

Федеральное государственное образовательное бюджетное учреждение  
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ  
РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**(Финансовый университет)**

**АЛТАЙСКИЙ ФИЛИАЛ**

**Кафедра «Учет и информационные технологии в бизнесе»**

**Д.А. Жевнов**

**ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

ОП «Прикладные информационные системы в экономике и финансах»

Барнаул 2022

Федеральное государственное образовательное бюджетное учреждение  
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ  
РОССИЙСКОЙ ФЕДЕРАЦИИ»  
(Финансовый университет)**

**АЛТАЙСКИЙ ФИЛИАЛ**

**Кафедра «Учет и информационные технологии в бизнесе»**

УТВЕРЖДАЮ

Директор филиала



Иванова В.А.

«26» апреля 2022 г.

**Д.А. Жевнов**

**ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

ОП «Прикладные информационные системы в экономике и финансах»

*Рекомендовано Ученым Советом Алтайского филиала*

*(протокол №48 от «26» апреля 2022 г.)*

*Одобрено кафедрой «Учет и информационные технологии в бизнесе»*

*(протокол №9 от «31» марта 2022 г.)*

Барнаул 2022

**Рецензент:** **Богданова М.М.**, к.э.н., доцент кафедры «Учет и информационные технологии в бизнесе» Алтайского филиала Финуниверситета.

**Жевнов Д.А. Практикум по программированию.** Рабочая программа дисциплины для студентов бакалавриата, обучающихся по направлению подготовки 09.03.03 «Прикладная информатика». - Барнаул: Алтайский филиал Финуниверситета, 2022 – 35 с.

В рабочей программе дисциплины определены ее цель, место в структуре ОП, требования к результатам освоения дисциплины, содержание программы, тематика практических занятий, формы самостоятельной работы, оценочные средства для текущего контроля и промежуточной аттестации, учебно-методическое и информационное обеспечение.

*Учебное издание*

**Жевнов Денис Анатольевич**

## **ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ**

### **Рабочая программа дисциплины**

Рабочая программа дисциплины

Компьютерный набор, верстка: Жевнов Д.А.

Формат 60х90/16. Гарнитура *Times New Roman*

Усл.п.л.1,9 Изд. № \_\_\_\_ Тираж \_\_\_\_ экз.

Отпечатано в Алтайском филиале Финуниверситета

© Жевнов Денис Анатольевич, 2022

© Алтайский филиал Финуниверситета, 2022

## Содержание

1. Наименование дисциплины.....	4
2. Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.....	4
3. Место дисциплины в структуре образовательной программы.....	5
4. Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	7
5.1. Содержание дисциплины.....	7
5.2. Учебно-тематический план.....	9
5.3. Содержание семинаров, практических занятий.....	10
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю.....	15
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	17
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.....	32
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	33
10. Методические указания для обучающихся по освоению дисциплины.....	33
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем.....	34
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	35

1.....Наименование дисциплины

«Практикум по программированию».

## 2. Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Дисциплина «Практикум по программированию» обеспечивает формирование следующих компетенций: ОПК-2, ОПК-3

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции <sup>1</sup>	Результаты обучения (владения <sup>2</sup> , умения и знания), соотнесенные с компетенциями/индикаторами и достижения компетенции
ОПК-2	Способность использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	1. Демонстрирует знания об основных информационных технологиях и программных средствах, позволяющих их использовать	<b>Знать</b> язык программирования Python, отличия версий языка, <b>Уметь</b> программировать на языке Python
		2. Рационально выбирает информационные технологии и реализующие их программные средства, в том числе, с учетом страны происхождения программных средств	<b>Знать</b> структуры данных языка Python, их особенности и сферы применения <b>Уметь</b> использовать различные структуры данных при решении задач профессиональной деятельности, в том числе, структуры данных из библиотек
		3. Использует современные информационные технологии и программные средства при решении задач разработки программного обеспечения для экономических и финансовых приложений	<b>Знать</b> основные библиотеки, используемые при решении экономических и финансовых задач <b>Уметь</b> использовать библиотеки в своих приложениях
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информации	1. Использует информационно-коммуникационные технологии и библиографические источники при поиске информации, для решения стандартных задач.	<b>Знать</b> основные источники информации по программированию на языке Python, в том числе литературу и ресурсы в сети Интернет <b>Уметь</b> использовать официальную документацию по языку и библиотекам Python

