

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Гаранин Максим Александрович
 Должность: Ректор
 Дата подписания: 02.10.2025 11:09:58
 Уникальный программный ключ:
 7708e3a47e66a8ee02711b298d7c78bd1e40bf88

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПРИВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»

МОДУЛЬ "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ"

Программирование

рабочая программа дисциплины (модуля)

Направление подготовки 09.03.03 Прикладная информатика
 Направленность (профиль) Управление цифровой инфраструктурой организации
 Квалификация **бакалавр**
 Форма обучения **очная**
 Общая трудоемкость **10 ЗЕТ**

Виды контроля в семестрах:
 экзамены 2
 зачеты 1

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		2 (1.2)		Итого	
Неделя	16 1/6		16 1/6			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	32	32	32	32	64	64
Лабораторные	32	32	32	32	64	64
Конт. ч. на аттест. в период ЭС	0,15	0,15	2,3	2,3	2,45	2,45
Итого ауд.	64	64	64	64	128	128
Контактная работа	64,15	64,15	66,3	66,3	130,45	130,45
Сам. работа	107	107	89	89	196	196
Часы на контроль	8,85	8,85	24,7	24,7	33,55	33,55
Итого	180	180	180	180	360	360

Программу составил(и):

к.п.н., доцент, Тюжина И.В.

Рабочая программа дисциплины

Программирование

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана: 09.03.03-25-2-ПИБ.plm.plx

Направление подготовки 09.03.03 Прикладная информатика Направленность (профиль) Управление цифровой инфраструктурой организации

Рабочая программа одобрена на заседании кафедры

Цифровые технологии

Зав. кафедрой к.э.н., доцент Ефимова Т.Б.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1	формирование у бакалавров систематизированных знаний и навыков в области программирования, ознакомление с различными парадигмами программирования.
-----	--

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОП:	Б1.О.13.01
-------------------	------------

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью

ОПК-4.1 Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-7.1 Разрабатывает алгоритмы и программы на языке программирования высокого уровня

В результате освоения дисциплины (модуля) обучающийся должен

3.1	Знать:
3.1.1	методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы;
3.1.2	стандарты и рекомендации по разработке программ и на языке высокого уровня;
3.2	Уметь:
3.2.1	реализовывать алгоритмы на языке высокого уровня; структурировать программы, составлять и вызывать подпрограммы;
3.2.2	находить и читать техническую документацию;
3.3	Владеть:
3.3.1	навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач; документирования исходного кода;

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1. Процедурное программирование			
1.1	Введение в программирование на Python: базовые концепции и синтаксис. /Лек/	1	2	
1.2	Понятие интерпретируемого языка программирования. Динамическая типизация. Оператор присваивания. Операторы ввода/вывода. /Лек/	1	2	
1.3	Основы синтаксиса. Знакомство со средой разработки. Основные математические операции. /Лаб/	1	2	
1.4	Числовые типы данных (int, float, complex). Основные операции с числами. Значения NaN и None. /Лек/	1	2	
1.5	Структура программы. Работа с числами. Вычисление по формулам. Ввод и вывод данных. /Лаб/	1	2	
1.6	Логический тип данных. Условный оператор. Конструкция if... elif... else. /Лек/	1	2	
1.7	Вложенные операторы. Составные условия. /Лек/	1	2	
1.8	Алгоритм ветвления. Условный оператор. Конструкция if... elif... else. /Лаб/	1	2	
1.9	Списки: объявление и обработка. Методы списков. /Лек/	1	2	
1.10	Списки. Методы списков. /Лаб/	1	2	
1.11	Индексы, срезы, глубокие и поверхностные копии. /Лек/	1	2	
1.12	Индексы, срезы, глубокие и поверхностные копии. /Лаб/	1	2	
1.13	Строковый тип данных: объявление и обработка. Методы строк. /Лек/	1	2	
1.14	Обработка строковых данных. /Лаб/	1	2	

