Тема 1. Основные концепции C++.

Тема 2. Базовые и продвинутое методы управления ресурсами на C++.

Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ.

Тема 4. Разработка и проектирование графического пользовательского интерфейса.

Компьютерная программа по темам

Тема 1. Основные концепции C++.

Тема 2. Базовые и продвинутое методы управления ресурсами на C++.

Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ.

Тема 4. Разработка и проектирование графического пользовательского интерфейса.

Выполнение каждого оценочного средства оценивается по шкале: отлично, хорошо, удовлетворительно, неудовлетворительно.

Общая оценка за текущий контроль представляет собой среднее значение между полученными оценками за все оценочные средства.

#### Промежуточная аттестация - зачет

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, принимаящий 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 практическое задание.

Выполнение каждого задания за промежуточную аттестацию оценивается по шкале: отлично, хорошо, удовлетворительно, неудовлетворительно.

Общая оценка за промежуточную аттестацию представляет собой среднее значение между полученными оценками за все оценочные средства промежуточной аттестации.

В случае невозможности установления среднего значения оценки за промежуточную аттестацию (например, «хорошо» или «отлично»), итоговая оценка выставляется экзаменатором, исходя из принципа справедливости и беспристрастности на основании общего впечатления о качестве и добросовестности освоения обучающимся дисциплины (модуля).

Виды оценок:

Для зачета:

Зачтено

Не зачтено

### 4. Оценочные средства, порядок их применения и критерии оценивания

#### 4.1. Оценочные средства текущего контроля

##### 4.1.1. Лабораторные работы

Тема 1. Основные концепции C++.

Тема 2. Базовые и продвинутое методы управления ресурсами на C++.

Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ.

Тема 4. Разработка и проектирование графического пользовательского интерфейса.

#### 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

##### Оценка «отлично» ставится, если обучающийся:

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

##### Оценка «хорошо» 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.

##### Оценка «удовлетворительно» 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.

##### Оценка «неудовлетворительно» 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

##### Лабораторная работа 1. Умные указатели

###### Задание №1

Если в вашем классе есть умный указатель в качестве элемента вашего класса, то почему вы должны стараться избегать динамического выделения объектов этого класса?

###### Задание №2

Измените следующую программу, заменив обычный указатель на умный указатель `std::unique_ptr`, где это необходимо:

```
1      #include <iostream>
2
3      class Fraction
4      {
5      private:
6          int m_numerator = 0;
7          int m_denominator = 1;
8
9      public:
10         Fraction(int numerator = 0, int denominator = 1) :
11             m_numerator(numerator), m_denominator(denominator)
```

```

12     {
13     }
14
15     friend std::ostream& operator<<(std::ostream& out, const Fraction &f1)
16     {
17         out << f1.m_numerator << "/" << f1.m_denominator;
18         return out;
19     }
20 };
21
22 void printFraction(const Fraction* const ptr)
23 {
24     if (ptr)
25         std::cout << *ptr;
26 }
27
28 int main()
29 {
30     Fraction *ptr = new Fraction(7, 9);
31
32     printFraction(ptr);
33
34     delete ptr;
35
36     return 0;
37 }

```

## Лабораторная работа 2. Управление памятью

Задание.

1. Напишите программу, которая выделяет 1000 блоков памяти по 120 байт каждый и выводит статистику (malloc\_stats). Сколько блоков памяти было выделено в разделе «малых блоков»? Измените программу так, чтобы размер блока рассчитывался как  $i * 1024$ , где  $i$  – номер итерации цикла. Повторите эксперимент и ответьте на тот же вопрос.
2. Напишите программу, которая перехватывая функций malloc и free реализует собственную систему управления динамической памяти (кучу). Память для кучи может выделяться как стандартной функцией malloc, так и с помощью вызова mmap. Программа поддерживает выделение блоков размером в диапазоне от «ГР» до «МР\*ГР», где МР – номер месяца Вашего рождения, ГР – год Вашего рождения. Выделение блоков других размеров недопустимо. Максимальный размер кучи задается параметром командной строки. Блоки выделяются по принципу «первый подходящий». Свободные блоки хранятся в виде одного двунаправленного несортированного списка.