<sup>1</sup> Заполняется при реализации актуализированных ОС ВО ФУ и ФГОС ВО3++

<sup>2</sup> Владения формулируются только при реализации ОС ВО ФУ первого поколения и ФГОС ВО 3+

ной и библиографической культуры с применением информационных коммуникационных технологий и с учетом основных требований информационной безопасности.	2. Демонстрирует умение решать стандартные задачи разработки информационных систем.	<b>Знать.</b> основные принципы объектно-ориентированного проектирования приложений <b>Уметь</b> решать задачи профессиональной деятельности с использованием программирования
	3. Владеет навыками обеспечения информационной безопасности автоматизированных систем	<b>Знать:</b> приемы и подходы к разработке качественного программного обеспечения <b>Уметь:</b> анализировать программный код, выявлять и исправлять в нем ошибки

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Практикум по программированию» относится к Модулю математики и информатики (информационный модуль) направления подготовки 09.03.03 «Прикладная информатика» профиль: «ИТ-сервисы и технологии обработки данных в экономике и финансах».

Дисциплина «Практикум по программированию» базируется на знаниях, приобретаемых в рамках дисциплин «Алгоритмы и структуры данных в языке Python» и «Операционные системы семейства UNIX и сетевые технологии», которые студенты изучают в течение первого и второго года обучения.

Дисциплина «Практикум по программированию» служит для закрепления и углубления знаний и навыков в области программирования, получаемых при освоении других дисциплин, изучаемых в рамках направления подготовки бакалавров «Прикладная информатика».

### 4. Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

#### *Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах»*

Общая трудоёмкость дисциплины составляет 6 зачётных единиц.  
Вид промежуточной аттестации –зачет (1,2,3,4 семестр).

Вид текущего контроля – контрольные работы (2, 4 семестр).

*Очная, очно-заочная форма обучения*

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)	Семестр 3 (в часах)	Семестр 4 (в часах)
<b>Общая трудоемкость дисциплины</b>	6/216	54	54	54	54
<b>Контактная работа - Аудиторные занятия</b>	72	18	18	18	18
<i>Лекции</i>	-	-	-	-	-
<i>Семинары, практические занятия</i>	72	18	18	18	18
<b>Самостоятельная работа</b>	144	36	36	36	36
Вид текущего контроля	Контрольные работы	-	Контрольная работа	-	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет	Зачет	Зачет	Зачет

*Заочная форма обучения*

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 3 (в часах)	Семестр 4 (в часах)
<b>Общая трудоемкость дисциплины</b>	6/216	108	108
<b>Контактная работа - Аудиторные занятия</b>	24	12	12
<i>Лекции</i>	-	-	-
<i>Семинары, практические занятия</i>	24	12	12
<b>Самостоятельная работа</b>	192	96	96
Вид текущего контроля	Контрольные работы	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет	Зачет

## **5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий**

### **5.1. Содержание дисциплины**

#### **1. Основы языка Python**

Обработка числовой информации. Встроенные функции и модули для работы с числами. Реализация числовых алгоритмов с использованием инструкции ветвления и циклов.

Обработка текстовой информации. Функции и методы для работы со строками. Регулярные выражения.

Списки. Использование списков для хранения информации. Методы списков. Многомерные списки. Кортежи.

Словари. Типовые случаи использования словарей в программах. Методы для работы со словарями. Множества.

#### **2. Функции и модули**

Создание и использование функций. Функции с необязательными параметрами и переменным числом параметров. Анонимные функции. Глобальные и локальные переменные. Вложенные функции. Функции высшего порядка.

Создание и использование модулей и пакетов. Модули стандартной библиотеки.

#### **3. Обработка исключений. Работа с файлами средствами языка Python**

Понятие исключения. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Инструкции raise и assert. Инструкция with ... as.

Файлы. Использование текстовых файлов в программе. Двоичные файлы. Сохранение объектов в файл. Работа с файлами различных форматов: csv, docx, xlsx.

#### **4. Объектно-ориентированное программирование на Python**

Создание классов и объектов. Методы классов и статические переменные и методы. Специальные методы классов. Наследование и полиморфизм.

