

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
Елабужского института ИИ КФУ
И.П. Михайлова
И.П. Михайлова
« 01 » марта 2024 г.
МП

Программа дисциплины (модуля)

ОП.04 Основы алгоритмизации и программирования

Направление подготовки/специальность: 09.02.07 «Информационные системы и программирование»

Квалификация выпускника: Программист

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2022

г.Елабуга, 2024

1. Цели освоения дисциплины

формирование знаний по:

- понятию алгоритмизации, свойств алгоритмов, общих принципов построения алгоритмов, основным алгоритмическим конструкциям;
- эволюции языков программирования, их классификации, понятию системы программирования;
- основным элементам языка, структуре программы, операторам и операциям, управляющим структуры, структуре данных, файлам, классам памяти;
- подпрограммам, составлению библиотек подпрограмм
- объектно-ориентированной модели программирования, основным принципам объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

формирование умений по:

- разработке алгоритмов для конкретных задач;
- использованию программ для графического отображения алгоритмов;
- определению сложностей работы алгоритмов;
- работе в среде программирования;
- оформлению кода программы в соответствии со стандартом кодирования;
- выполнению проверки, отладки кода программы.

2. Место дисциплины в структуре ПССЗ

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования» является общепрофессиональной дисциплиной профессионального цикла в соответствии с ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование».

Осваивается на втором и третьем курсах курса (3, 4 и 5 семестры).

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины формируются компетенции:

| Индекс компетенции | Расшифровка приобретаемой компетенции |
|--------------------|---|
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 02 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности |
| ОК 09 | Использовать информационные технологии в профессиональной деятельности |
| ПК 1.1 | Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием |

В результате освоения дисциплины обучающийся должен **знать**:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования

и переопределения.

- В результате освоения дисциплины обучающийся должен **уметь**:
- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

4. Структура и содержание дисциплины

4.1. Общая трудоемкость дисциплины в часах

Общая трудоемкость дисциплины составляет 188 часов.

Форма промежуточной аттестации по дисциплине: дифференцированный зачет в 4 семестре, экзамен в 5 семестре.

| Разделы и темы дисциплины | | Семестр | Виды и часы аудиторной работы, их трудоемкость (в часах) | | | Самостоятельная работа | Консультации | Текущие формы контроля |
|---------------------------|---|---------|--|----------------------|---------------------|------------------------|--------------|---|
| | | | Лекции | Практические занятия | Лабораторные работы | | | |
| Тема 1 | Теоретические основы алгоритмизации и программирования | 3 | 16 | 0 | 16 | 2 | 1 | Контрольная работа 1* (в форме тестирования) |

| | | | | | | | | |
|--|--|------------|-----------|----------|-----------|----------|----------|---|
| Тема 2 | Основы программирования. Программирование структур данных | 4 | 46 | 0 | 46 | 2 | 1 | Контрольная работа 2* (в форме тестирования) |
| Тема 3 | Объектно-ориентированное программирование | 5 | 24 | 0 | 24 | 2 | 1 | Защита программного приложения |
| <i>Всего: 181</i> | | | 86 | 0 | 86 | 6 | 3 | |
| <i>Дифференцированный зачет</i> | | 0 | | | | | | |
| <i>Экзамен</i> | | 7 | | | | | | |
| <i>Всего по дисциплине</i> | | 188 | | | | | | |

* письменная контрольная точка (выполняется за компьютером)

4.2. Содержание дисциплины

| Наименование разделов и тем | Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся | Объем часов (лек/ лаб.р./самост) | Уровень освоения* |
|--|---|--|-------------------|
| 1 | 2 | 3 | 4 |
| Тема 1 Теоретические основы алгоритмизации и программирования | Содержание учебного материала | 34 (16/16/2) | |
| | 1. Развитие языков программирования. | 16 | 1 |
| | 2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. | | |
| | 3. Жизненный цикл программы. Программа. Программный продукт и его характеристики. | | |
| | 4. Основные этапы решения задач на компьютере. | | |
| | Лабораторные работы: 1. Структурная организация данных. Модели объектов и процессов. 2. Эволюция языков программирования. Правила оформления текстов программ. 3. Функция сложности алгоритма. Методы сортировки и поиска. 4. Итеративные и рекурсивные алгоритмы. 5. Основные определения теории графов. Алгоритмы построения остовного (покрывающего) дерева сети. Алгоритмы нахождения кратчайших путей на графах. 6. Эвристические алгоритмы. Метод ветвей и границ. Задача коммивояжера. 7. Моделирование с использованием генераторов случайных чисел. 8. Машина Тьюринга. Элементы математической логики. | 16 | 2 |
| | Самостоятельная работа обучающегося Подготовка к контрольной работе по темам: Функция сложности алгоритма. Методы сортировки. Методы поиска. Итеративные и рекурсивные алгоритмы. Колдаев, В. Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст: электронный. - URL: https://znanium.com/read?id=378685 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 316-321 | 2 | 3 |
| Тема 2. Типы данных | Содержание учебного материала | 94 (46/46/2) | |
| | 1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы | 46 | 1 |

| | | | |
|--|--|----|---|
| | <p>данных.</p> <p>2. Этапы решения задач на компьютере.</p> <p>3. Основные понятия языка программирования Python</p> <p>4. Введение в Python</p> <p>5. Линейный алгоритм</p> <p>6. Разветвляющийся алгоритм</p> <p>7. Циклический алгоритм</p> <p>8. Работа с кортежами и списками</p> <p>9. Работа со строками</p> <p>10. Обработка вложенных последовательностей</p> <p>11. Работа с функциями. Создание модулей</p> <p>12. Работа с файлами</p> | | |
| | <p>Лабораторные работы</p> <p>1. Алгоритм. Свойства алгоритма. Способы описания алгоритма.</p> <p>2. Этапы решения задач на компьютере.</p> <p>3. Основные понятия языка программирования Python</p> <p>4. Введение в Python</p> <p>5. Линейный алгоритм</p> <p>6. Разветвляющийся алгоритм</p> <p>7. Циклический алгоритм</p> <p>8. Работа с кортежами и списками</p> <p>9. Работа со строками</p> <p>10. Обработка вложенных последовательностей</p> <p>11. Работа с функциями. Создание модулей</p> <p>12. Работа с файлами</p> | 46 | 2 |
| | <p>Самостоятельная работа обучающегося</p> <p>Подготовка к контрольной работе по темам: Разветвляющийся алгоритм; Циклический алгоритм; Работа с кортежами и списками; Работа со строками; Обработка вложенных последовательностей; Работа с функциями; Создание модулей; Работа с файлами.</p> <p>Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 41-241</p> | 2 | 3 |

| | | | |
|--|--|-------------------------|---|
| Тема 3. Объектно-ориентированное программирование | Содержание учебного материала | 50 (24/24/2) | |
| | 1. Понятия объектно-ориентированного программирования | 24 | 1 |
| | 2. Создание классов | | |
| | 3. Создание конструкторов | | |
| | 4. Инкапсуляция | | |
| | 5. Создание свойств | | |
| | 6. Наследование | | |
| | 7. Примеры решения задач | | |
| | 8. Событийно-ориентированное программирование | | |
| | 9. Создание формы и виджетов Кнопка, Текстовое поле, Надпись | | |
| | 10. Создание виджета Флажок | | |
| | 11. Создание виджета Переключатель | | |
| | 12. Примеры решения задач | | |
| | Лабораторные работы | 24 | 2 |
| | 1. Понятия объектно-ориентированного программирования | | |
| 2. Создание классов | | | |
| 3. Создание конструкторов | | | |
| 4. Инкапсуляция | | | |
| 5. Создание свойств | | | |
| 6. Наследование | | | |
| 7. Примеры решения задач | | | |
| 8. Событийно-ориентированное программирование | | | |
| 9. Создание формы и виджетов Кнопка, Текстовое поле, Надпись | | | |
| 10. Создание виджета Флажок | | | |
| 11. Создание виджета Переключатель | | | |
| 12. Примеры решения задач | | | |
| Самостоятельная работа обучающегося | 2 | 3 | |
| <p>Разработка программного приложения по темам: Объектно-ориентированное программирование, событийное программирование.</p> <p>Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 243 – 281</p> | | | |

| | | |
|--|--------------------------------|--|
| Примерная тематика курсовой работы (проекта) | не предусмотрен о | |
| Всего | 178 (86/86/6) | |
| Консультация | 3 | |
| Экзамен | 7 | |
| Всего по дисциплине | 188 | |

*Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);
2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

