Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 18.02.2026 11:37:26
Уникальный программный ключ:
48505f11ec15acaa386f5219d3113d727fefda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Казанский (Приволжский) федеральный университет»
Колледж Елабужского института (филиала) КФУ

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

Программа дисциплины (модуля)
ОП.13 Программирование на языке высокого уровня

Специальность: 09.02.07 Информационные системы и программирование

Квалификация выпускника: Программист

Форма обучения: очная

На базе: основного общего образования

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

1. Цели освоения дисциплины

формирование знаний по:

— основным этапам разработки программного обеспечения;
— основным принципам технологии структурного программирования;
— основным принципам отладки и тестирования программных продуктов;
— языку программирования Python.

формирование умений по:

— осуществлению разработки кода программного модуля на языках низкого и высокого уровней;
— созданию программы на языке программирования Python по разработанному алгоритму;
— выполнению отладки и тестирования программы на языке программирования Python,

2. Место дисциплины в структуре ППССЗ

Учебная дисциплина ОП.13 «Программирование на языке высокого уровня» является общепрофессиональной дисциплиной профессионального цикла в соответствии с ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование».

Осваивается четвёртом курсе (8 семестр).

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 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Использовать информационные технологии в профессиональной деятельности
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

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

— основные этапы разработки программного обеспечения;
— основные принципы технологии структурного программирования;
— основным принципы отладки и тестирования программных продуктов;
— язык программирования Python.

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

— осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

— создавать программы на языке программирования Python по разработанному алгоритму;
— выполнять отладку и тестирование программ на языке программирования Python.

4. Структура и содержание дисциплины

4.1. Общая трудоемкость дисциплины в часах

Общая трудоемкость дисциплины составляет 108 часов.

Форма промежуточной аттестации по дисциплине: экзамен в 8 семестре.

Разделы и темы дисциплины		Семестр	Виды и часы аудиторной работы, их трудоемкость (в часах)			Самостоятельная работа	Текущие формы контроля
			Лекции	Практические занятия	Лабораторные работы		
Тема 1	**Разработка графических интерфейсов В программе на языке Python**	8	16	0	16	1	Проверочная работа 1*
Тема 2	**Инструменты для создания графических Интерфейсов пользователя (GUI)**	8	16	0	16	1	Проверочная работа 2*
Тема 3	**Элементы графического интерфейса**	8	16	0	16	2	Проверочная работа 3*
Итого:100			**48**	**0**	**48**	**4**	
Консультация						**1**	
Экзамен						**7**	
Всего по дисциплине						**108**	

* письменная Проверочная 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 Разработка графических интерфейсов	**Содержание учебного материала**	**33 (16/16/1)**	
	1. Событие.	4	*1*
	2. Обработчик события.	6	*1*
	3. Цикл обработки событий.	6	*1*
	Лабораторные работы: 1.Событие. 2.Обработчик события. 3.Цикл обработки событий.	 4 6 6	 2 2 2
	Самостоятельная работа обучающегося Подготовка к проверочной работе по темам: Событие. Обработчик события. Цикл обработки событий. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. стр. 180-182	2	*3*
Тема 2. Инструменты для создания графических	**Содержание учебного материала**	**33 (16/16/1)**	
	1. Общие сведения о GUI Python.	4	*1*
	2. Отслеживание событий.	6	*1*
	3. Создание базового окна.	6	*1*
	Лабораторные работы 1. Общие сведения о GUI Python. 2. Отслеживание событий. 3. Создание базового окна.	 4 6 6	 2 2 2
	Самостоятельная работа обучающегося	2	*3*

	Подготовка к проверочной работе по темам: Общие сведения о GUI Python. Отслеживание событий. Создание базового окна. Жуков, Р. А. Язык программирования Python. Практикум: учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. Стр. 182-185		
Тема 3. Элементы графического интерфейса	**Содержание учебного материала**	**34 (16/16/2)**	
	1. Создание и конфигурирование виджета. Менеджер размещения.	4	*1*
	2. Использование элементов Button (Кнопка). Label (Надпись). Entry (Поле ввода).	6	*1*
	3. Checkbutton (Флажок). Radiobutton (Переключатель). Другие виджеты. Метод bind. Canvas.	6	*1*
	Лабораторные работы 1. Создание и конфигурирование виджета. Менеджер размещения. 2. Использование элементов Button (Кнопка). Label (Надпись). Entry (Поле ввода). 3. Checkbutton (Флажок). Radiobutton (Переключатель). Другие виджеты. Метод bind. Canvas.	 4 6 6	 2 2 2
	Самостоятельная работа обучающегося Подготовка к проверочной работе по темам: Создание и конфигурирование виджета. Менеджер размещения. Использование элементов Button (Кнопка). Label (Надпись). Entry (Поле ввода). Checkbutton (Флажок). Radiobutton (Переключатель). Другие виджеты. Метод bind. Canvas. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. Стр. 187-205	2	*3*
Примерная тематика курсовой работы (проекта)		не предусмотрено	
Итого:		**100 (48/48/4)**	