#### **5. Функциональное программирование на Python**

Понятие о функциональном программировании. Декораторы. Функции map, filter, reduce, any, all. Итераторы. Функции-генераторы.

#### **6. Алгоритмы и структуры данных**



Динамические массивы. Стеки, очереди, деки. Связные списки. Реализация связных списков на языке Python. Бинарные деревья. Использование бинарных деревьев в прикладных задачах. Двоичное дерево поиска.

Асимптотическая оценка сложности алгоритма. Алгоритмы сортировки и поиска. Бинарный поиск. Простые методы сортировки: обменные сортировки, сортировка выбором, сортировка вставками. Сортировка Шелла. Быстрая сортировка. Сортировка слиянием.

## **7. Паттерны проектирования**

Основные принципы объектно-ориентированного проектирования приложений. Принцип абстракции. Уменьшение зависимости. Основные паттерны проектирования: интерфейс, делегирование. Порождающие шаблоны: фабричный метод, абстрактная фабрика, строитель. Структурные шаблоны: адаптер, мост, декоратор, фасад. Поведенческие шаблоны: цепочка обязанностей, команда, посредник, наблюдатель, состояние, стратегия, посетитель, шаблонный метод.

## **8. Программирование графических интерфейсов**

Событийно-ориентированное программирование. События и обработчики событий. События мыши и клавиатуры. Основные библиотеки графических интерфейсов в Python: tkinter, PyQt, PyGTK. Главный цикл программы. Основные элементы управления: кнопки, ползунки, поля ввода. Работа с графикой.

## **9. Системное программирование на Python**

Чтение и запись файлов. Работа с путями в Windows и Linux. Обход папок. Работа с архивами. Работа с офисными форматами. Работа со структурированными данными. Основные форматы хранения данных: CSV, XML, JSON.

## **10. Сетевое программирование на Python**

Получение и разбор HTML-страниц. Библиотеки HTML-парсинга. Сокеты. Клиент-серверные приложения. Обращение к внешним API. Многопоточность. Библиотеки многопоточности и многопроцессности. Отправка и получение электронных писем.

## **11. Тестирование программ на Python**

Основные виды тестирования программного обеспечения. Модульное и интеграционное тестирование. Понятие регрессии. Фиксирование и

формализация требований. Библиотеки автоматизированного тестирования. Написание автоматических модульных тестов по техническому заданию.

Разработка через тестирование. Проектирование через тестирование. Методология TDD.

## 12. Документирование и развертывание программ на Python

Основные приемы документирования программного кода. Соглашения о стиле документирования кода. Написание программной документации в формате docstring. Языки форматирования документации: ReST, Markdown. Автоматическая сборка документации с использованием Sphinx. Экспорт документации в популярные форматы.

### 5.2. Учебно-тематический план

*Очная, очно-заочная / заочная форма обучения*

№ п/п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах						Формы текущего контроля успеваемости
		Всего	Аудиторная работа				Самосто ятельна я работа	
			Обща я, в т.ч.:	Ле кц ии	Семин ары, практи ческие заняти я	Занят ия в интер актив ной форм е		
1.	Основы языка Python	42/ 42	14/2	-	14/2	7/1	28/40	Опрос, проверка выполненных заданий
2.	Функции и модули	12/ 12	4/2	-	4/2	2/1	8/10	
3.	Обработка исключений. Работа с файлами средствами языка Python	12/ 12	4/2	-	4/2	2/1	8/10	
4.	Объектно- ориентированное программирование на Python	18/ 18	6/2	-	6/2	3/1	12/16	
5	Функциональное программирование на Python	12/ 12	4/2	-	4/2	2/1	8/10	
6	Алгоритмы и структуры данных	12/ 12	4/2	-	4/2	2/1	8/10	
7	Паттерны проектирования	18/ 18	6/2	-	6/2	3/1	12/16	
8	Программирование графических интерфейсов	18/ 18	6/2	-	6/2	3/1	12/16	