## Лабораторная работа №3. Шаблоны

Задача «Углубление» функции»

Реализовать с использованием метапрограммирования шаблон функции, который позволил бы применять заданную функцию многократно.

```

deerp<f, 3>(10); // То же, что и f(f(f(10)))
deerp<g, 1>(x); // g(x)

```

Задача «Шаблонная функция sum\_all»

Реализовать функцию sum\_all, которая позволяла бы суммировать аргумент практически любого типа, который ей передан.

**Входные данные**

```

vector<int> v1 = { 1, 2, 3 };
vector<double> v2 = { 1, 2, 3 };
vector<string> v3 = { "a", "bc", "def" };
vector<char> v4 = { 'a', 'b', 'c' };

```

**Описание задачи**

sum\_all будет представлять из шаблон функции с частичной специализацией для vector<T>.

## Выходные данные

```
sum_all(5); // 5
sum_all(3.0); // 3.0
sum_all(v1); // 6
sum_all(v2); // 6.0
sum_all(v3); // строка "abcdef"
sum_all(v4); // строка "abc"
```

## Усложнения

1. Добавить частичные специализации для `list<T>`, `forward_list<T>`, `set<T>`, `unordered_set<T>`.
2. Добавить частичные специализации для `map<K,V>` и `unordered_map<K,V>` с суммированием по значениям (V).

## Задача «Оптимизированный `copy`»

Реализовать собственную версию шаблонной функции [copy](#) с ускорением работы для частных случаев.

### Мотивация

Шаблонная функция `std::copy` реализует простой цикл. При копировании, например, массивов байтов (`char` или `unsigned char`) процесс будет происходить побайтово. Однако в силу архитектуры памяти и процессора копирование одного байта требует столько же времени, сколько и копирование машинного слова (4 байта для 32-разрядной архитектуры, 8 байт — для 64-разрядной). Функции побайтового копирования из стандартной библиотеки C (`memcpy`, `memmove`, `strncpy` и т.д.) учитывают этот нюанс, и мы могли бы воспользоваться ими.

### Описание задачи

Возьмите два частных случая:

1. Итераторы являются указателями.
2. Итераторы относятся к `std::vector`.

И при условии, что элемент в принципе допускает побайтовое копирование (для проверки можно воспользоваться [is\\_fundamental](#)), реализуйте вызов `memmove` в соответствующих частных специализациях.

### Тестирование

Проверьте, что копирование работает корректно, а также то, что частные специализации достаточно ограничены, чтобы не использовать `memmove` для более сложных типов данных. Например, для итераторов `std::vector<std::string>` мы не можем использовать `memmove`. Здесь должен сработать общий алгоритм копирования.

## Лабораторная работа 4. Графический интерфейс

### Задание

Создать проект приложения с графическим пользовательским интерфейсом.

### 4.1.2. Компьютерная программа

Тема 1. Основные концепции C++.

Тема 2. Базовые и продвинутые методы управления ресурсами на C++.

Тема 3. Применение шаблонов C++ для проектирования программ. Разработка и проектирование многопоточных программ.

Тема 4. Разработка и проектирование графического пользовательского интерфейса.

#### 4.1.2.1. Порядок проведения.

На основе изученного программного обеспечения построения интеллектуальных информационных систем студентам предлагается написать компьютерную программу по вариантам.

#### 4.1.2.2 Критерии оценивания

##### Оценка «отлично» ставится, если обучающийся:

Код компьютерной программы правильный. Компьютерная программа выполняется без ошибок. Результат компьютерной программы полностью соответствует её целям.

##### Оценка «хорошо» ставится, если обучающийся:

Код компьютерной программы в основном правильный. Компьютерная программа выполняется без ошибок. Результат компьютерной программы в основном соответствует её целям.

##### Оценка «удовлетворительно» ставится, если обучающийся:

При выполнении компьютерной программы возможны ошибки. Код компьютерной программы частично правильный. Результат компьютерной программы частично соответствует её целям.