Консультация	**1**	
Экзамен	**7**	
Всего	**108**	

*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	Разработка графических интерфейсов	**Самостоятельная работа обучающегося** Подготовка к проверочной работе по темам: Событие. Обработчик события. Цикл обработки событий. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. стр. 180-182	1	Проверочная работа 1
Тема 2	Инструменты для создания графических	**Самостоятельная работа обучающегося** Подготовка к проверочной работе по темам: Общие сведения о GUI Python. Отслеживание событий. Создание базового окна. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. Стр. 182-185	1	Проверочная работа 2*
Тема 3	Элементы графического интерфейса	**Самостоятельная работа обучающегося** Подготовка к проверочной работе по темам: Создание и конфигурирование виджета. Менеджер размещения. Использование элементов Button (Кнопка). Label (Надпись). Entry (Поле ввода). Checkbutton (Флажок). Radiobutton (Переключатель). Другие виджеты. Метод bind. Canvas. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст: электронный. - URL:	2	Проверочная работа 3

		https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке. Стр. 187-205		
Всего по дисциплине			**4**	

5. Образовательные технологии

Освоение дисциплины «Программирование на языке высокого уровня» 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	Разработка графических интерфейсов	Информационно-проблемная лекция	16
Тема 2	Инструменты для создания графических	Информационно-проблемная лекция	16
Тема 3	Элементы графического интерфейса	Информационно-проблемная лекция	16
Всего по дисциплине			16

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

6.1 Оценочные средства для текущего контроля успеваемости

Проверочная работа 1 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Пример заданий проверочной работы 1:

1 Укажите возможные события (не менее семи), которые могут возникнуть при диалоге пользователя с программой.
2. Основываясь на GUI, представленном на рисунке, предложите, какие события должна отслеживать программа.
3. Составьте алгоритм и представьте его в виде блок-схемы обработчика запроса на загрузку интернет-страницы в браузер, предполагая, что модель взаимодействия основана на клиент-серверной архитектуре.

Проверочная работа 2 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Пример заданий проверочной работы 2:

Создайте окно с названием «Студенты», где разместите кнопки «Вывести список», «Ввести данные», «Создать запрос», «Выйти» тремя рассмотренными способами. Сравните результат и прокомментируйте преимущества и недостатки использования данных методов.

Проверочная работа 3 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)
Примеры заданий:

1. Разработайте диалоговую программу на тему: Российские кинофильмы.
2. Разработайте диалоговую программу на тему: Спорт.
3. Разработайте диалоговую программу на тему: Автомобили.
4. Разработайте диалоговую программу на тему: Российская эстрада.
5. Разработайте диалоговую программу на тему: Программирование.
6. Разработайте диалоговую программу на тему: Животные.
7. Разработайте диалоговую программу на тему: Города России.

6.2 Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

Пример билета к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

1. Что такое Python?
2. Как преобразовать список в строку?
3. Создайте класс ФИГУРА с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создайте дочерние классы ПРЯМОУГОЛЬНИК, КРУГ, ТРЕУГОЛЬНИК со своими методами вычисления площади и периметра. Создайте список n фигур и выведите полную информацию о фигурах на экран.

Практическое задание

1. Студент регистрируется на информационно-образовательном портале вуза. Составьте концептуальную модель главного цикла обработчика событий, который включает основные события, обработчик событий, вызываемые методы.
2. Сотрудник склада вводит информацию о поступивших товарах в информационную систему организации. Составьте концептуальную модель главного цикла обработчика событий, который включает основные события, обработчик событий, вызываемые методы.
3. Схематически представьте обработчик формирования заказа в магазине.
4. Основываясь на предыдущем задании, включите в него обработчик исключений. Опишите основные возможные исключения.
5. Производится оплата заказа через терминал. Составьте концептуальную модель главного цикла обработчика событий, которая включает основные события, обработчик событий, вызываемые методы.