9	Системное программирование на Python	18/ 18	6/2	-	6/2	3/1	12/16	
10	Сетевое программирование на Python	18/ 18	6/2	-	6/2	3/1	12/16	
11	Тестирование программ на Python	18/ 18	6/2	-	6/2	3/1	12/16	
12	Документирование и развертывание программ на Python	18/ 18	6/2	-	6/2	3/1	12/16	
	В целом по дисциплине	216	72/ 24	-	72/24	36/12	144/ 192	Контрольные работы
	Итого в %					50%		

### 5.3. Содержание семинаров, практических занятий

#### *Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения)*

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Семестр 1		
Основы языка Python	Решение задач с использованием инструкции ветвления. Решение задач с использованием циклов. Создание и обработка списков. Многомерные списки. Обработка текстовой информации. Использование словарей и множеств. Совместное использование различных типов данных. Основная литература - [8.1], [8.2] Дополнительная литература - [8.3]	Интерактивная форма, работа на компьютере
Функции и модули	Создание и использование функций. Анонимные функции. Функции высшего порядка. Основная литература - [8.1], [8.2] Дополнительная литература - [8.3]	Интерактивная форма, работа на компьютере
Семестр 2		
Обработка исключений. Работа с файлами средствами языка Python	Работа с текстовыми файлами. Работа с двоичными файлами. Сохранение объектов в файл. Дополнительная литература - [8.3], [8.11]	Интерактивная форма, работа на компьютере
Объектно-ориентированное программирование на Python	Создание классов и объектов. Наследование и полиморфизм. Специальные методы классов. Основная литература - [8.1], [8.2] Дополнительная литература - [8.3]	Интерактивная форма, работа на компьютере

Функциональное программирование на языке Python	Функции map, filter, reduce, any, all. Декораторы. Итераторы. Функции генераторы. Основная литература - [8.2]	Интерактивная форма, работа на компьютере
Алгоритмы и структуры данных	Создание связанных списков. Использование стеков и очередей при решении задач. Алгоритмы сортировки и поиска. Дополнительная литература - [8.4]	Интерактивная форма, работа на компьютере
Семестр 3		
Паттерны проектирования	Создание примеров программных реализаций для распространенных шаблонов проектирования. Дополнительная литература - [8.5], [8.6], [8.7], [8.8]	Интерактивная форма, работа на компьютере
Программирование графических интерфейсов	Создание графического приложения с использованием встроенных возможностей языка и библиотек Дополнительная литература - [8.9], [8.10]	Интерактивная форма, работа на компьютере
Системное программирование на Python	Автоматизация рутинных административных задач. Создание Web-парсера. Дополнительная литература - [8.11]	Интерактивная форма, работа на компьютере
Семестр 4		
Сетевое программирование на Python	Создание многопоточного многопользовательского многофункционального сервера на сокетах. Дополнительная литература - [8.11]	Интерактивная форма, работа на компьютере
Тестирование программ на Python	Написание модульных тестов по описанию. Разработка программы по тестам. Дополнительная литература - [8.12], [8.13]	Интерактивная форма, работа на компьютере
Документирование и развертывание программ на Python	Документирование программы по описанию и исходному коду. Экспорт документации в формате сайта. Дополнительная литература - [8.14] Ресурсы «Интернет» - [9.3], [9.5]	Интерактивная форма, работа на компьютере

***(заочная форма обучения)***

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Семестр 3		
Основы языка Python	Создание и обработка списков. Обработка текстовой информации. Использование словарей. Основная литература - [1], [2] Дополнительная литература - [3]	Интерактивная форма, работа на компьютере
Функции и модули	Создание и использование функций. Анонимные функции. Функции высшего порядка. Основная литература - [1], [2] Дополнительная литература - [3]	Интерактивная форма, работа на компьютере