##### Оценка «неудовлетворительно» ставится, если обучающийся:

Компьютерная программа не выполняется. Код компьютерной программы выполнен с ошибками. Результат программы не соответствует её целям.

#### 4.1.2.3. Содержание оценочного средства

1. Разработка программы управления ИТ-проектами: задача прогнозирования себестоимости ИТ-проекта
2. Программное обеспечение системы резервирования билетов

3. Разработка программы автоматизации работы деканата ВУЗа
4. Разработка программы автоматизации учета изделий на предприятии
5. Разработка программы автоматизации автоматизированного рабочего места операциониста библиотеки
6. Разработка программы автоматизации предприятий автосервиса
7. Разработка программы автоматизации учета кадров на предприятии
8. Разработка программы автоматизации строительной организации
9. Разработка программы автоматизации гостиничного комплекса
10. Разработка программы автоматизации аптеки
11. Разработка программы автоматизации туристической фирмы
12. Программные комплексы поддержки принятия управленческих решений (разработка системы учета договоров и расчетов с субподрядчиками)
13. Разработка программного модуля Складской комплекс
14. Разработка программного обеспечения информационного киоска торговой компании
15. Разработка программного обеспечения, решающего задачу распределения готовой продукции между складами
16. Программирование оценки кредитоспособности физических лиц
17. Программный учёт материально-технических средств на примере предприятия
18. Программирование выдачи справок клиентам строительной компании

#### **4.2. Оценочные средства промежуточной аттестации**

По дисциплине предусмотрен зачёт. Зачёт проходит по билетам. В каждом билете два теоретических вопроса и одно практическое задание. Зачёт проводится в устной / письменной и 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 ответ**

###### **4.2.1.1. Порядок проведения.**

Промежуточная аттестация нацелена на комплексную проверку освоения дисциплины. Обучающийся получает вопрос(ы)/задание(я) и время на подготовку. Промежуточная аттестация 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.

###### **4.2.1.2. Критерии оценивания.**

###### **Оценка «отлично» 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.

###### **Оценка «хорошо» 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.

###### **Оценка «удовлетворительно» 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.

###### **Оценка «неудовлетворительно» 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. Развитие и стандартизация языка C++.
2. Типы данных.
3. Наследование.
4. Полиморфизм.
5. Инкапсуляция.
6. Специальные функции.
7. Шаблоны.
8. Стандартная библиотека.
9. Концепции C++.
10. Основные стратегии копирования-владения.
11. Стратегия запрета копирования.
12. Стратегия исключительного владения.
13. Стратегия глубокого копирования.
14. Стратегия совместного владения.

15. Определение функции обмена состояниями для класса.
16. Удаление промежуточных копий компилятором.
17. Реализация семантики перемещения.
18. Возможные варианты реализации стратегии совместного владения.
19. Стратегия исключительного владения и семантика перемещения.
20. Жизненный цикл ресурса и объекта-владельца ресурса
21. Шаблон C++.
22. Обзор промышленных программных библиотек.
23. Многопоточное программирование.
24. Управление потоками.
25. Разделение данных между потоками.
26. Синхронизация параллельных операций.
27. Проектирование параллельных программ.
28. Продвинутое управление потоками.
29. Тестирование и отладка многопоточных приложений.
30. Графический пользовательский интерфейс.
31. Проектирование графического пользовательского интерфейса на C++.
32. Инструментарий разработчика.
33. Разбор существующего проекта с открытым исходным кодом.

#### 4.2.2. Практическое задание

##### 4.2.2.1. Порядок проведения.

Практические навыки проверяются путём выполнения обучающимися практических заданий в условиях, полностью или частично приближенных к условиям профессиональной деятельности. Проверяется знание теоретического материала, необходимое для правильного совершения необходимых действий, умение выстроить последовательность действий, практическое владение приёмами и методами решения профессиональных задач.

##### 4.2.2.2. Критерии оценивания

**Оценка «отлично» ставится, если обучающимся:**

Задание выполнено полностью и правильно.

**Оценка «хорошо» ставится, если обучающимся:**

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

**Оценка «удовлетворительно» ставится, если обучающимся:**

Задание выполнено частично или с фактическими ошибками.