Вариант 2

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

Шифр компе	Планируемые результаты	Оценочные средства	Критерии оценивания результатов обучения (баллы)

тенции	**обучения**		2	3	4	5
	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
ОК 01	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	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 02	Знать основные этапы разработки программного	Проверочные работы 1-3 Вопросы к экзамену 1-80	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без	Знает достаточно в базовом объёме	Демонстрирует высокий уровень

	обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Практические варианты к экзамену 1-2		грубых ошибок		знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	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 04	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программиров	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	ания; основным принципы отладки и тестирования программных продуктов; язык программирования Python					
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	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 05	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	продуктов; язык программирования Python					
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	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 основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять	Проверочные работы 1-3	Не умеет, демонстриру	Демонстрирует	Умеет применять	Демонстрирует

	разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	ет частичные умения, допуская грубые ошибки	частичные умения без грубых ошибок	знания на практике в базовом объёме	высокий уровень умений
ОК 10	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	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

	высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python					
ПК 1.2	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программиров	Проверочные работы 1-3 Вопросы к экзамену 1-80	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

	ания Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python					
ПК 2.4	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и	Проверочные работы 1-3 Вопросы к экзамену 1-80	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

	тестирование программ на языке программирования Python					

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 прохождение 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

9.1. Основная литература:

1. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2023). – Режим доступа: по подписке.
2. Голицына, О. Л. Программирование на языке высокого уровня : 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: 28.08.2023). – Режим доступа: по подписке.
3. Гуриков, С. Р. Программирование на языке высокого уровня 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/catalog/product/1356003 (дата обращения: 28.08.2021). – Режим доступа: по подписке.

9.2. Дополнительная литература:

1. 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: 28.08.2021). — Режим доступа: для авториз. пользователей.
2. Ночка, Е. И. Программирование на языке высокого уровня. 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: 28.08.2021). – Режим доступа: по подписке.
3. Ахмедханлы, Д. М. Программирование на языке высокого уровня : учебно-методическое пособие / Д. М. Ахмедханлы, Н. В. Ушмаева. — Тольятти : ТГУ, 2016. — 123 с. — ISBN 978-5-8259-1022-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/139958 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.
4. Программирование на языке высокого уровня : учебное пособие / составитель Е. И. Николаев. — Ставрополь : СКФУ, 2015. — 211 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/155245 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.
5. 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: 28.08.2021). — Режим доступа: для авториз. пользователей.

Руководитель библиотеки Л.В. Беляева

10. Материально-техническое и программное обеспечение дисциплины

Освоение дисциплины ОП.13 «Программирование на языке высокого уровня» предполагает использование следующего материально-технического обеспечения:

Наименование дисциплины	Наименование кабинета, перечень оборудования
ОП.13 Программирование на языке высокого уровня	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: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus

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.

Программа составлена в соответствии с требованиями ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование».

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
_____ С.Ю. Бахвалов
«19» 12 2025 г.
МП

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОП.13 Программирование на языке высокого уровня

09.02.07 Информационные системы и программирование
(код и наименование специальности)

Программист
(квалификация выпускника)

Елабуга, 2025

Паспорт
фонда оценочных средств по дисциплине
ОП.13 Программирование на языке высокого уровня

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 основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2

		высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2

		высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2
ОК 09	Использовать информационные технологии в профессиональной деятельности	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2

		высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python. Уметь осуществлять разработку кода программного модуля на языках низкого и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2

		высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	

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

Тема 1 Разработка графических интерфейсов

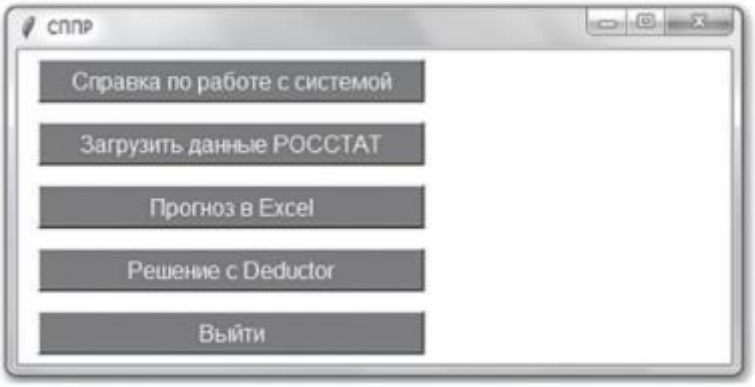
Комплект заданий для проверочной работы 1
ОП.13 «Программирование на языке высокого уровня»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Вариант 1