Обработка исключений. Работа с файлами средствами языка Python	Работа с текстовыми файлами. Сохранение объектов в файл. Дополнительная литература - [3], [11]	Интерактивная форма, работа на компьютере
Объектно-ориентированное программирование на Python	Создание классов и объектов. Наследование и полиморфизм. Специальные методы классов. Основная литература - [1], [2] Дополнительная литература - [3]	Интерактивная форма, работа на компьютере
Функциональное программирование на языке Python	Функции map, filter, reduce. Декораторы. Основная литература - [2]	Интерактивная форма, работа на компьютере
Алгоритмы и структуры данных	Создание связанных списков. Использование стеков и очередей при решении задач. Дополнительная литература - [4]	Интерактивная форма, работа на компьютере
Семестр 4		
Паттерны проектирования	Создание примеров программных реализаций для распространенных шаблонов проектирования. Дополнительная литература - [5], [6], [7], [8]	Интерактивная форма, работа на компьютере
Программирование графических интерфейсов	Создание графического приложения с использованием встроенных возможностей языка и библиотек Дополнительная литература - [9], [10]	Интерактивная форма, работа на компьютере
Системное программирование на Python	Автоматизация рутинных административных задач. Создание Web-парсера. Дополнительная литература - [11]	Интерактивная форма, работа на компьютере
Сетевое программирование на Python	Создание многопоточного многопользовательского многофункционального сервера на сокетах. Дополнительная литература - [11]	Интерактивная форма, работа на компьютере
Тестирование программ на Python	Написание модульных тестов по описанию. Разработка программы по тестам. Дополнительная литература - [12], [13]	Интерактивная форма, работа на компьютере
Документирование и развертывание программ на Python	Документирование программы по описанию и исходному коду. Экспорт документации в формате сайта. Дополнительная литература - [14] Ресурсы «Интернет» - [3], [5]	Интерактивная форма, работа на компьютере

## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

### 6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

*Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения)*

<b>Наименование тем (разделов) дисциплины</b>	<b>Перечень вопросов, отводимых на самостоятельное освоение</b>	<b>Формы внеаудиторной самостоятельной работы</b>
Основы языка Python	Функции модуля math, random, сору. Регулярные выражения. Кортежи. Множества.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Функции и модули	Создание и использование модулей и пакетов.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Обработка исключений. Работа с файлами средствами языка Python	Понятие исключения. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Инструкции raise и assert. Инструкция with ... as. Работа с файлами различных форматов: csv, docx, xlsx.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Объектно-ориентированное программирование на Python	Специальные методы классов.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Функциональное программирование на языке Python	Функции any, all, zip.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Алгоритмы и структуры данных	Бинарные деревья. Использование бинарных деревьев в прикладных задачах. Двоичное дерево поиска.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Паттерны проектирования	Внедрение зависимости, инверсия управления	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Программирование графических интерфейсов	Создание игр. Библиотека PyGame.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Системное программирование на Python	Создание административных скриптов	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Сетевое	Создание веб-сервиса	Работа с учебной

программирование на Python		литературой и документацией. Решение задач. Выполнение домашних заданий.
Тестирование программ на Python	Экстремальное программирование	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Документирование и развертывание программ на Python	Непрерывная интеграция	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.

***(заочная форма обучения)***

<b>Наименование тем (разделов) дисциплины</b>	<b>Перечень вопросов, отводимых на самостоятельное освоение</b>	<b>Формы внеаудиторной самостоятельной работы</b>
Основы языка Python	Инструкции ветвления и цикла. Функции модуля math, random, сору. Регулярные выражения. Многомерные списки. Кортежи. Множества.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Функции и модули	Функции с необязательными параметрами и переменным числом параметров. Создание и использование модулей и пакетов.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Обработка исключений. Работа с файлами средствами языка Python	Понятие исключения. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Инструкции raise и assert. Инструкция with ... as. Двоичные файлы. Работа с файлами различных форматов: csv, docx, xlsx.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Объектно-ориентированное программирование на Python	Специальные методы классов.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Функциональное программирование на языке Python	Функции any, all, zip. Итераторы. Функции генераторы.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Алгоритмы и структуры данных	Бинарные деревья. Использование бинарных	Работа с учебной литературой и

	деревьев в прикладных задачах. Двоичное дерево поиска. Алгоритмы сортировки и поиска.	документацией. Решение задач. Выполнение домашних заданий.
Паттерны проектирования	Внедрение зависимости, инверсия управления	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Программирование графических интерфейсов	Создание игр. Библиотека PyGame.	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Системное программирование на Python	Создание административных скриптов	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Сетевое программирование на Python	Создание веб-сервиса	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Тестирование программ на Python	Экстремальное программирование	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.
Документирование и развертывание программ на Python	Непрерывная интеграция	Работа с учебной литературой и документацией. Решение задач. Выполнение домашних заданий.