4.3. Структура и содержание самостоятельной работы студентов

| Темы дисциплины | | Виды самостоятельной работы | Трудоемкость (в часах) | Формы контроля самостоятельной работы |
|-----------------|---|---|------------------------|---|
| 1 | 2 | 3 | 4 | 5 |
| Тема 1 | Теоретические основы алгоритмизации и программирования | <p>Самостоятельная работа обучающегося</p> <p>Подготовка к контрольной работе по темам: Функция сложности алгоритма. Методы сортировки. Методы поиска. Итеративные и рекурсивные алгоритмы. Колдаев, В. Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: https://znanium.com/read?id=378685 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 316-321</p> | 2 | Контрольная работа 1 (в форме тестирования) |
| Тема 2 | Основы программирования. Программирование структур данных | <p>Самостоятельная работа обучающегося</p> <p>Подготовка к контрольной работе по темам: Разветвляющийся алгоритм; Циклический алгоритм; Работа с кортежами и списками; Работа со строками; Обработка вложенных последовательностей; Работа с функциями; Создание модулей; Работа с файлами. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 41-241</p> | 2 | Контрольная работа 2 (в форме тестирования) |
| Тема 3 | Объектно-ориентированное программирование | <p>Самостоятельная работа обучающегося</p> <p>Разработка программного приложения по темам: Объектно-ориентированное программирование, событийное программирование. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва:</p> | 2 | Разработка программного приложения |

| | | | | |
|-----------------------------------|--|--|----------|--|
| | | ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2023). – Режим доступа: по подписке. Стр. 243 – 281 | | |
| <i>Всего по дисциплине</i> | | | 6 | |

5. Образовательные технологии

Освоение дисциплины «Основы алгоритмизации и программирования» предполагает использование как традиционных (лекции, практические занятия с использованием методических материалов), так использованием в учебном процессе активных и интерактивных форм проведения занятий:

работа в малых группах, решение кейсов (анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений), проблемное обучение (стимулирование студентов к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы). Выполнение заданий требует использования не только учебников и пособий, но и информации, содержащейся в Интернете.

На лекциях и практических занятиях используются:

- информационная и презентационная лекция;
- тематические опросы;
- решение задач;
- коллективное выполнение заданий в подгруппах для обобщения тематического теоретического материала в схемах, таблицах;
- выполнение практических заданий с использованием среды программирования разработки PyCharm Community Edition.

Занятия, проводимые в активной и интерактивной формах

| Номер темы | Наименование темы | Форма проведения занятия | Объем в часах |
|---------------------|---|---------------------------------|---------------|
| Тема 1 | Теоретические основы алгоритмизации и программирования | Информационно-проблемная лекция | 16 |
| Тема 2 | Основы программирования. Программирование структур данных | Информационно-проблемная лекция | 46 |
| Тема 3 | Объектно-ориентированное программирование | Информационно-проблемная лекция | 24 |
| Всего по дисциплине | | | 86 |

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

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.

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

7 Таблица соответствия компетенций, критериев оценки их освоения и оценочных средств

| Шифр компетенци | Планируемые результаты | Оценочные средства | Критерии оценивания результатов обучения (баллы) | | | |
|-----------------|------------------------|--------------------|--|---|---|---|
| | | | 2 | 3 | 4 | 5 |

| и | обучения | | | | | |
|----------|---|---|--|--|---|--------------------------------------|
| ОК 01 | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |
| | Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объеме | Демонстрирует высокий уровень умений |
| ОК 02 | Знать эволюцию языков программирования, их классификацию, понятие системы программирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |
| | Уметь оформлять код программы в соответствии со стандартом кодирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объеме | Демонстрирует высокий уровень умений |

| | | | | | | |
|--------|---|---|--|--|---|--------------------------------------|
| | | экзамену 1-14 Практические задания к экзамену (тест)1-20 | | | | |
| ОК 09 | Знать подпрограммы, составление библиотек подпрограмм | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |
| | Уметь работать в среде программирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объеме | Демонстрирует высокий уровень умений |
| ПК 1.1 | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |

| | | | | | |
|---|--|---|---|--|---|
| <p>Уметь оформлять код программы в соответствии со стандартом кодирования</p> | <p>Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20</p> | <p>Не умеет, демонстрирует частичные умения, допуская грубые ошибки</p> | <p>Демонстрирует частичные умения без грубых ошибок</p> | <p>Умеет применять знания на практике в базовом объеме</p> | <p>Демонстрирует высокий уровень умений</p> |
|---|--|---|---|--|---|

8. Методические указания для обучающихся при освоении дисциплины

Работа на лабораторных занятиях предполагает активное участие в дискуссиях и решении задач. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них.

При работе с терминами необходимо обращаться к словарям, в том числе доступным в Интернете, например на сайте <http://dic.academic.ru>.

При подготовке обучающихся по темам используются конспекты лекций и источники основной и дополнительной литературы. Подготовка докладов осуществляется с использованием нормативно-правовых документов и учебников.

Устный опрос по этой теме проводится в форме беседы. Работа на практических занятиях предполагает активное участие в дискуссиях и решении задач. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них.

Решение задач проводится в группе с обсуждением хода решения, применяемых, схем, способов, проверкой результатов и проведением работы над ошибками.

Тестирование проводится после ознакомления с материалом темы. Обучающийся выполняет тестирование, рассчитанное по времени на 40-50 минут, на бумажном носителе. Тест включает в себя задания разного типа: на выбор одного или нескольких правильных ответов, на соответствие, краткий и числовой ответ. Для прохождения теста дается одна попытка. Далее сверяются и обсуждаются результаты с определением правильных ответов.

Промежуточная аттестация по этой дисциплине проводится в форме экзамена. При подготовке к экзамену необходимо опираться, прежде всего, на источники, которые разбирались на лекционных занятиях и на материалы практических занятий. В каждом билете экзамена содержится два вопроса.

9. Учебно-методическое и информационное обеспечение дисциплины

| Основная литература: |
|--|
| Канцедал, С. А. Алгоритмизация и программирование : учебное пособие / С. А. Канцедал. — Москва : ФОРУМ : ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1189320 (дата обращения: 14.10.2023). – Режим доступа: по подписке. |
| Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1735805 (дата обращения: 14.10.2023). – Режим доступа: по подписке. |
| Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1150328 (дата обращения: 14.10.2023). – Режим доступа: по подписке. |
| Дополнительная литература: |
| Ульянова, Н. Д. Основные принципы алгоритмизации : учебно-методическое пособие / Н. Д. Ульянова. — Брянск : Брянский ГАУ, 2020. — 56 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/172114 (дата обращения: 29.10.2023). — Режим доступа: для авториз. пользователей. |
| Ночка, Е. И. Основы алгоритмизации и программирования. Ответы на контрольные вопросы.: Учебник / Ночка Е.И. - Москва :КУРС, НИЦ ИНФРА-М, 2017. - 59 с.: ISBN 978-5-906818-82-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/772548 (дата обращения: 14.10.2023). – Режим доступа: по подписке. |
| Бедердинова, О. И. Основы алгоритмизации и структурного программирования : учебное пособие / О. И. Бедердинова. — Архангельск : САФУ, 2017. — 88 с. — ISBN 978-5-261-01227-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/161718 (дата обращения: 29.10.2023). — Режим доступа: для авториз. пользователей. |
| Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2024. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/2083383 (дата обращения: 01.11.2023). – Режим доступа: по подписке. |
| Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1913856 (дата обращения: 01.11.2023). – Режим доступа: по подписке. |

Быкадорова, Е. А. Программирование. Практикум : учебное пособие для спо / Е. А. Быкадорова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 60 с. — ISBN 978-5-8114-8952-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/185905> (дата обращения: 01.11.2023). — Режим доступа: для

авториз. пользователей.

10. Материально-техническое и программное обеспечение дисциплины
Освоение дисциплины ОП.04 Основы алгоритмизации и программирования предполагает использование следующего материально-технического обеспечения:

| Наименование дисциплины | Наименование кабинета, перечень оборудования |
|--|--|
| ОП.04 Основы алгоритмизации и программирования | Лаборатория «Программирования и баз данных» Автоматизированные рабочие места обучающихся: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), автоматизированное рабочее место преподавателя: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, выход в интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus. Сервер в лаборатории (8-ми ядерный процессор Intel Core i7, тактовая частота 3.8ГГц, оперативная память 16гб, жёсткий диск 1ТБ, программное обеспечение: Virtual box); Программное обеспечение: EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, MicrosoftVisioProfessional, MicrosoftVisualStudio, IntelliJIDEA |

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен обучающимся. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов среднего профессионального образования нового поколения.

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе Издательства "Лань", доступ к которой предоставлен обучающимся. ЭБС Издательства "Лань" включает в себя электронные версии книг издательства "Лань" и других ведущих издательств учебной литературы, а также электронные версии периодических изданий по естественным, техническим и гуманитарным наукам. ЭБС Издательства "Лань" обеспечивает доступ к научной, учебной литературе и научным периодическим изданиям.