1 Укажите возможные события (не менее семи), которые могут возникнуть при диалоге пользователя с программой.
2. Основываясь на GUI, представленном на рисунке, предложите, какие события должна отслеживать программа.
3. Составьте алгоритм и представьте его в виде блок-схемы обработчика запроса на загрузку интернет-страницы в браузер, предполагая, что модель взаимодействия основана на клиент-серверной архитектуре.

Вариант 2

1 Основываясь на GUI, представленном на рисунке, предложите свой вариант интерфейса СППР (система поддержки принятия решений). Как можно задействовать радиокнопки?
2. Составьте концептуальную модель (описательная модель на естественном языке) обработчика нажатия ЛКМ на кнопку панели инструментов «Сохранить» в MS Office.
3. Основываясь на GUI, представленном на рисунке, предложите, какие события должна отслеживать программа.



Тема 2. Инструменты для создания графических интерфейсов

Комплект заданий для проверочной работы 2
ОП.13 «Программирование на языке высокого уровня»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)
Вариант 1

Создайте окно с названием «Студенты», где разместите кнопки «Вывести список», «Ввести данные», «Создать запрос», «Выйти» тремя рассмотренными способами. Сравните результат и прокомментируйте преимущества и недостатки использования данных методов.

Вариант 2

Создайте окно с названием «Покупатели», где разместите кнопки «Вывести список», «Ввести данные», «Создать запрос», «Выйти» тремя рассмотренными способами. Сравните результат и прокомментируйте преимущества и недостатки использования данных методов.

Тема 3. Элементы графического интерфейса

Комплект заданий для проверочной работы 3
ОП.13 «Программирование на языке высокого уровня»
Защита программного приложения (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

1. Разработайте диалоговую программу на тему: Российские кинофильмы.
2. Разработайте диалоговую программу на тему: Спорт.
3. Разработайте диалоговую программу на тему: Автомобили.
4. Разработайте диалоговую программу на тему: Российская эстрада.
5. Разработайте диалоговую программу на тему: Программирование.
6. Разработайте диалоговую программу на тему: Животные.
7. Разработайте диалоговую программу на тему: Города России.
8. Разработайте диалоговую программу на тему: Достопримечательности моего города.
9. Разработайте диалоговую программу на тему: Кулинария.
10. Разработайте диалоговую программу на тему: Мотоциклы.
11. Разработайте диалоговую программу на тему: Живопись.
12. Разработайте диалоговую программу на тему: Книги.
13. Разработайте диалоговую программу на тему: Мультфильмы.
14. Разработайте диалоговую программу на тему: Зарубежное кино.
15. Разработайте диалоговую программу на тему: Зарубежная эстрада.
16. Разработайте диалоговую программу на тему: Военная техника.
17. Разработайте диалоговую программу на тему: Классическая музыка.
18. Разработайте диалоговую программу на тему: Домашние животные.
19. Разработайте диалоговую программу на тему: Информатика.
20. Разработайте диалоговую программу на тему: Моря и океаны.
21. Разработайте диалоговую программу на тему: Великие пустыни мира.
22. Разработайте диалоговую программу на тему: Герои олимпиад.
23. Разработайте диалоговую программу на тему: История России.
24. Разработайте диалоговую программу на тему: Русские поэты.
25. Разработайте диалоговую программу на тему: Дикие животные.
26. Разработайте диалоговую программу на тему: Российские киноактеры.
27. Разработайте диалоговую программу на тему: Интернет.
28. Разработайте диалоговую программу на тему: Русские народные сказки.
29. Разработайте диалоговую программу на тему: Легенды российского балета.
30. Разработайте диалоговую 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 основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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 02	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого	Не умеет, демонстрирует частичные умения,	Демонстрирует частичные умения без грубых	Умеет применять знания на практике в базовом	Демонстрирует высокий уровень умений

	уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	допуская грубые ошибки	ошибок	объёме	
ОК 04	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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 05	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования;	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	основным принципы отладки и тестирования программных продуктов; язык программирования Python				
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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 08	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование	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

	программ на языке программирования Python				
ОК 09	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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
ПК 1.2	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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
ПК 2.4	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципы отладки и тестирования программных продуктов; язык программирования Python	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и тестирование программ на языке программирования Python	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

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

Вопросы к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

1. Что такое Python?
2. Каковы преимущества использования Python?
3. Как Python интерпретируется как язык?
4. Как интерпретируется Python?
5. Как в Python управляется память?
6. Что такое PEP 8?
7. Как писать комментарии на Python?
8. Как прокомментировать несколько строк в Python?
9. Что такое строки документации в Python?
10. Обязательны ли отступы в Python?
11. Что такое функция в Python?
12. Что такое локальные и глобальные переменные в Python?
13. Что такое лямбда-функция?
14. Почему лямбда-формы в Python не имеют операторов?
15. Какие типы данных поддерживаются в Python?
16. Что такое индексы?
17. Что такое отрицательные индексы и почему они используются?
18. Что такое словарь в Python?
19. Как получить доступ к значениям в словаре?
20. Как получить список всех ключей в словаре?
21. В чем разница между списком и кортежем?
22. Что такое итераторы в Python?
23. Что делает [:: -1}?
24. Как можно использовать тернарные операторы в Python?
25. Как работает break?
26. Каков смысл оператора pass в Python?
27. Что такое функция map в Python?
28. Что такое функция enumerate в Python?
29. Что такое абстракции у словаря и списка в Python?
30. Что такое slicing в Python?
31. Каков смысл not в операторе?
33. Как добавить новое значение в объект списка?
34. Что такое поверхностная копия?
35. Что такое глубокая копия?
36. Как создать пустой класс в Python?
37. Что означает ключевое слово self в Python?
38. Будет ли цикл do-while работать, если вы не закончите его точкой с запятой?
39. Как преобразовать список в строку?
41. Что такое операторы тождественности в Python?
42. Как обрабатывать входные данные в Python?
43. Что делает функция zip()?
44. В чем разница между тем, когда функция range() принимает один аргумент, два и три?
45. Напишите лучший код для перестановки двух чисел местами.
46. Как можно объявить несколько назначений в одной строке кода?
47. Как вырваться из бесконечного цикла?

49. Когда выполняется except, в блоке try-except?
51. Что такое модули Python?
52. Что такое PYTHONPATH?
53. Назовите пример режимов обработки файлов с помощью Python?
54. Что такое pickling и unpickling?
55. Являются ли массивы Python NumPy лучше списков?
56. Как вычислять процентили с помощью NumPy?
57. Как получить текущий рабочий каталог с помощью Python?
58. Что вы видите снизу? Что произойдет, если мы выполним этот код?
59. Как бы вы произвели рандомизацию содержимого списка на месте?
60. Что такое приведение в Python?
62. Как можно удалить переменные в Python?
63. Что такое pandas в Python?
64. В чем разница между методами append() и extend()?
65. Как получить текущую версию Python?
66. Что значит *args, **kwargs? И зачем нам их использовать?
67. Что за функции help() и dir() в Python?
68. Что означает одинарное и двойное подчеркивание перед именем объекта?
69. Каков результат нижеприведенного запроса?
70. Напишите программу на Python, чтобы создать треугольник из звезд
71. Что плохого в следующем коде:
72. Как разделить строку на список?
73. Напишите программу на Python, чтобы проверить, является ли последовательность, которую вы вводите, палиндромом
74. Что такое генератор?
75. Напишите программу на Python для проверки, является ли число простым
76. Какова цель одинарной переменной подчеркивания (‘ _ ’) в Python?
77. Каковы типы наследования в Python?
78. Что такое распаковка кортежа?
79. Освобождается ли вся память при выходе из Python?
80. Является ли функция допустимой, если она не имеет оператора return?