## 6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

### *Примерный перечень вопросов к контрольной работе № 1*

1. Создание классов.
2. Специальные методы класса.
3. Реализация связанных списков.
4. Понятие стека и очереди.
5. Работа с текстовым файлом.
6. Создание и обработка исключений.

### *Пример контрольной работы №1*



Создайте класс Queue, реализующий работу с очередью. Для хранения элементов очереди используйте связный список. Количество элементов в очереди не ограничено, но можно включить режим, когда при достижении заданного количества элементов в очереди возбуждается исключение. В классе должны быть:

- конструктор с параметрами. Параметр n определяет количество элементов в очереди, при достижении которого возбуждается исключение (по умолчанию n = 10). Параметр monitor определяет, нужно ли контролировать количество элементов в очереди. По умолчанию monitor = False (контроля нет). Если monitor = True, то контроль выполняется.
- метод push(self, value) – добавляет элемент в очередь. Если количество элементов очереди становится кратным n, то после добавления элемента возбуждается исключение;
- метод pop(self) – удаляет элемент из очереди и возвращает его значение. Если очередь пуста, то возвращает None;
- top() – возвращает элемент, находящийся в начале очереди, элемент из очереди не удаляется. Если очередь пуста, то возвращает None;
- size() – возвращает количество элементов в очереди;
- isempty() – возвращает True, если очередь не содержит элементов;
- empty() – удаляет все элементы из очереди;
- методы \_\_next\_\_(self) и \_\_iter\_\_(self);
- метод \_\_str\_\_(self).

Используя созданный класс решите следующую задачу. Имеется текстовый файл data.txt, содержащий целые числа. В каждой строке файла может находиться любое количество чисел, в том числе строка может быть пустой. Числа разделяются запятыми. Перед числом и после него могут быть пробелы. Создайте новый файл, содержащий эти же числа, в котором в каждой строке содержится ровно 5 чисел, которые разделены запятой и пробелом. В последней строке может быть меньше 5 чисел. Файл не содержит пустых строк и лишних пробелов.

Форма сдачи работы: файл с текстом программы. Имя файла (для студента по фамилии Иванов) – Иванов.py. Файл с данными должен иметь имя, указанное в задании. Файл с текстом программы должен содержать необходимые комментарии.

### ***Примерный перечень вопросов к контрольной работе №2***

1. Паттерны проектирования
2. Сетевое программирование. Библиотека socket
3. Параллельное программирование. async.
4. Параллельное программирование. threading.
5. Параллельное программирование. multiprocessing.

### ***Пример контрольной работы № 2***

Реализовать сканер TCP-портов. Программа должна запрашивать имя хоста/IP-адрес у пользователя. Затем программа должна пробовать подключиться к этому хосту ко всем портам по очереди. При успешном подключении программа должна выводить в консоль сообщение “Порт N открыт”.

Модифицировать эту программу, чтобы сканирование портов происходило параллельно. Для этого нужно распараллелить сканирование портов по нескольким потокам.

Обеспечить вывод списка открытых портов по порядку.

Реализовать progress bar в командной строке, показывающий прогресс сканирования.

### **Критерии балльной оценки различных форм текущего контроля успеваемости**

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Департамента анализа данных, принятия решений и финансовых технологий.

### **7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2 «Перечень планируемых результатов освоения образовательной программы с

указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

**Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний**

Наименование компетенции	Примеры типовых заданий для оценки индикаторов достижения компетенций
<p><b>ОПК-2</b> Способность использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности</p>	<p><b>1. Демонстрирует знания об основных информационных технологиях и программных средствах, позволяющих их использовать</b></p> <p align="center"><b>Задание 1</b></p> <p>Для треугольника со сторонами <math>x</math>, <math>y</math>, <math>z</math> угол <math>\alpha</math> между сторонами <math>x</math>, <math>y</math> можно вычислить следующим образом:  <math display="block">d = \cos \alpha = \frac{x^2 + y^2 - z^2}{2xy}; \alpha = \arccos d = \frac{\pi}{2} - \arcsin d</math></p> <p>Напишите функцию, которая находит угол <math>\alpha</math> для треугольника со сторонами <math>x</math>, <math>y</math>, <math>z</math> в градусах. Используя эту функцию напишите еще одну функцию, которая по заданным сторонам треугольника находит все его углы (в градусах). Функция возвращает кортеж из 3 чисел, причем первое число – угол, находящийся напротив стороны <math>x</math>, второе – угол напротив <math>y</math>, третье – угол напротив <math>z</math>.</p> <p align="center"><b>Задание 2</b></p> <p>Создайте класс Студент, имеющий:</p> <ul style="list-style-type: none"> <li>• закрытый атрибут Имя – строка, содержащая фамилию;</li> <li>• метод <code>__init__</code>. При создании объекта указывается имя, список Дисциплины пустой;</li> <li>• закрытый атрибут Дисциплины – словарь сданных дисциплин. Ключом является название дисциплины, значением – оценка.</li> <li>• метод <code>put</code> добавляет новую дисциплину в атрибут Дисциплины. Параметрами метода являются название дисциплины и оценка;</li> <li>• свойство <code>Сдано</code> возвращает список названий сданных дисциплин;</li> </ul> <p>Создайте экземпляр класса, продемонстрируйте работу с атрибутами, методами и свойствами.</p> <p align="center"><b>Задание 3</b></p> <p>Создайте в текстовом редакторе (Блокнот) текстовый файл, содержащий информацию о товарах и ценах на них. Каждая строка файла имеет вид: НАЗВАНИЕ ТОВАРА: ЦЕНА. Используя данный файл средствами языка Python:</p> <ul style="list-style-type: none"> <li>• найдите цену указанного товара, или выдайте сообщение о том, что цена не известна;</li> <li>• добавьте в файл информацию о трех новых товарах;</li> <li>• удалите из файла информацию о товаре;</li> <li>• создайте новый файл, в котором товары будут упорядочены в порядке возрастания цен. Выведите на экран информацию о двух самых дешевых и двух</li> </ul>

	<p>самых дорогих товарах.</p> <p><b>2. Рационально выбирает информационные технологии и реализующие их программные средства, в том числе, с учетом страны происхождения программных средств</b></p> <p style="text-align: center;"><b>Задание 1</b></p> <p>Информация о продажах вводится в виде строки «Покупатель Товар Количество», где Покупатель — имя покупателя (строка без пробелов), Товар — название товара (строка без пробелов), Количество — количество приобретенных единиц товара (целое число). Конец ввода – пустая строка.</p> <p>Задание:</p> <ul style="list-style-type: none"> <li>• выведите на экран списки покупателей и товаров;</li> <li>• для каждого покупателя подсчитайте количество приобретенных им единиц каждого вида товаров;</li> <li>• для каждого товара получите информацию о том, кто и сколько его купил.</li> </ul> <p>Результат выполнения программы может выглядеть примерно так:</p> <pre>=&gt;a t1 1 =&gt;b t2 2 =&gt;b t1 10 =&gt;a t1 3 =&gt; Товары: t2,t1 Покупатели: b,a Товары покупателей {'b': {'t2': 2, 't1': 10}, 'a': {'t1': 4}} Покупатели товара {'t2': {'b': 2}, 't1': {'b': 10, 'a': 4}} &gt;&gt;&gt;</pre> <p style="text-align: center;"><b>Задание 2</b></p> <p>Имеется список слов. Используя 2 очереди, преобразовать исходный список в список, в котором сначала идут все слова, содержащие только буквы, а потом – слова, в которых есть буквы и цифры. Относительный порядок слов должен сохраниться.</p> <p style="text-align: center;"><b>Задание 3</b></p> <p>Участники олимпиады решали 3 задачи. Известны фамилии тех, кто решил первую, вторую и третью задачи (для каждой задачи отдельный список). Найдите и выведите на экран фамилии тех, кто</p> <ul style="list-style-type: none"> <li>• решил хотя бы одну задачу (любую);</li> <li>• решил все задачи;</li> <li>• решил ровно 1 задачу (любую);</li> <li>• решил ровно 2 задачи (любые);</li> <li>• решил не больше 2 задач (любых).</li> </ul> <p><b>3. Использует современные информационные технологии и программные средства при решении задач разработки программного обеспечения для экономических и финансовых приложений</b></p> <p style="text-align: center;"><b>Задание 1 (Абстрактная фабрика)</b></p> <p>Реализуйте интерфейс для создания семейств похожих или зависимых объектов без указания их конкретного класса.</p>
--	--