11. Методы обучения для обучающихся инвалидов и лиц с ограниченными возможностями здоровья

В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими обучающимися, создании комфортного психологического климата в студенческой группе.

Условия обучения инвалидов и лиц с ограниченными возможностями здоровья:

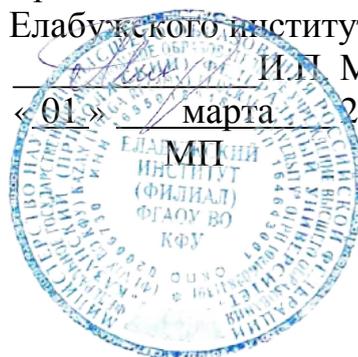
- учебные аудитории, в которых проводятся занятия со студентами с нарушениями слуха, оборудованы мультимедийной системой (ПК и проектор), компьютерные тифлотехнологии базируются на комплексе аппаратных и программных средств, обеспечивающих преобразование компьютерной информации доступные для слабовидящих формы (укрупненный текст);
- в образовательном процессе используются социально-активные и рефлексивные методы обучения: кейс-метод, метод проектов, исследовательский метод, дискуссии в форме круглого стола, конференции, метод мозгового штурма.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ИИ КФУ
И.П. Михайлова
« 01 » марта 20 24 г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОП.04 Основы алгоритмизации и программирования
(наименование дисциплины)

09.02.07 Информационные системы и программирование
(код и наименование специальности)

Программист
(квалификация выпускника)

**Паспорт
фонда оценочных средств по дисциплине
ОП.04 Основы алгоритмизации и программирования**

| Индекс компетенции | Расшифровка компетенции | Показатель формирования компетенции для данной дисциплины | Оценочные средства |
|--------------------|---|---|---|
| 1 | 2 | 3 | 4 |
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 |
| ОК 02 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности | Знать эволюцию языков программирования, их классификацию, понятие системы программирования Уметь оформлять код программы в соответствии со стандартом кодирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 |
| ОК 09 | Использовать информационные технологии в профессиональной деятельности | Знать подпрограммы, составление библиотек подпрограмм Уметь работать в среде программирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 |
| ПК 1.1 | Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции Уметь оформлять код программы в соответствии со стандартом кодирования | Контрольные работы 1-2 (в форме тестирования) Вопросы к дифференцированному зачету 1-20 Вопросы к экзамену 1-14 Практические задания к экзамену (тест)1-20 |

Тема 1 Теоретические основы алгоритмизации и программирования

**Комплект заданий для контрольной работы 1
вариант**

1

**ОП.4 «Основы алгоритмизации и программирования»
(ОК 01, ОК 02, ОК 09, ПК 1.1)**

1. Алгоритм — это:

- а) протокол вычислительной сети
- б) описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов
- в) правила выполнения определенных действий

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

2. Алгоритм называется линейным, если:

- а) ход его выполнения зависит от истинности тех или иных условий
- б) представлен в табличной форме
- в) его команды выполняются в порядке следования друг за другом

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

3. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач:

- а) определенность
- б) массовость
- в) понятность.

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

4. Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа:

- а) паскаль
- б) ассемблер
- в) компилятор

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

5. Как называется набор однотипных данных, имеющий общее для всех своих элементов имя:

- а) множество
- б) массив
- в) запись

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

6. Появление алгоритмов связывают с зарождением:

- а) астрономии
- б) физики
- в) математики

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

7. Какой язык программирования был создан в 1979 году и назван в честь первого в мире программиста:

- а) Паскаль
- б) Фортран
- в) Ада

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

8. Алгоритм называют циклическим, если:

- а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- б) ход его выполнения зависит от истинности тех или иных условий
- в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

9. Свойство алгоритма, заключающееся в том, что один и тот же алгоритм можно использовать с разными исходными данными, называется:

- а) дискретность
- б) массовость
- в) детерминированность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

10. Суть такого свойства алгоритма как результативность заключается в том, что:

- а) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд
- б) алгоритм должен быть разбит на последовательность отдельных шагов
- в) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

11. Какой тип алгоритмической структуры необходимо применить, если последовательность команд выполняется или не выполняется в зависимости от условия:

- а) цикл
- б) ветвление
- в) линейный

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

12. Программа — это:

- а) система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи
- б) указание на выполнение действий из заданного набора
- в) область внешней памяти для хранения текстовых, числовых данных и другой информации
- г) последовательность команд, реализующая алгоритм решения задачи.

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

13. Программа-компилятор выполняет:

- а) переводит исходный текст в машинный код
- б) записывает машинный код в форме загрузочного файла.
- в) формирует текстовый файл

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

14. Последовательность действий, многократно повторяющаяся в процессе выполнения цикла получила название

- а) тело цикла
- б) мозг цикла
- в) действие цикла

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

15. Языком высокого уровня является:

- а) Ассемблер
- б) Фортран
- в) Макроассемблер

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

16. Определите, как называется процесс перестановки элементов массива с целью упорядочивания их в соответствии с каким-либо критерием:

- а) поиск
- б) перебор
- в) сортировка

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

17. Для чего используют программы-ассемблеры:

- а) для перевода программ в машинные коды
- б) для обеспечения бесперебойной работы программ
- в) для создания программ

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

18. Какой язык называется машинно-ориентированным:

- а) язык, в основу которого заложены принципы объектно-ориентированного программирования
- б) любой универсальный язык программирования
- в) язык, определяющийся набором команд конкретного процессора

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

19. Язык Паскаль был разработан в этом году:

- а) 1970
- б) 1980
- в) 1990

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

20. Алгоритм называют линейным, если:

- а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- б) ход его выполнения зависит от истинности тех или иных условий
- в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

Ключ:

| | | | | | | | | | | | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Номер вопроса | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Правильный ответ | б | в | б | в | б | в | в | а | б | в | б | г | б | а | б | в | а | в | а | в |

1. Как называются формальные языки, созданные для разработки программ:

- а) языки высокого уровня
- б) популярные языки
- в) алгоритмические языки

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов(ОК 01, ОК 02, ОК 09, ПК 1.1)

2. Свойство алгоритма, означающее, что путь решения задачи разделен на отдельные шаги:

- а) массовость
- б) конечность
- в) дискретность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

3. К какому уровню языков относятся языки ассемблера:

- а) среднего уровня
- б) низкого уровня
- в) высокого уровня

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

4. Определите, если число повторений цикла известно и задано наибольшее допустимое значение n, то лучше использовать:

- а) цикл с предусловием
- б) цикл с постусловием
- в) цикл со счетчиком

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

5. Какой язык программирования, созданный в 1957 году, является одним из первых алгоритмических языков и до сих пор применяется для научных вычислений:

- а) Паскаль
- б) Фортран
- в) Ада

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

6. Свойство алгоритма, заключающееся в том, что алгоритм должен состоять из конкретных действий, следующих в определенном порядке, называется:

- а) массовость
- б) детерминированность
- в) дискретность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

7. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса однотипных задач:

- а) доступность
- б) массовость
- в) понятность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

8. Что предусматривает алгоритм структуры «ветвление»:

- а) выбор условий
- б) выбор алгоритмов
- в) выбор команд

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

9. Алгоритм называют линейным, если:

- а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- б) ход его выполнения зависит от истинности тех или иных условий
- в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

10. Язык Си разработан Деннисом Ритчи в этом году:

- а) 1982
- б) 1972
- в) 1985

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

11. Свойствами алгоритма являются:

- а) информативность
- б) массовость
- в) оперативность
- г) определенность
- д) дискретность
- е) цикличность
- ж) результативность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

12. Числовые данные могут быть представлены как:

- а) целые
- б) с фиксированной запятой
- в) в виде строк
- г) с плавающей запятой

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

13. Выберите, какой метод применяется для поиска в упорядоченных массивах:

- а) бинарный поиск
- б) прямой выбор
- в) прямой обмен

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

14. Алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз:

- а) одиночный алгоритм
- б) линейный алгоритм
- в) не повторяющийся алгоритм

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

15. Алгоритм:

- а) последовательность действий, которая приводит к решению задачи
- б) набор команд для компьютера
- в) ориентированный граф, указывающий порядок выполнения команд

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

16. Какое свойство алгоритма определяет завершение каждого действия в отдельности и алгоритма в целом за конечное число шагов:

- а) детерминированность
- б) дискретность
- в) конечность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

17. $x:=y*x$ является фрагментом такого алгоритма:

- а) циклического
- б) линейного
- в) разветвляющего

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

18. Суть такого свойства алгоритма как дискретность заключается в том, что:

- а) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд
- б) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату
- в) алгоритм должен быть разбит на последовательность отдельных шагов

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

19. Какая из представленных ниже команд присваивания составлена правильно:

- а) $A:=B$
- б) $A=B+1$
- в) $A=B$

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

20. Какое название носит алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» либо нет) предусмотрен выбор одной из двух последовательностей действий:

- а) цикличность
- б) повторение
- в) ветвление

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

Ключ:

| | | | | | | | | | | | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|----|------------|---------|----|----|----|----|----|----|----|----|
| Номер вопроса | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Правильный ответ | в | в | б | б | б | в | б | в | в | б | б, г, д, ж | а, б, г | а | б | а | в | б | в | а | в |

Тема 2. Основы программирования. Программирование структур данных

Комплект заданий для контрольной работы 2

1 вариант

ОП.4 «Основы алгоритмизации и программирования»

(ОК 01, ОК 02, ОК 09, ПК 1.1)

1. Название алгоритмической конструкции, отображающей естественный, последовательный порядок действий:

- а) ветвление
- б) следование
- в) значение

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

2. Название алгоритмической конструкции, в которой в зависимости от результата проверки условия предусмотрен выбор одной из двух последовательностей действий:

- а) ветвление
- б) следование
- в) указание

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

3. Различают столько типов циклов:

- а) 4
- б) 2
- в) 3

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

4. Операторы цикла можно разделить:

- а) для программирования циклов с переменной цикла и параметрами
- б) для программирования циклов с постоянной цикла и параметрами
- в) для программирования циклов без явно выраженной постоянной цикла

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

5. Что делает метод `normalize()` DOM-объекта:

- а) готовит XML для красивого вывода
- б) сводит воедино все идущие подряд текстовые узлы
- в) исправляет XML, добавляя пропущенные теги

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

6. Для чего применяется метод `nextset()` объекта-курсора:

- а) для перехода к следующему набору записей результата запроса
- б) для перехода к следующей записи результата запроса
- в) для получения следующего набора записей результата запроса

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

7. Какой из перечисленных обработчиков `mod_python` выполняется раньше других:

- а) `PythonPostReadRequestHandler`
- б) `PythonHandler`
- в) `PythonFixupHandler`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

8. Что такое регулярное выражение:

- а) шаблон, описывающий множество строк
- б) синтаксически правильное выражение на языке Python
- в) шаблон для поиска файлов в каталоге

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

9. Какое значение `threadsafety` соответствует ситуации, когда потоки могут одновременно использовать как DB-API 2.0 совместимый модуль, так и соединения, получаемые на основе этого модуля:

- а) 1
- б) 0
- в) 2

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

10. Экземпляры какого класса сочетают замок и средство коммуникации между потоками:

- а) Event
- б) Lock
- в) Condition

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

11. Как средствами самого Python определить имена формальных аргументов функции `func()`, если известно, что функция написана на Python:

- а) `func.func_globals`
- б) `inspect.getargspec(func)`
- в) `func.func_locals`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

12. Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

- а) `items()`
- б) `get_payload()`
- в) `is_multipart()`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

13. Что включает в себя Zope:

- а) СУБД общего назначения
- б) поддержку CGI-сценариев
- в) собственный web-сервер

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

14. К какому уровню модели взаимодействия открытых систем относится протокол FTP:

- а) приложений
- б) представления
- в) сетевому

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

15. Какой модуль стандартной библиотеки Python позволяет работать с WWW на более низком уровне:

- а) `httplib`
- б) `urlparse`
- в) `urllib2`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

16. Какое из приведенных ниже регулярных выражений некорректно:

- а) `a+b++`
- б) `(?P(ac))`
- в) `(a+b+)+`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

17. Какие новые имена появятся в текущем модуле после выполнения следующего кода:
`import sre as re`

`from re import compile`

- а) имена `sre`, `re` и `compile`
- б) только имена `re` и `compile`
- в) только имена `sre` и `compile`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

18. Каким образом в модуле `poplib` представлен сеанс работы с POP3-сервером:

- а) набор функций
- б) экземпляр класса POP3
- в) список кортежей

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

19. Какие кодировки исходного текста программы поддерживает интерпретатор Python:

- а) большинство кодировок, распространенных сегодня
- б) ASCII, Latin-1, UTF-8
- в) ASCII

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

20. Какая встроенная функция Python лучше всего подходит для цепочечных вычислений (в частности, вычисления значения многочлена по схеме Горнера):

- а) `chain()`
- б) `map()`
- в) `reduce()`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

Ключ:

| | | | | | | | | | | | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Номер вопроса | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Правильный ответ | б | а | в | а | б | а | а | а | в | в | б | в | в | а | а | а | б | б | а | в |

**Комплект заданий для контрольной работы 2
2 вариант**

**ОП.4 «Основы алгоритмизации и программирования»
(ОК 01, ОК 02, ОК 09, ПК 1.1)**

1. Название алгоритма, содержащего конструкцию ветвления:

- а) разветвляющийся алгоритм
- б) циклический алгоритм
- в) практический алгоритм

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

2. Какая из представленных ниже алгоритмических конструкций, представляет собой последовательность многократно выполняемых действий:

- а) повторение
- б) ветвление
- в) следование

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. ((ОК 01, ОК 02, ОК 09, ПК 1.1)

3. Для записи условий используется:

- а) операции повторения
- б) операции цикличности
- в) операции сравнения

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

4. Операторы цикла можно разделить:

- а) для программирования циклов без явно выраженной постоянной цикла
- б) для программирования циклов с явно выраженной переменной цикла
- в) для программирования циклов без явно выраженной переменной цикла

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

5. Скрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс, называется:

- а) абстракцией
- б) инкапсуляцией
- в) агрегацией

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

6. Как перевести Unicode-строку u в кодировку koi8-r:

- а) u.decode('koi8-r')
- б) u.encode('koi8-r')
- в) string.decode(u, 'koi8-r')

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов (ОК 01, ОК 02, ОК 09, ПК 1.1)

7. Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

- а) items()
- б) get_main_type()
- в) get_payload()

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

8. Можно ли в XML использовать собственные теги:

- а) нельзя
- б) можно
- в) можно, если указаны пространства имен

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

9. Что включает в себя Zope:

- а) поддержку CGI-сценариев
- б) поддержку сценариев DTML
- в) СУБД общего назначения

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

10. Для чего нужны функции модуля gettext:

- а) для получения текста от пользователя
- б) для обеспечения интернационализации программы
- в) для чтения строки со стандартного ввода

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

11. Что будет получено в результате вычисления следующего выражения:

(0 < 5 <= 3) and (0 / 0):

- а) True (или 1)
- б) False (или 0)
- в) синтаксическая ошибка

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

12. Зачем в XML пространства имен:

- а) пространства имен позволяют указывать опции для приложения, обрабатывающего XML
- б) пространства имен позволяют включать однотипные XML-документы друг в друга
- в) для сочетания в одном документе XML с различными DTD

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

13. Дан массив >>> c = array([[1,2], [2,3], [4,5]])

Чему равен срез c[:,1]:

- а) array([1, 2, 4])
- б) array([2, 3])
- в) array([2, 3, 5])

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

14. В каком модуле нужно искать функции, помогающие тестировать программу:

- а) dictutils
- б) profile
- в) unittest

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

15. Чему будет равен результат выполнения:

`urlparse.urlsplit(«http://google.com/search?q=Python#1»):`

- а) ('http', 'google.com', '/search', », 'q=Python', '1')
- б) ('http://', 'google.com/', 'search?', 'q=Python#', '1')
- в) ('http', 'google.com', '/search', 'q=Python', '1')

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

16. Как используется строка Main Heading в следующем примере:

Main Heading

- а) в качестве заглушки, чтобы в средствах визуального форматирования было видно, что форматировается
- б) в качестве имени переменной для макроподстановки
- в) содержит значение, на которое можно сослаться в другом месте документа

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

17. Алгоритмы, отдельные действия в которых повторяются несколько раз, называются:

- а) функциями циклической структуры
- б) алгоритмами циклической структуры
- в) последовательностью циклической структуры

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

18. Цикл заканчивается, когда параметр:

- а) пропадает
- б) исчерпал себя
- в) принимает определенное значение

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

19. Циклическая структура, в которой число повторений цикла заранее неизвестно, а определяется только в процессе выполнения алгоритма, называется:

- а) интеративной
- б) итеративной
- в) интерактивной

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

20. При всем многообразии алгоритмов решения задач в них можно выделить ... основных вида вычислительных процессов:

- а) 5
- б) 3
- в) 7

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01, ОК 02, ОК 09, ПК 1.1)

Ключ:

| Номер вопроса | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Правильный ответ | а | а | в | в | б | б | б | б | б | б | б | в | в | в | в | а | б | в | б | б |

Тема 3. Объектно-ориентированное программирование

ОП.4 «Основы алгоритмизации и программирования»

Защита программного приложения (ОК 01, ОК 02, ОК 09, ПК 1.1)

1. Разработайте тестовую GUI-программу на тему: Российские кинофильмы.
2. Разработайте тестовую GUI-программу на тему: Спорт.
3. Разработайте тестовую GUI-программу на тему: Автомобили.
4. Разработайте тестовую GUI-программу на тему: Российская эстрада.
5. Разработайте тестовую GUI-программу на тему: Программирование.
6. Разработайте тестовую GUI-программу на тему: Животные.
7. Разработайте тестовую GUI-программу на тему: Города России.
8. Разработайте тестовую GUI-программу на тему: Достопримечательности моего города.