Практические задания к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Вариант 1

1. Студент регистрируется на информационно-образовательном портале вуза. Составьте концептуальную модель главного цикла обработчика событий, который включает основные события, обработчик событий, вызываемые методы.
2. Сотрудник склада вводит информацию о поступивших товарах в информационную систему организации. Составьте концептуальную модель главного цикла обработчика событий, который включает основные события, обработчик событий, вызываемые методы.
3. Схематически представьте обработчик формирования заказа в магазине.
4. Основываясь на предыдущем задании, включите в него обработчик исключений. Опишите основные возможные исключения.
5. Производится оплата заказа через терминал. Составьте концептуальную модель главного цикла обработчика событий, которая включает основные события, обработчик событий, вызываемые методы.

Вариант 2

1. Создайте концептуальную модель GUI регистрации студента на образовательном портале. Определите размер главного окна.
2. Определите основные виджеты, которые должны быть размещены в главном окне.
3. Определите основные события, которые должен идентифицировать обработчик.

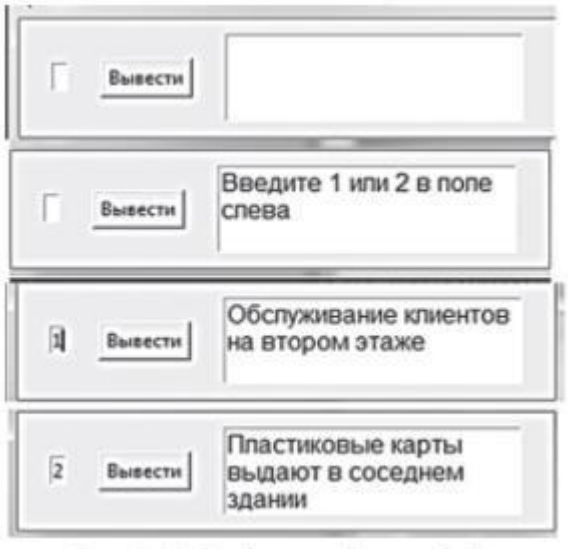
4. Какие действия должен совершить обработчик для событий, которые определены в предыдущем задании?
5. Создайте главное окно в соответствии с вашей концепцией из задания 1. Добавьте свойства окна следующими строчками pr ["bg"] ="Peachpuff" и pr.iconbitmap('star.ico').

Вариант 3

1. Создайте программу, которая бы формировала окно (см. рисунок). Используйте объектно-ориентированный стиль. Таблицу цветовых кодов можно найти в открытом доступе1.
2. Создайте программу, которая бы формировала следующее окно (рис. 5.24). После ввода цифры и нажатия на кнопку «вывести» в соответствующем окне должна отобразиться справочная информация.
3. Создайте приложение, в котором меняется размер фрейма в зависимости от того, какая из трех объектов-кнопок была нажата.
4. Создайте программу с меню, содержащим два пункта: Цвет и Размер. Пункт «Цвет» должен содержать три команды (Красный, Синий и Зеленый), меняющие цвет рамки на главном окне. Пункт «Размер» должен содержать две команды (500×500 и 700×400), изменяющие размер рамки.
5. Разработайте программу с графическим интерфейсом, которая формирует список студентов, выводит размер их стипендии, считает среднее значение стипендии и средний балл за три дисциплины. Должно быть три группы разных направлений, не менее пяти студентов в каждой
группе. Должна быть реализована выборка по группам, дисциплине и наличию либо отсутствию стипендии.

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
from tkinter import*
import math
#
tk=Tk()
tk.title("Graph")
#
button=Button(tk,text='Закрыть', command=quit)
button.pack()
#
canvas=Canvas(tk)
canvas["height"]=360
canvas["width"]=480
canvas["background"]="#eeeeff"
canvas["borderwidth"]=2
canvas.pack()
#
canvas.create_text(20,10,text="20,10")
canvas.create_text(460,350,text="460,350")
#
points=[]
ay=150
y0=150
x0=50
x1=470
dx=10
#
for n in range(x0,x1,dx):
    y=y0-ay*math.cos(n*dx)
    pp=(n,y)
    points.append(pp)
#
canvas.create_line(points,fill="blue",smooth=1)
#
y_axe=[]
yy=(x0,0)
y_axe.append(yy)
yy=(x0,y0+ay)
y_axe.append(yy)
canvas.create_line(y_axe,fill="black",width=2)
#
x_axe=[]
xx=(x0,y0)
x_axe.append(xx)
xx=(x1,y0)
x_axe.append(xx)
canvas.create_line(x_axe,fill="black",width=2)
#
tk.mainloop()
```



Критерии оценки на экзамене

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает не точности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.