**Оценка «неудовлетворительно» ставится, если обучающимся:**

Задание не выполнено или выполнено с большим количеством фактических ошибок.

##### 4.2.3.3. Содержание оценочного средства

1) Задание

Используя шаблоны функций и частичную специализацию шаблонов добиться корректного использования пользовательского класса в коллекциях [unordered\\_map](#), [unordered\\_set](#).

Описание

Неупорядоченные коллекции реализованы на базе хеш-таблиц. Для манипулирования элементами (ключами) коллекции необходимо уметь строить хеш от элемента и сравнивать два элемента коллекции. Для встроенных или стандартных типов данных хеш-функция и операция сравнения уже определены. Для пользовательских классов хеш-функцию и сравнение необходимо реализовать самостоятельно.

Пример

```
#include <unordered_set>
```

```
class Point {
```

```
    int x, y;
```

```
};
```

```
//...
```

```
// Point p, p1, p2;
```

```
// хеш от p: p.x^p.y,
```

```
// сравнение p1 и p2: (p1.x==p2.x) && (p1.y==p2.y)
```

```
//...
```

```
std::unordered_set < Point > s;
```

2) Задание. Реализовать обобщённый итерационный алгоритм вычисления квадратного корня (и не только), в котором все детали вынесены в функторы, в виде функции `iterate`.

Потребуется три функтора:

`initial` – функтор, вычисляющий начальное приближение. Для квадратного корня из  $x$  можно взять  $x/2$ .

`nnext` – функтор, строящий следующее приближение. Для квадратного корня:  $(y + x/y)/2$ , где  $x$  – аргумент функции корня, а  $y$  – текущее приближение.

enough – предикат, оценивающий точность. Если он возвращает true, итерационную процедуру можно завершить. Для квадратного корня можно взять  $\text{fabs}(x - y^*y) < 1e-8$ , где  $x$  и  $y$  имеют тот же смысл, что и выше. Таким образом, шаблонная функция `iterate` будет иметь четыре параметра шаблона: тип значения и три функтора.

### 3) Задание

В директории лежат входные текстовые файлы, проименованные следующим образом: `in_<N>.dat`, где  $N$  - натуральное число. Каждый файл состоит из двух строк. В первой строке - число, обозначающее действие, а во второй - числа с плавающей точкой, разделенные пробелом.

Действия могут быть следующими:

1 - сложение

2 - умножение

3 - сумма квадратов

Необходимо написать многопоточное приложение, которое выполнит требуемые действия над числами и сумму результатов запишет в файл `out.dat`. Название рабочей директории передается в виде аргумента рабочей строки. В реализации приветствуется использование полиморфизма и паттернов проектирования.

### 4) Задание

Разработайте многопоточное приложение, выполняющее вычисление произведения матриц  $A (m \times n)$  и  $B (n \times k)$ . Элементы  $c_{ij}$  матрицы произведения  $C = A \times B$  вычисляются параллельно  $p$  однотипными потоками. Если некоторый поток уже вычисляет элемент  $c_{ij}$  матрицы  $C$ , то следующий приступающий к вычислению поток выбирает для расчета элемент  $c_{ij+1}$ , если  $j < k$ , и  $c_{i+1k}$ , если  $j = k$ . Выполнив вычисление элемента матрицы-произведения, поток проверяет, нет ли элемента, который еще не рассчитывается. Если такой элемент есть, то приступает к его расчету. В противном случае отправляет (пользовательское) сообщение о завершении своей работы и приостанавливает своё выполнение. Главный поток, получив сообщения о завершении вычислений от всех потоков, выводит результат на экран и запускает поток, записывающий результат в конец файла-протокола. В каждом потоке должна быть задержка в выполнении вычислений (чтобы дать возможность поработать всем потокам). Синхронизацию потоков между собой организуйте через критическую секцию или мьютекс.