9. Разработайте тестовую GUI-программу на тему: Кулинария.
10. Разработайте тестовую GUI-программу на тему: Мотоциклы.
11. Разработайте тестовую GUI-программу на тему: Живопись.
12. Разработайте тестовую GUI-программу на тему: Книги.
13. Разработайте тестовую GUI-программу на тему: Мультфильмы.
14. Разработайте тестовую GUI-программу на тему: Зарубежное кино.
15. Разработайте тестовую GUI-программу на тему: Зарубежная эстрада.
16. Разработайте тестовую GUI-программу на тему: Военная техника.
17. Разработайте тестовую GUI-программу на тему: Классическая музыка.
18. Разработайте тестовую GUI-программу на тему: Домашние животные.
19. Разработайте тестовую GUI-программу на тему: Информатика.
20. Разработайте тестовую GUI-программу на тему: Моря и океаны.
21. Разработайте тестовую GUI-программу на тему: Великие пустыни мира.
22. Разработайте тестовую GUI-программу на тему: Герои олимпиад.
23. Разработайте тестовую GUI-программу на тему: История России.
24. Разработайте тестовую GUI-программу на тему: Русские поэты.
25. Разработайте тестовую GUI-программу на тему: Дикие животные.
26. Разработайте тестовую GUI-программу на тему: Российские киноактеры.
27. Разработайте тестовую GUI-программу на тему: Интернет.
28. Разработайте тестовую GUI-программу на тему: Русские народные сказки.
29. Разработайте тестовую GUI-программу на тему: Легенды российского балета.
30. Разработайте тестовую GUI-программу на тему: Оргтехника.

Критерии оценки:

| Компетенции | Планируемые результаты обучения | Критерии оценивания результатов обучения (баллы) | | | |
|-------------|---|--|---|---|---|
| | | «неудовлетворительно» | «удовлетворительно» | «хорошо» | «отлично» |
| | | [критерии выставления оценки «неудовлетворительно»] | [критерии выставления оценки «удовлетворительно»] | [критерии выставления оценки «хорошо»] | [критерии выставления оценки «отлично»] |
| ОК 01 | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |
| | Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объеме | Демонстрирует высокий уровень умений |
| ОК 02 | Знать эволюцию языков программирования, их классификацию, | Не знает, допускает грубые | Демонстрирует частичные | Знает достаточно в базовом | Демонстрирует высокий |

| | | | | | |
|-------|--|--|--|---|--------------------------------------|
| | понятие системы программирования | ошибки | знания без грубых ошибок | объёме | уровень знаний |
| | Уметь оформлять код программы в соответствии со стандартом кодирования | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объёме | Демонстрирует высокий уровень умений |
| ОК 09 | Знать подпрограммы, составление библиотек подпрограмм | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объёме | Демонстрирует высокий уровень знаний |
| | Уметь работать в среде программирования | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует Ует частичные умения без грубых ошибок | Умеет Применять знания на практике в базовом объёме | Демонстрирует высокий уровень умений |

| | | | | | |
|--------|---|--|--|---|--------------------------------------|
| ПК 1.1 | Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции | Не знает, допускает грубые ошибки | Демонстрирует частичные знания без грубых ошибок | Знает достаточно в базовом объеме | Демонстрирует высокий уровень знаний |
| | Уметь оформлять код программы в соответствии со стандартом кодирования | Не умеет, демонстрирует частичные умения, допуская грубые ошибки | Демонстрирует частичные умения без грубых ошибок | Умеет применять знания на практике в базовом объеме | Демонстрирует высокий уровень умений |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

**Вопросы к дифференцированному зачёту (ОК 01, ОК 02, ОК 09,
ПК 1.1)**

1. Типы данных. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Данные, с которыми работает программа, хранятся в оперативной памяти. Естественно, что компилятору необходимо точно знать, сколько места они занимают, как именно закодированы и какие действия с ними можно выполнять. Все это задается при описании данных с помощью типа.

Тип данных однозначно определяет:

внутреннее представление данных, а, следовательно, и множество их возможных значений; допустимые действия над данными (операции и функции).

2. Простые типы данных. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

У переменной простого типа под ключевым словом скрывается одно значение (зачастую читаемое как число) и к нему есть прямой доступ. Наиболее известные простые типы это: целое число со знаком, целое число без знака, дробное число (с запятой), символ, логическое значение. В различных языках они могут немного отличаться.

3. Производные типы данных. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Это типы данных, которые не зависят от реализации и могут быть реализованы тем или иным способом. Они обычно создаются с помощью комбинации основных или встроенных типов данных и связанных с ними операций. Например:

список;

массив;

стек;

очередь.

4. Структурированные типы данных. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

В случае структурированных типов под одним ключевым словом группируются несколько совместных значений, таких, например, как координаты точки или имя и фамилия человека. В таком виде набор данных разом легче передавать. В то же время использовать или изменять данные внутри структуры приходится по одному.

5. Этапы решения задач на компьютере. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Работа по решению любой задачи с использованием компьютера делится на следующие этапы.

Постановка задачи.

Формализация задачи.

Построение алгоритма.

Составление программы на языке программирования.

Отладка и тестирование программы.

Проведение расчётов и анализ полученных результатов.

6. Основные понятия языка программирования Python (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Python это язык программирования общего назначения, нацеленный в первую очередь на повышение продуктивности самого программиста, нежели кода, который он пишет. На Python

можно написать практически что угодно (веб-/настольные приложения, игры, скрипты по автоматизации, комплексные системы расчёта, системы управления жизнеобеспечением и многое другое) без ощутимых проблем. Имеет низкий порог вхождения, а код во многом лаконичный и понятный даже тому, кто никогда на нём не писал. За счёт простоты кода, дальнейшее сопровождение программ, написанных на Python, становится легче и приятнее.

Основные особенности языка программирования Python:

Скриптовый язык. Код программ определяется в виде скриптов.

Поддержка самых различных парадигм программирования, в том числе объектно-ориентированной и функциональной парадигм.

Интерпретация программ. Для работы со скриптами необходим интерпретатор, который запускает и выполняет скрипт.

Выполнение программы на Python выглядит следующим образом. Сначала мы пишем в текстовом редакторе скрипт с набором выражений на данном языке программирования. Передаем этот скрипт на выполнение интерпретатору. Интерпретатор транслирует код в промежуточный байткод, а затем виртуальная машина переводит полученный байткод в набор инструкций, которые выполняются операционной системой.

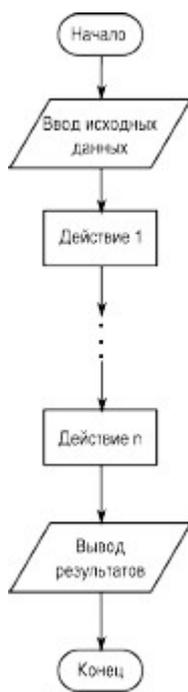
7. Введение в Python (ОК 01-02, ОК 09, ПК 1.1) Ответ:

Создание Python было начато Гвидо ван Россумом (Guido van Rossum) в 1991 году, когда он работал над распределенной ОС Амеба. Ему требовался расширяемый язык, который бы обеспечил поддержку системных вызовов. За основу были взяты ABC и Модула-3. В качестве названия он выбрал Python в честь комедийных серий BBC "Летающий цирк Монти-Пайтона", а вовсе не по названию змеи. С тех пор Python развивался при поддержке тех организаций, в которых Гвидо работал. Особенно активно язык совершенствуется в настоящее время, когда над ним работает не только команда создателей, но и целое сообщество программистов со всего мира. И все-таки последнее слово о направлении развития языка остается за Гвидо ван Россумом.