1.15	Циклический алгоритм. Оператор цикла for. Инструкция range. /Лек/	1	2	
1.16	Оператор циклы. Решение типовых задач с использованием оператора цикла с известным числом повторений. /Лаб/	1	4	
1.17	Циклический алгоритм. Операторы цикла while. Операторы break и continue. /Лек/	1	2	
1.18	Оператор циклы. Решение типовых задач с использованием оператора цикла с предусловием. /Лаб/	1	4	
1.19	Обработка основных исключений в Python. Иерархия исключений. /Лек/	1	2	
1.20	Обработка основных исключений в Python. Иерархия исключений. /Лаб/	1	2	
1.21	Словари. Понятие ключ-значение. Методы словарей. /Лек/	1	2	
1.22	Словари. Понятие ключ-значение. Методы словарей. /Лаб/	1	2	
1.23	Работа с файлами. Открытие на чтение, запись и дозапись. Построчное чтение файлов. /Лек/	1	2	
1.24	Работа с файлами. Открытие на чтение, запись и дозапись. Построчное чтение файлов. /Лаб/	1	2	
1.25	Работа с библиотеками и сторонними модулями в Python. Модули math и random. /Лек/	1	2	
1.26	Регулярные выражения. Модуль re. Методы модуля re: match(), search(), findall(), split(), sub(), compile(). Специальные символы. /Лек/	1	2	
1.27	Регулярные выражения. Модуль re. Методы модуля re: match(), search(), findall(), split(), sub(), compile(). Специальные символы. /Лаб/	1	2	
1.28	Основы алгоритмизации. Структура программы. /Лек/	1	2	
1.29	Комплексная работа. Игры с текстовым интерфейсом. /Лаб/	1	4	
	Раздел 2. Функциональное программирование			
2.1	Подпрограммы. Именные функции. Инструкции def и return. /Лек/	2	2	
2.2	Области видимости и глобальные переменные. /Лек/	2	2	
2.3	Программирование с использованием функций. /Лаб/	2	2	
2.4	Анонимные функции, инструкция lambda. /Лек/	2	2	
2.5	Рекурсия в программировании. /Лек/	2	2	
2.6	Рекурсия. Разработка программ с использованием функций. /Лаб/	2	4	
2.7	Программы с графическим интерфейсом. Библиотека Tkinter. Параметры окна. Виджеты. Кнопка, текстовое поле. /Лек/	2	2	
2.8	Программы с графическим интерфейсом. Библиотека Tkinter. Параметры окна. Виджеты. Кнопка, текстовое поле. /Лаб/	2	2	
2.9	Программы с графическим интерфейсом. Библиотека Tkinter. Виджеты. Надпись, многострочное текстовое поле, селекторы. /Лек/	2	2	
2.10	Программы с графическим интерфейсом. Библиотека Tkinter. Виджеты. Надпись, многострочное текстовое поле, селекторы. /Лаб/	2	2	
2.11	Программы с графическим интерфейсом. Библиотека Tkinter. Менеджеры геометрии. Позиционирование. /Лек/	2	2	
2.12	Программы с графическим интерфейсом. Библиотека Tkinter. Менеджеры геометрии. Позиционирование. /Лаб/	2	2	
2.13	Программы с графическим интерфейсом. Работа с изображениями. Всплывающие сообщения. Многооконные приложения. Компиляция приложения. /Лек/	2	2	
2.14	Программы с графическим интерфейсом. Работа с изображениями. Всплывающие сообщения. Многооконные приложения. Компиляция приложения. /Лаб/	2	2	
2.15	Программы с графическим интерфейсом. Библиотека Tkinter. Обработка событий клавиатуры и мыши. Метод bind. /Лек/	2	2	
2.16	Программы с графическим интерфейсом. Библиотека Tkinter. Обработка событий клавиатуры и мыши. Метод bind. /Лаб/	2	2	

2.17	Программы с графическим интерфейсом. Комплексная лабораторная работа. /Лаб/	2	4	
2.18	Модуль Requests. GET-запросы. /Лек/	2	2	
2.19	Модуль Requests. GET-запросы. /Лаб/	2	2	
2.20	Библиотекой BeautifulSoup для парсинга HTML. /Лек/	2	2	
2.21	Библиотека BeautifulSoup. Навигация по дереву. /Лек/	2	2	
2.22	Библиотека BeautifulSoup. Навигация по дереву. Навигация с использованием имен тегов. Проход снизу вверх. /Лаб/	2	2	
2.23	Библиотека BeautifulSoup. Поиск по дереву. /Лек/	2	4	
2.24	Библиотека BeautifulSoup. Поиск по дереву. Фильтры. /Лаб/	2	4	
2.25	Парсинг структурированных данных: таблицы, списки, формы. /Лек/	2	2	
2.26	Парсинг структурированных данных: таблицы, списки, формы /Лаб/	2	2	
2.27	Библиотека BeautifulSoup. Парсинг изображений и файлов /Лек/	2	2	
2.28	Библиотека BeautifulSoup. Парсинг изображений и файлов /Лаб/	2	2	
Раздел 3. Самостоятельная работа				
3.1	Роль программирования в различных сферах деятельности, используемые программные средства в экономике, бизнесе /Ср/	1	12	
3.2	Обзор языков высокого уровня. Их сравнительная характеристика /Ср/	1	12	
3.3	PEP8, рекомендации по написанию кода на Python /Ср/	1	12	
3.4	Машиночитаемые форматы файлов, csv, Json. /Ср/	1	11	
3.5	Разработка и реализация рекурсивных алгоритмов /Ср/	1	14	
3.6	Подготовка к лекциям /Ср/	1	16	
3.7	Подготовка к лабораторным. /Ср/	1	30	
3.8	Работа с учебной литературой, поиск идей, алгоритмов и методов на специализированных форумах программистов /Ср/	2	25	
3.9	Автоматическое тестирование. Библиотека Selenium. Изучение материалов онлайн курса. /Ср/	2	20	
3.10	Библиотека BeautifulSoup. Дерево синтаксического разбора. Чтение данных с сайтов в автоматическом формате. /Ср/	2	10	
3.11	Подготовка к лекциям /Ср/	2	6	
3.12	Подготовка к лабораторным /Ср/	2	20	
3.13	Разработка усложненных программ со структурированными типами данных. /Ср/	2	8	
Раздел 4. Контактные часы на аттестацию				
4.1	Зачёт /КЭ/	1	0,15	
4.2	Экзамен /КЭ/	2	2,3	