### 5) Задание.

При нажатии кнопки "Start" диалоговое приложение запускает консольное приложение. Последующие нажатия кнопки "Start" должны привести к созданию в консольном приложении  $N$  новых рабочих потоков, где  $N$  - значение из поля с числовым счетчиком. Нажатие кнопки "Stop" должно привести к закрытию последнего созданного рабочего потока в консольном приложении, а в случае отсутствия рабочих потоков - к его завершению. Консольное приложение также должно завершиться и при завершении диалогового. Диалоговое приложение должно отслеживать, закрыл ли пользователь консольное приложение самостоятельно. Следующее после закрытия консоли нажатие "Start" возобновляет рабочий цикл.

После запуска консоли выпадающий список содержит строки "Все потоки" и "Главный поток", по мере создания новых потоков в него добавляются строки, содержащие их номера (разумеется, при удалении потоков удаляются и соответствующие им строки). Корректировка выпадающего списка осуществляется лишь после получения подтверждения от консоли об успешном создании потоков. Взаимодействие между приложениями реализовать с помощью объектов ядра "события" (events). Добавить в приложения передачу информации с помощью файла, отображаемого в память (memory mapped file). Реализовать транспортные функции в виде динамически подключаемой библиотеки с неявной загрузкой. При нажатии кнопки "Send" диалоговое приложение пересылает консоли текст, введенный в поле ввода. Выпадающий список задает поток-адресат. При выборе строки "Все потоки" сообщение отправляется всем потокам, каждый поток считывает информацию независимо. Главный поток выводит полученный текст на стандартный вывод (stdout), рабочие потоки создают текстовые файлы с именами <свой номер>.txt и дописывают в них полученный текст. Файл, отображаемый в память, защищается от совместного использования с помощью объекта синхронизации mutex. Все действия подтверждаются сообщениями от консоли.

**Перечень литературы, необходимой для освоения дисциплины (модуля)**

Направление подготовки: 23.03.01 Технология транспортных процессов  
Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами  
Квалификация выпускника: бакалавр  
Форма обучения: заочное  
Язык обучения: русский  
Год начала обучения по образовательной программе: 2025

**Основная литература:**

1. Агальцов, В. П. Информатика для экономистов : учебник / В. П. Агальцов, В. М. Титов. - Москва : ФОРУМ : ИНФРА-М, 2022. - 448 с. - (Высшее образование). - ISBN 978-5-8199-0274-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1832127>
2. Бобьрь, М. В. Программирование на языке Java. Практический курс : учебное пособие / М.В. Бобьрь. — Москва : ИНФРА-М, 2025. — 189 с. — (Высшее образование). — DOI 10.12737/2160989. - ISBN 978-5-16-020136-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2160989>
- 3 Кузин, А. В. Программирование на языке Си : учебное пособие / А.В. Кузин, Е.В. Чумакова. — Москва : ФОРУМ : ИНФРА-М, 2024. — 143 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-556-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2137197>

**Дополнительная литература**

1. Корнеев, В. И. Программирование графики на C++. Теория и примеры : учебное пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — Москва : ИНФРА-М, 2024. — 517 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/23113. - ISBN 978-5-16-017914-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2111934>
2. Лянг, В. Ф. Программирование в САПР: пространственное моделирование колонного аппарата в среде Autodesk Inventor : учебное пособие / В.Ф. Лянг. — Москва : ИНФРА-М, 2023. — 249 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/991773. - ISBN 978-5-16-014573-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2125021>

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

Направление подготовки: 23.03.01 Технология транспортных процессов

Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами

Квалификация выпускника: бакалавр

Форма обучения: заочное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

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

1. Операционная система Microsoft office professional plus 2010, или Microsoft Windows 7 Профессиональная, или Windows XP (Volume License)
2. Пакет офисного программного обеспечения Microsoft Office 365, или Microsoft office professional plus 2010
3. Adobe Reader XI или Adobe Acrobat Reader DC
4. Браузер Mozilla Firefox
5. Браузер Google Chrome
6. Kaspersky Endpoint Security для Windows
7. Программная система для обнаружения текстовых заимствований в учебных и научных работах. АО «Антиплагиат»
8. Электронная библиотечная система «ZNANIUM.COM»
9. Электронная библиотечная система Издательства «Лань»
10. Электронная библиотечная система «Консультант студента»