8. Линейный алгоритм (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Линейный алгоритм — алгоритм, в котором вычисления выполняются строго последовательно. Типичная блок-схема линейного алгоритма показана на рис

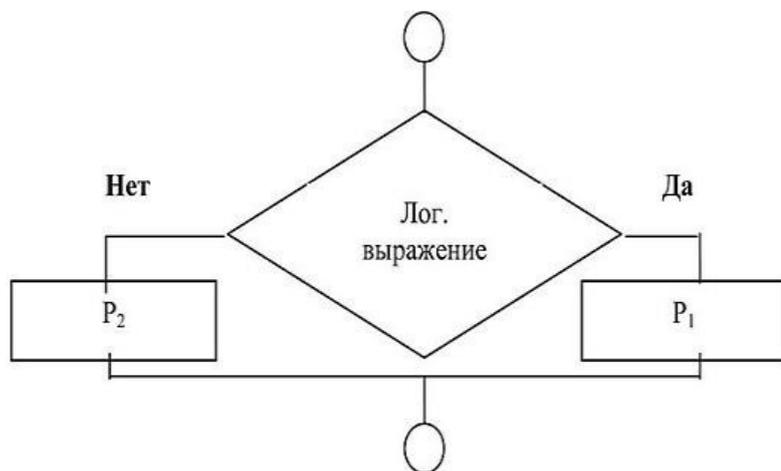


9. Разветвляющийся алгоритм (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Алгоритм называется разветвляющимся, если последовательность выполнения шагов алгоритма изменяется в зависимости от выполнения некоторых условий. Условие — это логическое выражение, которое может принимать одно из двух значений: «ДА» - если условие верно (истинно), и «НЕТ» - если условие неверно (ложно).

Разветвляющийся алгоритм можно реализовать в программах с помощью простого, сокращенного, составного операторов, а также конструкции многозначных ветвлений.



10. Циклический алгоритм (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Циклом называется фрагмент алгоритма или программы, который может повторяться несколько раз (в том числе и нуль раз). Каждая циклическая конструкция начинается заголовком цикла и заканчивается конечным оператором. Между ними располагаются операторы, называемые «телом цикла». Количество повторений выполнения команд (операторов), составляющих тело цикла, 12 / 34И.А.Хахаев определяется условием окончания цикла. Условием окончания может быть достижение некоторого значения специальной переменной, называемой параметром цикла (переменной цикла), или выполнение (прекращение выполнения) некоторого условия. Для организации циклов с параметром в языках программирования используется составной оператор FOR («для»), а в циклах с условием чаще всего используется составной оператор WHILE («пока»).

11. Работа с кортежами и списками (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Кортежи в Python аналогичны структурам, которые называются массивами. Приемы работы с массивами рассматриваются при обучении таким популярным языкам программирования, как C, Microsoft Visual Basic, Delphi и др. Стоит отметить, что массивы в них определяются как формальное объединение нескольких однотипных объектов (чисел, символов, строк и т. п.), рассматриваемое как единое целое.

В Python кортеж определяется как один из видов последовательностей, с которыми можно работать в этом языке программирования. Отличие от привычных массивов данных (если вы умеете программировать на других языках) заключается в том, что последовательность данных, объединенная в кортеж, не позволяет изменять свои значения. В этом есть определенные преимущества. Во-первых, увеличивается скорость обработки элементов кортежа, поскольку системе заранее известно, что значения не будут изменяться. Во-вторых, такая структура, как кортеж, может применяться для образования других структур: например, словари, работа с которыми будет рассмотрена позднее. В-третьих, кортеж занимает меньше памяти, чем, например, списки, и, кроме того, элементы кортежа защищены от случайных изменений.

Списки в Python функционируют подобно кортежам, с той разницей, что являются изменяемой последовательностью. Таким образом, если такие действия, как добавление нового элемента, удаление элемента, сортировка и др., в кортеже невозможны, то методы работы со списком позволяют это сделать.

Точно так же, как и элементы кортежа, элементы списка содержат лишь ссылку на объект, поэтому в списки могут входить разные типы данных. Данный подход отличается от обработки традиционных массивов в других языках программирования, хотя те, кто пробовал с ними работать, заметит сходства в методах обработки списков в Python с методами обработки массивов.

Рассмотрим синтаксис объявления списков.

Имя списка = [элемент1, элемент2, ...элемент!Ч]

Например, spisok=[1, 2, 3, 4, 5]

Типичными задачами при работе со списками являются: определение факта наличия в них заданного элемента и отбор элементов, удовлетворяющих определенным условиям. В обоих случаях используется циклическое сравнение элементов списка с заданным образцом. Для определения факта наличия заданного образца в списке достаточно единственного совпадения, после чего дальнейший просмотр прекращается. Если условие отбора может выполняться для нескольких элементов списка, то необходим просмотр всего списка до конца.

12. Работа со строками (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Строка в Python - это объект класса str, поэтому ранее, для того чтобы работать с числовыми данными, мы применяли функции приведения типов, например, int, как это продемонстрировано в следующем примере:

```
ind=int(input(" ВВедите индекс элемента "))
```

При работе со строками важно знать, что символ "А" - это совсем не то же самое, что символ "а". Длина строки ограничена лишь объемом оперативной памяти компьютера, и для обращения к элементу строки достаточно указать его индекс в квадратных скобках. Строки в Python представляют собой неизменяемую последовательность и обрабатываются с использованием цикла с оператором for. Важно понять, что изменить символ строки, обратившись к нему по индексу, невозможно

13. Обработка вложенных последовательностей (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Можно выделить ряд приемов, позволяющих реализовать основные задачи обработки вложенных последовательностей, а именно:

1. Нахождение количества элементов вложенной последовательности при заданном условии;
2. Нахождение суммы значений элементов вложенной последовательности при заданном условии;
3. Нахождение произведения значений элементов вложенной последовательности при заданном условии;
4. Поиск экстремальных значений элементов вложенной последовательности (поиск максимального и/или минимального значения);
5. Обмен столбцов элементов вложенной последовательности;
6. Обмен строк элементов вложенной последовательности;
7. Удаление заданной строки вложенной последовательности;
8. Замена значений элементов вложенной последовательности.

14. Работа с функциями. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Для объявления функции используется следующий синтаксис, def
ИмяФункции(Параметр(ы)):

Операторы функции

return возвращаемое значение где

- def (от англ. define - определять, устанавливать);
- ИмяФункции - уникальное имя создаваемой функции. На имя функции распространяются общие правила написания идентификаторов, перечисленные в параграфе 1.5;
- Параметры (аргументы) - список необязательных аргументов, разделенных запятыми и используемых в данной функции;
- Операторы функции - блок операторов, который выполняет работу функции;
- return - необязательный оператор, с помощью которого можно указать место, где в блоке кода функции требуется вернуть значение в вызывающую программу, и каково это возвращаемое значение. После выполнения данного оператора происходит выход из функции, и управление передается в то место программы, откуда эта функция была вызвана.

Вызов пользовательской функции происходит следующим образом:
ИмяПеременной=ИмяФункции(Параметр(ы))

15. Работа с файлами (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

В соответствии с этапами работы с файлами, перечисленными выше, первое, что нужно сделать, - открыть файл. Для этого в Python существует функция open, синтаксис которой следующий:

Файловая переменная=open (FileName, Mode, Encoding)

Файловая переменная - переменная, в которой будет находиться значение, возвращенное функцией open.

16. Символьные величины в Python. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

В языке Python часто приходится работать со строками. Ввод значений с клавиатуры происходит в символьном формате, печать данных в текстовый файл также производится в символьном формате. Часто в программе приходится совершать конвертацию числовых величин в символьные.

В языке Python есть один символьный тип - строки (Text Sequence Type).

17. Напишите синтаксис оператора цикла while. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Оператор цикла while выполняет указанный набор инструкций до тех пор, пока условие цикла истинно. Истинность условия определяется также как и в операторе if. Синтаксис оператора while выглядит так.

while выражение:

```
инструкция_1
инструкция_2
...
инструкция_n
```

Выполняемый набор инструкций называется телом цикла.

Пример.

```
a = 0
while a < 7:
    print("A")
    a += 1
```

Буква "А" будет выведена семь раз в столбик.

Пример бесконечного цикла.

```
a = 0
while a == 0:
    print("A")
```

18. Перечислите классические способы обработки кортежей. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Решение задач, связанных с обработкой последовательностей данных, базируется на их элементарных приемах обработки. Разбив задачу на логические части, можно значительно

ускорить ее решение. Типичными задачами работы с кортежами являются определение наличия в нем заданного элемента, отбор элементов, удовлетворяющих определенным условиям и т. д.