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценочные материалы для проведения промежуточной аттестации обучающихся приведены в приложении к рабочей программе дисциплины.

Формы и виды текущего контроля по дисциплине (модулю), виды заданий, критерии их оценивания, распределение баллов по видам текущего контроля разрабатываются преподавателем дисциплины с учетом ее специфики и доводятся до сведения обучающихся на первом учебном занятии.

Текущий контроль успеваемости осуществляется преподавателем дисциплины (модуля) в рамках контактной работы и самостоятельной работы обучающихся. Для фиксирования результатов текущего контроля может использоваться ЭИОС.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
6.1. Рекомендуемая литература				
6.1.1. Основная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Чернышев С. А.	Основы программирования на Python: учебное пособие для вузов	Москва: Юрайт, 2022	osnovy-programmirovani
6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Орлов С. А.	Теория и практика языков программирования: учебник для бакалавров и магистров. Стандарт третьего поколения	Санкт-Петербург: Питер, 2013	
Л2.2	Иванова Г.С.	Технология программирования	Москва: КноРус, 2018	://www.book.ru/book/926
6.2 Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине (модулю)				
6.2.1 Перечень лицензионного и свободно распространяемого программного обеспечения				
6.2.1.1	Colaboratory			
6.2.1.2	IDLE			
6.2.2 Перечень профессиональных баз данных и информационных справочных систем				
6.2.2.1	База книг и публикаций Электронной библиотеки "Наука и Техника"- http://www.n-t.ru			
6.2.2.2	Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки- https://github.com/			
6.2.2.3	Портал для разработчиков электронной техники: http://www.espec.ws/			
6.2.2.4	База данных «Библиотека программиста» https://proglib.io/			
6.2.2.5	Консультант плюс			
6.2.2.6	Информационная система ГАРАНТ			
7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
7.1	Учебные аудитории для проведения занятий лекционного типа, укомплектованные специализированной мебелью и техническими средствами обучения: мультимедийное оборудование для предоставления учебной информации большой аудитории и/или звукоусиливающее оборудование (стационарное или переносное).			
7.2	Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения: мультимедийное оборудование и/или звукоусиливающее оборудование (стационарное или переносное)			
7.3	Помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета.			
7.4	Помещения для хранения и профилактического обслуживания учебного оборудования			
7.5	Помещения для курсового проектирования / выполнения курсовых работ, укомплектованные специализированной мебелью и техническими средствами обучения (стационарными или переносными).			

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

ПРОГРАММИРОВАНИЕ

Направление подготовки / специальность

09.03.03 Прикладная информатика

Направленность (профиль)/специализация

Управление цифровой инфраструктурой организации

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: зачёт - 1 семестр, экзамен – 2 семестр

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ОПК-4: Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1: Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1: Разрабатывает алгоритмы и программы на языке программирования высокого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы(семестр 1)
ОПК-4.1: Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы	Обучающийся знает: стандарты и рекомендации по разработке программ и на языке высокого уровня;	Задания(№1-10)
	Обучающийся умеет: находить и читать техническую документацию;	Задания (№41-50)
	Обучающийся владеет: навыками документирования исходного кода;	Задания (№55-62)
ОПК-7.1: Разрабатывает алгоритмы и программы на языке программирования высокого уровня	Обучающийся знает: методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы;	Задания(№11-40)
	Обучающийся умеет: реализовывать алгоритмы на языке высокого уровня; структурировать программы, составлять и вызывать подпрограммы;	Задания (№51-55)
	Обучающийся владеет: навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач.	Задания (№55-62)

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) собеседование;
- 2) выполнение заданий в ЭИОС университета.

2. Типовые¹ контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ОПК-4.1: Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы	Обучающийся знает: стандарты и рекомендации по разработке программ и на языке высокого уровня;
ОПК-7.1: Разрабатывает алгоритмы и программы на языке программирования высокого уровня	Обучающийся знает: методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы;

Задание №1

Выберите верные утверждения об отступах в языке Python:

Выберите несколько из 4 вариантов ответа:

- | | |
|----|--|
| 1) | количество пробелов в отступах произвольно и выбирается каждым на свое усмотрение, однако по договоренности равняется четырем пробелам |
| 2) | количество пробелов в отступах произвольно и выбирается каждым на свое усмотрение, однако по договоренности равняется двум пробелам |
| 3) | отступ внутри блока должен быть одинаковым |
| 4) | количество пробелов в отступах строго фиксировано и равно четырем |

Задание №2

Укажите валидные участки кода (язык программирования Python):

Выберите один из 4 вариантов ответа:

- | | |
|----|-------------|
| 1) | While True: |
| 2) | while true: |
| 3) | while True |
| 4) | while True: |

Задание №3

Отметьте верные утверждения о документировании кода на Python

Выберите несколько из 4 вариантов ответа:

- | | |
|----|---|
| 1) | Код документируется с помощью знака # поставленного перед строкой комментария |
| 2) | Код документируется с помощью знака # поставленного после строки комментария |
| 3) | Строка комментария должна располагаться между знаков # |
| 4) | Знак # предназначен для однострочных комментариев |

¹Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

Задание №4

Что вернет запись

```
>>> a = [2,4,6,7]
```

```
>>> a[-3]
```

Выберите один из 4 вариантов ответа:

- | | |
|----|-------------------------------------|
| 1) | 4 |
| 2) | IndexError: list index out of range |
| 3) | 2 |
| 4) | 3 |

Задание №5

Что такое модуль в Python?

Выберите один из 4 вариантов ответа:

- | | |
|----|---|
| 1) | Файл, содержащий код на Python. Может быть импортирован в другие файлы. |
| 2) | Библиотека, предоставляющая функции и классы для использования в программе. |
| 3) | Компонент программы, выполняющий определённую задачу. |
| 4) | Часть программы, содержащая объявления переменных и констант. |

Задание №6

Что выведет следующий код, при его исполнении? `kvps = {"user","bill", "password","hillary"}; print(kvps['password'])`

Выберите один из 4 вариантов ответа:

- | | |
|----|--------------------------|
| 1) | user |
| 2) | hillary |
| 3) | password |
| 4) | Ничего. Возникнет ошибка |

Задание №7

Опишите какие имена следует давать классам согласно рекомендациям PEP8?

Типовой ответ:

Имена классов должны использовать стиль написания CamelCase, где каждое слово начинается с заглавной буквы, а пробелы между словами отсутствуют. Например: • MyClass • EmployeeRecord • DataProcessor
В именах классов не рекомендуется использовать символы подчеркивания (нижнее подчеркивание), за исключением случаев, когда это необходимо для ясности (например, если класс является частью определённого пространства имён).
Имена классов должны быть понятными и описательными, чтобы отражать их предназначение и функциональность.

Задание №8

Укажите какие однобуквенные идентификаторы не рекомендует использовать PEP8?

Типовой ответ:

Никогда не используйте символы `l` (маленькая латинская буква «эль»), `O` (заглавная латинская буква «о») или `I` (заглавная латинская буква «ай») как однобуквенные идентификаторы.
В некоторых шрифтах эти символы неотличимы от цифры один и нуля. Если очень нужно `l`, пишите вместо неё заглавную `L`.

Задание №9

Что такое PEP (Python Enhancement Proposal)?

Типовой ответ:

PEP — это документ, содержащий рекомендации и правила, которым стоит следовать, чтобы ваш код был легко читаемым, понятным и единообразным.

Задание №10

Какие типы данных существуют в Python? Назовите как минимум 5 типов данных.

Типовой ответ:

целые числа, числа с плавающей точкой, строки, списки, кортежи, множества, словари и логические значения.

Задание №11

Укажите варианты кода, позволяющие получить слово «ход» из списка `m=[["пароход"], "доход"]`

Выберите несколько из 4 вариантов ответа:

1)	<code>m[0][0][-3:]</code>
2)	<code>m[-1][-3:]</code>
3)	<code>m[-1][0][-3:]</code>
4)	<code>m[-1][0][2:]</code>

Задание №12

Укажите варианты срезов, позволяющие получить слово «дрон» из строки `st="норд-вест"`

Выберите один из 4 вариантов ответа:

1)	<code>st[3::-1]</code>
2)	<code>st[:4:-1]</code>
3)	<code>st[0:3:-1]</code>
4)	<code>st[3:0:-1]</code>

Задание №13

`st = 'миру мир'`

Укажите варианты, которые вернут строку 'рим'?