Из всего многообразия существующих приемов, позволяющих манипулировать с данными, представленными в кортежах, можно выделить следующие:

1. Нахождение количества элементов при заданном условии.
 2. Нахождение суммы значений элементов при заданном условии.
 3. Нахождение произведения значений элементов при заданном условии.
 4. Поиск экстремальных значений элементов кортежа (поиск максимального и/или минимального значения).
 5. Объединение (сцепление) кортежей.
 6. Обмен значений элементов в кортеже.
 7. Срезы кортежей.
19. Поясните, в чем состоит отличие списков, созданных на языке Python, от кортежей. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Основное различие между списками и кортежами заключается в том, что списки являются изменяемыми, а кортежи — нет. Это означает, что список можно изменить после его создания, добавив, удалив или изменив один или несколько его элементов.

```
list_numbers.append(6) # Добавляем элемент в список  
list_numbers[0] = 0 # Изменяем первый элемент списка
```

В то время как кортежи являются неизменяемыми, то есть после создания кортежа его нельзя изменить. Попытка сделать это приведет к ошибке.

```
tuple_numbers.append(6) # Вызовет ошибку  
tuple_numbers[0] = 0 # Вызовет ошибку
```

20. Дайте определение такой структуры данных языка Python, как словарь.

Ответ: (ОК 01-02, ОК 09, ПК 1.1)

В языке программирования Python словари (тип dict) представляют собой еще одну разновидность структур данных наряду со списками и кортежами. Словарь - это изменяемый (как список) неупорядоченный (в отличие от строк, списков и кортежей) набор элементов "ключ:значение".

"Неупорядоченный" – значит, что последовательность расположения пар не важна, в следствие чего обращение к элементам по индексам невозможно.

В других языках структуры, схожие со словарями, называются по-другому. Например, в Java подобный тип данных называется отображением.

Чтобы представление о словаре стало более понятным, проведем аналогию с обычным словарем, например, англо-русским. На каждое английское слово в таком словаре есть русское слово-перевод: cat – кошка, dog – собака, table – стол и т. д. Если англо-русский словарь описать с помощью Python, то английские слова можно сделать ключами, а русские – их значениями:

```
{'cat': 'кошка', 'dog': 'собака', 'bird': 'птица'}
```

Вопросы к экзамену (ОК 01-02, ОК 09, ПК 1.1)

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Объектно-ориентированный подход заключается в следующем наборе основных принципов (цитируется по вышеупомянутой книге Т. Бадда):

Всё является объектом.

Вычисления осуществляются путём взаимодействия (обмена данными) между объектами, при котором один объект требует, чтобы другой объект выполнил некоторое действие. Объекты

взаимодействуют, посылая и получая сообщения. Сообщение — это запрос на выполнение действия, дополненный набором аргументов, которые могут понадобиться при выполнении действия.

Каждый объект имеет независимую память, которая состоит из других объектов.

Каждый объект является представителем класса, который выражает общие свойства объектов (таких, как целые числа или списки).

В классе задаётся поведение (функциональность) объекта. Тем самым все объекты, которые являются экземплярами одного класса, могут выполнять одни и те же действия.

Классы организованы в единую древовидную структуру с общим корнем, называемую иерархией наследования. Память и поведение, связанное с экземплярами определённого класса, автоматически доступны любому классу, расположенному ниже в иерархическом дереве.

Объектно-ориентированное программирование (ООП) - методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

В языках программирования понятие объекта реализовано как совокупность свойств (структур данных, характерных для данного объекта), методов их обработки (подпрограмм изменения их свойств) и событий, на которые данный объект может реагировать и, которые приводят, как правило, к изменению свойств объекта.

2. Понятие объекта. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Понятие объекта в программе совпадает с обыденным смыслом этого слова: объект представляется как совокупность данных, характеризующих его состояние, и функций их обработки, моделирующих его поведение. Вызов функции на выполнение часто называют посылкой сообщения объекту.

При создании объектно-ориентированной программы предметная область представляется в виде совокупности объектов. Выполнение программы состоит в том, что объекты обмениваются сообщениями.

3. Класс (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Класс - универсальный, комплексный тип данных, состоящий из тематически единого набора «полей» (переменных более элементарных типов) и «методов» (функций для работы с этими полями), то есть он является моделью информационной сущности с внутренним и внешним интерфейсами для оперирования своим содержимым (значениями полей). В частности, в классах широко используются специальные блоки из одного или чаще двух спаренных методов, отвечающих за элементарные операции с определённым полем (интерфейс присваивания и считывания значения), которые имитируют непосредственный доступ к полю. Эти блоки называются «свойствами» и почти совпадают по конкретному имени со своим полем (например, имя поля может начинаться со строчной, а имя свойства – с заглавной буквы). Другим проявлением интерфейсной природы класса является то, что при копировании соответствующей переменной через присваивание, копируется только интерфейс, но не сами данные, то есть класс – ссылочный тип данных. Переменная-объект, относящаяся к заданному классом типу, называется экземпляром этого класса. При этом в некоторых исполняющих системах класс также может представляться некоторым объектом при выполнении программы посредством динамической идентификации типа данных. Обычно классы разрабатывают таким образом, чтобы обеспечить отвечающие природе объекта и решаемой задаче целостность данных объекта, а также удобный и простой интерфейс. В свою очередь, целостность предметной области объектов и их интерфейсов, а также удобство их проектирования, обеспечивается наследованием.

4. Основные принципы ООП: Абстракция (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Абстра́кция — в объектно-ориентированном программировании это придание объекту характеристик, которые отличают его от всех других объектов, четко определяя его концептуальные границы. Основная идея состоит в том, чтобы отделить способ использования составных объектов данных от деталей их реализации в виде более простых объектов, подобно тому, как функциональная абстракция разделяет способ использования функции и деталей её реализации в терминах более примитивных функций, таким образом, данные обрабатываются функцией высокого уровня с помощью вызова функций низкого уровня. Такой подход является основой объектно-ориентированного программирования. Это позволяет работать с объектами, не вдаваясь в особенности их реализации. В каждом конкретном случае применяется тот или иной подход: инкапсуляция, полиморфизм или наследование. Например, при необходимости обратиться к скрытым данным объекта, следует воспользоваться инкапсуляцией, создав, так называемую, функцию доступа или свойство. Абстракция данных — популярная и в общем неверно определяемая техника программирования. Фундаментальная идея состоит в разделении несущественных деталей реализации подпрограммы и характеристик существенных для корректного ее использования. Такое разделение может быть выражено через специальный «интерфейс», сосредотачивающий описание всех возможных применений программы.

5. Основные принципы ООП: Инкапсуляция (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Инкапсуля́ция — свойство языка программирования, позволяющее пользователю не задумываться о сложности реализации используемого программного компонента (что у него внутри?), а взаимодействовать с ним посредством предоставляемого интерфейса (публичных методов и членов), а также объединить и защитить жизненно важные для компонента данные. При этом пользователю предоставляется только спецификация (интерфейс) объекта. Пользователь может взаимодействовать с объектом только через этот интерфейс. Реализуется с помощью ключевого слова: `public`. Пользователь не может использовать закрытые данные и методы. Реализуется с помощью ключевых слов: `private`, `protected`, `internal`. Инкапсуляция — один из четырёх важнейших механизмов объектно-ориентированного программирования (наряду с абстракцией, полиморфизмом и наследованием). Соккрытие реализации целесообразно применять в следующих случаях: предельная локализация изменений при необходимости таких изменений, прогнозируемость изменений (какие изменения в коде надо сделать для заданного изменения функциональности) и прогнозируемость последствий изменений.

6. Основные принципы ООП: Наследование (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Насле́дование — один из четырёх важнейших механизмов объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом. Другими словами, класс-наследник реализует спецификацию уже существующего класса (базовый класс). Это позволяет обращаться с объектами класса-наследника точно так же, как с объектами базового класса. Простое наследование: Класс, от которого произошло наследование, называется базовым или родительским (англ. `base class`). Классы, которые произошли от базового, называются потомками, наследниками или производными классами (англ. `derived class`). В некоторых языках используются абстрактные классы. Абстрактный класс — это класс, содержащий хотя бы один абстрактный метод, он описан в программе, имеет поля, методы и не может использоваться для непосредственного создания объекта. То есть от абстрактного класса можно только наследовать. Объекты создаются только на основе производных классов, наследованных от абстрактного. Например, абстрактным классом может быть базовый класс «сотрудник вуза», от которого наследуются классы «аспирант», «профессор» и т. д. Так как производные классы имеют общие поля и функции (например, поле «год рождения»), то эти члены класса могут быть описаны в базовом классе. В программе создаются объекты на

основе классов «аспирант», «профессор», но нет смысла создавать объект на основе класса «сотрудник вуза».