Выберите несколько из 4 вариантов ответа:

1)	<code>st[2::-1]</code>
2)	<code>st[:4:-1]</code>
3)	<code>st[-3::-1]</code>
4)	<code>st[-1:-3:-1]</code>

Задание №14

Укажите верные инструкции для создания кнопки `but`:

Выберите один из 4 вариантов ответа:

1)	<code>but = Button(root, text="Кнопка")</code>
2)	<code>but.Button(root, text="Кнопка")</code>
3)	<code>but.Button(root, text="Кнопка")</code>
4)	<code>but.bind('<Button-1>', str_to_sort_list)</code>

Задание №15

Укажите виджет однострочного текстового поля в библиотеке `tkinter`:

Выберите один из 4 вариантов ответа:

1)	<code>Entry</code>
2)	<code>Label</code>
3)	<code>Button</code>
4)	<code>Text</code>

Задание №16

Укажите единицы измерения свойства виджета `width`:

Выберите один из 4 вариантов ответа:

1)	пиксели
----	---------

2)		дюймы
3)		знакоместа
4)		миллиметры

Задание №17

Укажите инструкции, отвечающие за задание размеров окна root 400 на 500 пикселей.

Выберите один из 4 вариантов ответа:

1)		root.geometry("400x500")
2)		root.place(400,500)
3)		root.geometry(400,500)
4)		root.geometry("200x300+400+500")

Задание №18

Укажите инструкции, отвечающие за создание надписи на кнопке but:

Выберите несколько из 4 вариантов ответа:

1)		but = Button(text = "Кнопка")
2)		but['text'] = "Кнопка"
3)		but.text="Кнопка"
4)		but[text]=Кнопка

Задание №19

Укажите инструкции, устанавливающие связь между нажатием на кнопку but исполнением функции printer:

Выберите несколько из 4 вариантов ответа:

1)		but.bind('<Button-1>', printer)
2)		but.bind(<Button-1>, printer())
3)		but = Button(text="Кнопка", command=printer)
4)		but.Button(text="Кнопка", event=printer())

Задание №20

Укажите инструкцию, отвечающую за создание окна в библиотеке Tkinter:

Выберите один из 4 вариантов ответа:

1)		root = Tk()
2)		root = Tk
3)		root = Window()
4)		root = Win

Задание №21

Укажите корректные способы задания свойства font:

Выберите несколько из 4 вариантов ответа:

1)		font = "Arial 16"
2)		font = ("Comic Sans MS", 24, "bold")
3)		font = Times New Roman
4)		font = Arial, 14

Задание №22

Укажите корректные способы задать цвет фона во время нажатия и установки курсора мыши над кнопкой b1

Выберите несколько из 4 вариантов ответа:

1)		b1['activebackground'] = '#555555'
2)		b1['activebackground'] = 'red'
3)		b1['activebackground'] = (255,0,0)
4)		b1['activeforeground'] = red

Задание №23

Что будет выведено на экран в результате выполнения следующего участка кода `a = [1,2, None(),[.]]; print(len(a))`

Выберите один из 4 вариантов ответа:

1)		3
2)		5
3)		Syntax Error
4)		Infinity

Задание №24

Что будет выведено на экран в результате выполнения следующего участка кода `a=6.7; b = a // 2; print(b)`

Выберите один из 4 вариантов ответа:

1)		3;
2)		3.35
3)		Syntax Error
4)		0.7

Задание №25

Что будет выведено на экран в результате выполнения следующего участка кода `x = [1, 2, 7]; y = x; y[-1] = ['a', 'b']; print x`

Выберите один из 4 вариантов ответа:

1)		[1, 2, 'a', 'b'];
2)		[1, 2, ['a', 'b']];
3)		['a', 'b', 1, 2];
4)		[[1, 2], 'a', 'b']?

Задание №26

Что будет являться результатом выполнения следующих строк кода `b=6; a=b=8; print(a)`

Выберите один из 4 вариантов ответа:

1)		6
2)		8
3)		False
4)		True

Задание №27

Что будет являться результатом выполнения следующих строк кода: `x = input('введите число')` <перенос строки>
`print(type(x))`

Выберите один из 4 вариантов ответа:

1)		<class 'int'>;
2)		<class 'float'>;
3)		<class 'str'>;
4)		класс переменной будет зависеть от введенных данных?

Задание №28

Что вернет запись
`>>> st = 'абракадабра'`
`>>> st[1:4]`

Выберите один из 4 вариантов ответа:

1)		'брак'
2)		'бра'
3)		'абр'
4)		'абра'

Задание №29

Что вернет запись

```
>>> st='Привет'
>>> st[::-1]
```

Выберите один из 4 вариантов ответа:

- | | | |
|----|--|----------|
| 1) | | 'тевирП' |
| 2) | | 'Т' |
| 3) | | 'П' |
| 4) | | 'Приве' |

Задание №30

Что вернет запись
>>> t = ('a','b','c','d')
>>> t[3:]

Выберите один из 4 вариантов ответа:

- | | | |
|----|--|-----------|
| 1) | | ('d',) |
| 2) | | 'd' |
| 3) | | 'c','d' |
| 4) | | ('c','d') |

Задание №31

S='программа'. Каким будет результат выполнения команды S[::-1]

Выберите один из 4 вариантов ответа:

- | | | |
|----|--|---|
| 1) | | ['п', 'р', 'о', 'г', 'р', 'а', 'м', 'м', 'а'] |
| 2) | | 'аммаргорп' |
| 3) | | ‘программ’ |
| 4) | | ‘рограмма’ |

Задание №32

s="каракатица" .
Укажите участок кода, который вернет строку "рак".

Выберите несколько из 4 вариантов ответа:

- | | | |
|----|--|--------|
| 1) | | s[2:5] |
| 2) | | s[3:6] |
| 3) | | s[2:6] |
| 4) | | s[3:5] |

Задание №33

Имеются строки s1='вино' и s2='град'. Отметьте верные утверждения.

Выберите несколько из 4 вариантов ответа:

- | | | |
|----|--|-----------------------------------|
| 1) | | s1 + s2 == s2 + s1 |
| 2) | | len (s1) == len(s2) |
| 3) | | s1[0] == s2[-1] |
| 4) | | len(s1 + s2) == len(s1) + len(s2) |

Задание №34

Инструкция list задает

Выберите один из 4 вариантов ответа:

- | | | |
|----|--|-----------|
| 1) | | кортеж |
| 2) | | список |
| 3) | | множество |
| 4) | | строку |

Задание №35

Инструкция tuple задает

Выберите один из 4 вариантов ответа:

- | | |
|----|-----------|
| 1) | кортеж |
| 2) | список |
| 3) | множество |
| 4) | строку |

Задание №36

Какая из следующих конструкций вернёт длину слова s

Выберите один из 4 вариантов ответа:

- | | |
|----|------------|
| 1) | s.length |
| 2) | s.length() |
| 3) | s.len |
| 4) | len(s) |

Задание №37

Каков результат выполнения следующего кода на языке Python: a='privet'; print(a[-1]**2)?

Выберите один из 4 вариантов ответа:

- | | |
|----|------------------|
| 1) | 'tt' |
| 2) | 'ttttttt' |
| 3) | 't' |
| 4) | Возникнет ошибка |

Задание №38

Отметьте верные варианты импорта графической библиотеки Python

Выберите несколько из 6 вариантов ответа:

- | | |
|----|-----------------------|
| 1) | Import tk |
| 2) | from tkinter import * |
| 3) | import tkinter |
| 4) | export tkinter |
| 5) | from * import tk |
| 6) | import tkinter as tk |

Задание №39

Равенство в Python обозначается конструкцией

Выберите один из 4 вариантов ответа:

- | | |
|----|-----|
| 1) | <> |
| 2) | = = |
| 3) | != |
| 4) | =! |

Задание №40

Укажите варианты кода, позволяющие получить слово «кум» из списка m=[“документ”]

Выберите несколько из 4 вариантов ответа:

- | | |
|----|----------------|
| 1) | m[0][2:5] |
| 2) | m[-1][2:-3] |
| 3) | m[2:5] |
| 4) | m[-1][0][2:-3] |

Ответы:

- | | |
|-----------|-----|
| #1 (1 б.) | 1,3 |
| #2 (1 б.) | 4 |

#3 (1 б.)	1,4
#4 (1 б.)	1
#5 (1 б.)	1
#6 (1 б.)	4
#7 (эссе)	Типовой ответ: Имена классов должны использовать стиль написания CamelCase, где каждое слово начинается с заглавной буквы, а пробелы между словами отсутствуют. Например: • MyClass • EmployeeRecord • DataProcessor В именах классов не рекомендуется использовать символы подчеркивания (нижнее подчеркивание), за исключением случаев, когда это необходимо для ясности (например, если класс является частью определённого пространства имён). Имена классов должны быть понятными и описательными, чтобы отражать их предназначение и функциональность.
#8 (эссе)	Типовой ответ: Никогда не используйте символы l (маленькая латинская буква «эль»), O (заглавная латинская буква «о») или I (заглавная латинская буква «ай») как однобуквенные идентификаторы. В некоторых шрифтах эти символы неотличимы от цифры один и нуля. Если очень нужно l, пишите вместо неё заглавную L.
#9 (эссе)	Типовой ответ: PEP — это документ, содержащий рекомендации и правила, которым стоит следовать, чтобы ваш код был легко читаемым, понятным и единообразным.
#10 (эссе)	Любые два из предложенных: целые числа, числа с плавающей точкой, строки, списки, кортежи, множества, словари и логические значения.
#11 (1 б.)	1, 2
#12 (1 б.)	1
#13 (1 б.)	1, 2
#14 (1 б.)	1
#15 (1 б.)	1
#16 (1 б.)	3
#17 (1 б.)	1
#18 (1 б.)	1, 2
#19 (1 б.)	1, 3
#20 (1 б.)	1
#21 (1 б.)	1, 2
#22 (1 б.)	1, 2
#23 (1 б.)	2
#24 (1 б.)	1
#25 (1 б.)	2
#26 (1 б.)	2
#27 (1 б.)	3
#28 (1 б.)	2
#29 (1 б.)	1
#30 (1 б.)	1
#31 (1 б.)	2
#32 (1 б.)	1
#33 (1 б.)	2, 4
#34 (1 б.)	2
#35 (1 б.)	1
#36 (1 б.)	4
#37 (1 б.)	4
#38 (1 б.)	2, 3, 6
#39 (1 б.)	2
#40 (1 б.)	1, 2

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ОПК-4.1: Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы	Обучающийся умеет: находить и читать техническую документацию;
ОПК-7.1: Разрабатывает алгоритмы и программы на языке программирования высокого уровня	Обучающийся умеет: реализовывать алгоритмы на языке высокого уровня; структурировать программы, составлять и вызывать подпрограммы;

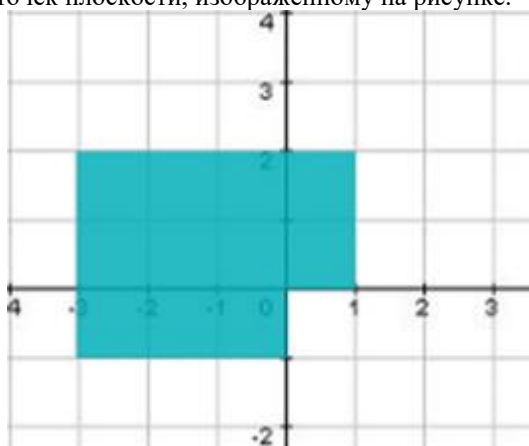
Задание 41

Составьте программу табулирования функции (вычисления всех её значений на отрезке $[-6, 4]$ с шагом 2.

$$y = \frac{1}{x^3 - 1}$$

Задание 42.

На языке программирования Python составьте программу, определяющую принадлежит ли точка с координатами (X,Y) множеству точек плоскости, изображенному на рисунке.



Задание 43.

Составьте программу вычисления кусочно-заданной функции. Проверьте работоспособность программы на каждом из трех отрезков.

$$(x) = \begin{cases} -10x, & x < -5 \\ x^2 + x^3, & -5 \leq x \leq 5 \\ \sqrt{\frac{x-23}{4}}, & x > 5 \end{cases}$$

Задание 44.

Напишите программу, которая принимает строку и возвращает новую строку, в которой удалены все дубликаты символов, сохраняя порядок их появления.

Задание 45.

Напишите программу, которая принимает строку и возвращает новую строку, состоящую из первых букв каждого слова в исходной строке.

Пример:

```
input_str = "Hello World from Python"
```

```
output_str = "HWfP"
```

Задание 46.

Напишите программу, которая принимает строку и возвращает словарь с частотой появления каждого символа в строке.