7. Основные принципы ООП: Полиморфизм (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Полиморфизм — возможность объектов с одинаковой спецификацией иметь различную реализацию. Язык программирования поддерживает полиморфизм, если классы с одинаковой спецификацией могут иметь различную реализацию — например, реализация класса может быть изменена в процессе наследования. Кратко смысл полиморфизма можно выразить фразой: «Один интерфейс, множество реализаций». Полиморфизм — один из четырёх важнейших механизмов объектно-ориентированного программирования (наряду с абстракцией, инкапсуляцией и наследованием). Полиморфизм позволяет писать более абстрактные программы и повысить коэффициент повторного использования кода. Общие свойства объектов объединяются в систему, которую могут называть по-разному — интерфейс, класс. Общность имеет внешнее и внутреннее выражение: внешняя общность проявляется как одинаковый набор методов с одинаковыми именами и сигнатурами (именем методов и типами аргументов, и их количеством); внутренняя общность — одинаковая функциональность методов. Её можно описать интуитивно или выразить в виде строгих законов, правил, которым должны подчиняться методы. Возможность приписывать разную функциональность одному методу (функции, операции) называется перегрузкой метода (перегрузкой функций, перегрузкой операций).

8. Достоинства ООП (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

использование при программировании понятий, близких к предметной области;
возможность успешно управлять большими объемами исходного кода благодаря инкапсуляции, то есть скрытию деталей реализации объектов и упрощению структуры программы;

возможность многократного использования кода за счет наследования;
сравнительно простая возможность модификации программ;
возможность создания и использования библиотек объектов.

9. Недостатки ООП (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

некоторое снижение быстродействия программы, связанное с использованием виртуальных методов;

идеи ООП не просты для понимания и в особенности для практического использования;
для эффективного использования существующих объектно-ориентированных систем требуется большой объем первоначальных знаний;

неграмотное применение ООП может привести к значительному ухудшению характеристик разрабатываемой программы.

10. Что понимают под интерфейсом? (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Интерфейс — это набор инструментов, который позволяет пользователю взаимодействовать с программой. В более широком смысле термин обозначает любые инструменты для соприкосновения между разными системами и сущностями.

11. Опишите словесный алгоритм создания в Python базового окна при использовании библиотеки tkinter. (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Основным пакетом для работы с mainloop в Python является tkinter. Он предоставляет набор инструментов для создания графического интерфейса и включает в себя метод mainloop(), который запускает основной цикл исполнения программы.

12. Кортеж (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Кортеж – это упорядоченная последовательность из нуля или более ссылок на объекты. Кортежи поддерживают тот же синтаксис получения срезов, что и строки. Это упрощает извлечение элементов из кортежа. Подобно строкам, кортежи относятся к категории неизменяемых объектов, поэтому мы не можем замещать или удалять какие-либо их элементы. Если нам необходимо иметь возможность изменять упорядоченную последовательность, то вместо кортежей можно просто использовать списки или, если в программе уже используется кортеж, который нежелательно модифицировать, можно преобразовать кортеж в список с помощью функции преобразования `list()` и затем изменять полученный список.

Именованные кортежи ведут себя точно так же, как и обычные кортежи, и не уступают им в производительности. Отличаются они возможностью ссылаться на элементы кортежа не только по числовому индексу, но и по имени, что в свою очередь позволяет создавать сложные агрегаты из элементов данных.

13. Список (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

Список – это упорядоченная последовательность из нуля или более ссылок на объекты. Списки поддерживают тот же синтаксис получения срезов, что и строки с кортежа. Это упрощает извлечение элементов из списка. В отличие от строк и кортежей списки относятся к категории изменяемых объектов, поэтому мы можем замещать или удалять любые их элементы. Кроме того, существует возможность вставлять, замещать и удалять целые срезы списков.

14. Создание виджета Флажок (ОК 01-02, ОК 09, ПК 1.1)

Ответ:

С помощью флажков (Checkbox) пользователь выбирает один или несколько элементов из предложенных ему альтернатив. Разработаем программу на основе тестового вопроса, использующую флажки.

Флажок имеет два состояния: включен - выключен. За текущее состояние флажка в программе будут отвечать атрибут `self.otvet1` для первого флажка, `self.otvet2` для второго флажка и `self.otvet3` для третьего флажка. Для обработки данных разных типов в библиотеке `tkinter` содержатся несколько классов таких, например, как `IntVar()` - для обработки целых чисел, `DoubleVar()` - для обработки дробных чисел, `BooleanVar()` - для обработки логических значений. В нашей программе атрибуты `self.otvet1`, `self.otvet2` и `self.otvet3` мы свяжем с классом `BooleanVar()`, что позволит определять статус флажков, а затем передавать его параметру `znach` каждого флажка.

Практические задания к экзамену (ОК 01-02, ОК 09, ПК 1.1)

1. Под объектами понимают:

- а) Всю абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
- б) Некоторую абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
- в) Некоторую видимую сущность, заданную набором имен атрибутов и имен методов поведения

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

2. Изменение состояния объекта в ответ на какое-либо действие:

- а) Значение
- б) Событие
- в) Данность

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

3. Характеристика объекта:

- а) Событие
- б) Данность
- в) Свойство

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

4. Свойство, при котором объекты содержат описание атрибутов и действий

одновременно:

- а) Наследование
- б) Полиморфизм
- в) Инкапсуляция

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

5. Один из принципов объектно-ориентированного программирования:

- а) Отдача
- б) Передача
- в) Наследование

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

6. Укажите принципы объектно-ориентированного программирования:

- а), наследование
- б) полиморфизм
- в) инкапсуляция

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

7. Совокупность свойств и методов:

- а) Объект
- б) Свойство
- в) Событие

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

8. Соккрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс, называется:

- а) абстракцией
- б) инкапсуляцией
- в) агрегацией

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

9. Можно ли в XML использовать собственные теги:

- а) нельзя
- б) можно
- в) можно, если указаны пространства имен

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

10. Для чего нужны функции модуля gettext:

- а) для получения текста от пользователя
- б) для обеспечения интернационализации программы
- в) для чтения строки со стандартного ввода

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

11. Что будет получено в результате вычисления следующего выражения:

$(0 < 5 \leq 3)$ and $(0 / 0)$:

- а) True (или 1)
- б) False (или 0)
- в) синтаксическая ошибка

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

12. В каком модуле нужно искать функции, помогающие тестировать программу:

- а) dictutils
- б) profile
- в) unittest

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

13. Дан массив `>>> c = array([[1,2], [2,3], [4,5]])`

Чему равен срез `c[:,1]`:

- а) `array([1, 2, 4])`
- б) `array([2, 3])`
- в) `array([2, 3, 5])`

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

14. Чему будет равен результат выполнения:

`urlparse.urlsplit(«http://google.com/search?q=Python#1»):`

- а) ('http', 'google.com', '/search', », 'q=Python', '1')
- б) ('http://', 'google.com/', 'search?', 'q=Python#', '1')
- в) ('http', 'google.com', '/search', 'q=Python', '1')

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

15. Какая встроенная функция Python лучше всего подходит для цепочечных вычислений (в частности, вычислений значения многочлена по схеме Горнера):

- а) chain()
- б) map()
- в) reduce()

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

16. Можно ли в XML использовать собственные теги:

- а) нельзя
- б) можно
- в) можно, если указаны пространства имен

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

17. Что включает в себя Zope:

- а) поддержку CGI-сценариев
- б) поддержку сценариев DTML
- в) СУБД общего назначения

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

18. Как перевести Unicode-строку u в кодировку koi8-r:

- а) u.decode('koi8-r')
- б) u.encode('koi8-r')
- в) string.decode(u, 'koi8-r')

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

19. Как используется строка Main Heading в следующем примере:

Main Heading

- а) в качестве заглушки, чтобы в средствах визуального форматирования было видно, что форматируется
- б) в качестве имени переменной для макроподстановки
- в) содержит значение, на которое можно сослаться в другом месте документа

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

20. Зачем в XML пространства имен:

- а) пространства имен позволяют указывать опции для приложения, обрабатывающего XML
- б) пространства имен позволяют включать однотипные XML-документы друг в друга
- в) для сочетания в одном документе XML с различными DTD

Оценка: дихотомическая. Правильное выполнение одного такого ТЗ задания оценивается 1 баллом, неправильное – 0 баллов. (ОК 01-02, ОК 09, ПК 1.1)

Ключ:

| | | | | | | | | | | | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Номер вопроса | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Правильный ответ | б | б | в | в | в | в | а | б | б | б | б | в | в | в | в | б | б | б | а | в |

Критерии оценки на экзамене

| «отлично» | «хорошо» | «удовлетворительно» | «неудовлетворительно» |
|---|--|--|--|
| студент раскрывает теоретический вопрос билета, уверенно отвечает на дополнительные вопросы | студент раскрывает теоретический вопрос, на дополнительные вопросы отвечает неуверенно, допускает неточности в определениях. | студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), | Теоретический вопрос не раскрыт, |
| студент выполнил тест на 91-100% и набрал 91-100 баллов. | студент выполнил тест на 71-90% и набрал 71-90 баллов. | студент выполнил тест на 60-70% и набрал 60-70 баллов. | студент выполнил тест на менее 0-59% и набрал 0-59 баллов. |