Пример:

```
input_str = "hello"
```

```
output = {'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

Задание 47.

Создайте список из 20 случайных целых чисел от 1 до 10 (числа могут повторяться). Напишите программу, которая удаляет все дубликаты из списка и возвращает новый список.

Задание 48.

Создайте список из 5 строковых значений. Напишите программу, которая переворачивает каждую строку в списке. Выведите оригинальный список и список с перевернутыми строками.

Задание 49.

Не используя подключаемые модули (в том числе модуль `math`), составьте программу вычисления суммы:

$$\sum_{i=1}^n \frac{i!}{(n+1)!}$$

Задание 50.

Не используя подключаемые модули (в том числе модуль `math`), напишите функцию `max2`, принимающую два аргумента и возвращающую наибольшее из них. Примените эту функцию для поиска наибольшего из чисел `x`, `y` и `x*y`. `x` и `y` вводит пользователь.

Задание 51.

Опишите какую информацию может содержать документация класса?

Задание 52.

Найдите спецификацию на модуль `math` языка Python. Изучите её и ответьте на следующие вопросы: для чего предназначен этот модуль? Перечислите не менее 5 функций модуля.

Задание 53.

Найдите документацию на библиотеку `tkinter` языка Python. Изучите её и ответьте на следующие вопросы: для чего предназначена эта библиотека? Перечислите не менее 4 виджетов и не менее 5 методов библиотеки.

Задание 54.

Получить доступ к встроенной справочной системе `python`. Укажите функцию.

Типовой ответ: функция `help`.

Задание 55.

Для чего предназначена библиотека `pydoc`. Приведите примеры её использования.

ОПК-4.1: Применяет стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы

Обучающийся владеет: навыками документирования исходного кода;

ОПК-7.1: Разрабатывает алгоритмы и программы на языке программирования высокого уровня

Обучающийся владеет: навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач.

Задание 56

Напишите функцию `is_mutually_simple`, принимающую 2 целых числа, и возвращающую `True`, если они взаимно простые, и `False` - иначе. Взаимно простые числа, это числа, у которых наибольший общий делитель 1. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать

Задание 57.

Напишите функцию, принимающую на вход список `L` и число `N`. Функция должна возвращать объект, находящийся на `N`-ом месте в списке. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать

Задание 58.

Составьте программу, которая построчно читает информацию из файла `X`, и перезаписывает в файл `Y` строки файла `X` с указанием их длины. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать

Задание 59.

Составьте программу, которая выводит на экран слова текстового файла, начинающиеся с буквы, введенной пользователем. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать

Задание 60.

Напишите функцию `count_of_composite`, принимающую 1 аргумент – целое положительное число, и возвращающую количество составных делителей этого числа. Например, у числа 12 три составных делителя – 4, 6, и 12. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать.

Задание 61.

Напишите функцию `my_mean` принимающую на вход три элемента и возвращающую их среднее арифметическое значение, первый и второй элемент по умолчанию равны 10 и 12, соответственно. Запустите функцию для следующих входных наборов 6,10, 4; 34, 1.5, 12; 13; 5,6. Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать.

Задание 62.

Используя библиотеку `TKinter` создайте программу, в которой по нажатию на кнопку последовательно меняется её цвет: красный, оранжевый, жёлтый, зеленый, голубой, синий, фиолетовый (цвет кнопки задается параметром `bg`). Используйте комментарии, которые объясняют, как работает фрагмент кода, что он делает и как его использовать.

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

1. Что такое Python и для чего он используется?
2. Какие основные типы данных существуют в Python?

3. Как создать переменную в Python?
4. Как выполнить операцию присваивания в Python?
5. Как вывести значение переменной на экран в Python?
6. Каково предназначение библиотеки Tkinter?
7. Какие основные компоненты графического интерфейса можно создать с помощью библиотеки Tkinter?
8. Как создать главное окно приложения с помощью библиотеки Tkinter?
9. Как добавить виджеты (кнопки, поля ввода и т. д.) в окно приложения?
10. Как изменить размер, цвет, шрифт и другие свойства виджетов?
11. Как обрабатывать события, происходящие с виджетами (нажатие кнопки, ввод текста и т. п.)?
12. Как организовать взаимодействие между виджетами?
13. Как сохранять данные приложения в файл и загружать их из файла?
14. Как сделать приложение многооконным?
15. Как настроить внешний вид окна приложения (цвет фона, рамки и т.д.)?
16. Как реализовать проверку вводимых данных?
17. Как использовать модули для расширения функциональности приложения?
18. Как протестировать приложение перед его использованием?
19. Что такое функция в Python и как её создать?
20. Что такое цикл for в Python и как его использовать?
21. Что такое список в Python и какие операции можно выполнять со списками?
22. Что такое словарь в Python и чем он отличается от списка?
23. Что такое кортеж в Python и зачем он нужен?
24. Что такое исключение в Python и как обработать исключение?
25. Что такое рекурсия в Python и когда она используется?
26. Что такое лямбда-функция в Python и зачем она нужна?
27. Что такое PEP (Python Enhancement Proposal) и для чего они создаются?
28. Что такое документация в Python и как с ней работать?

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объёма заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*
- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*
- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

Критерии формирования оценок по зачету с оценкой

«Отлично/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал

навык практического применения полученных знаний, не допустил логических и фактических ошибок
«Хорошо/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

«Удовлетворительно/зачтено» – студент допустил существенные ошибки.

«Неудовлетворительно/не зачтено» – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*