	<p>С точки зрения системы: Вам нужно построить каркас приложения для покупок, новые не хотите создавать конкретные классы продуктовых моделей и способов доставки, так как хотите, чтобы пользователи сами могли создавать фабрики моделей и методов и сами классы конкретных моделей и методов</p> <p>С точки зрения пользователя: Вы хотите иметь возможность создавать собственные классы для продуктовых моделей и способов доставки без необходимости модифицировать базовую модель, предоставленную каркасом приложения. В идеале, вы сможете создать собственную объектную модель продукта и просто ввести механизм взаимодействия с этим продуктом в каркас торгового приложения.</p> <p style="text-align: center;"><b>Задание 2 (Адаптер)</b></p> <p>Напишите приложение, способное принимать платежи посредством систем PayPal и через кредитные карты при условии, что программные интерфейсы (API) этих систем значительно различаются</p> <p style="text-align: center;"><b>Задание 3 (Декоратор)</b></p> <p>Напишите расширение к классу элемента ввода пользовательского текста, добавляющее метку и возможность обработки ошибочного ввода.</p>
<p><b>ОПК-3</b> Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p>	<p><b>1. Использует информационно-коммуникационные технологии и библиографические источники при поиске информации, для решения стандартных задач.</b></p> <p style="text-align: center;"><b>Задание 1 (Внедрение зависимости)</b></p> <p>Вам даны множество объектов, которым необходимо обращаться к базе данных. Вместо того, чтобы открывать множество соединений, являющихся объектами, напишите приложение, управляющее запросами через единый объект подключения</p> <p style="text-align: center;"><b>Задание 2 (фабричный метод)</b></p> <p>Напишите приложение, реализующее следующую функциональность: С точки зрения системы: Вам нужно построить каркас приложения для покупок, новые не хотите создавать конкретные классы продуктовых моделей и способов доставки, так как хотите, чтобы пользователи сами могли создавать фабрики моделей и методов и сами классы конкретных моделей и методов</p> <p>С точки зрения пользователя: Вы хотите иметь возможность создавать собственные классы для продуктовых моделей и способов доставки без необходимости модифицировать базовую модель, предоставленную каркасом приложения. В идеале, вы сможете создать собственную объектную модель продукта и просто ввести механизм взаимодействия с этим продуктом в каркас торгового приложения.</p> <p style="text-align: center;"><b>Задание 3 (Итератор)</b></p> <p>Напишите программу, подсчитывающую последовательность Фибоначчи. Алгоритм прохода должен быть разделен с</p>

	<p>алгоритмом подсчета самого элемента и доступа к нему.</p> <p><b>2. Демонстрирует умение решать стандартные задачи разработки информационных систем.</b></p> <p><b>Задание 1 (Преобразователь)</b></p> <p>Напишите класс, реализующий следующую функциональность: данные из базы данных (в форме ассоциативного массива) должны быть преобразованы в объектную модель. В будущем эти объекты понадобятся преобразовать обратно к ассоциативному массиву для записи в базу данных.</p> <p><b>Задание 2 (Наблюдатель)</b></p> <p>Модель данных может быть преобразована из произвольных участков кода приложения. Необходимо реализовать логгер, который фиксирует в специальном файле факт изменения модели данных.</p> <p><b>Задание 3 (Прототип)</b></p> <p>Напишите приложение, делегирующее создание частных объектных типов пользователю. Вместо использования фабрики и с учетом того, что объектов может быть большое количество, необходимо реализовать зависимости.</p>
	<p><b>3. Владеет навыками обеспечения информационной безопасности автоматизированных систем</b></p> <p><b>Задание 1 (Посредник)</b></p> <p>Вам дан класс изображения, который загружает файл в своем конструкторе. Необходимо создать класс-посредник, который загружает файл только в случае необходимости, например, при отображении на экране.</p> <p><b>Задание 2 (Стратегия)</b></p> <p>Вам дано множество объектов с бизнес-логикой, которые представляют позиции в торговой системе. Необходимо реализовать возможность пользователю отфильтровать результаты в окне поиска по различным произвольным критериям.</p> <p><b>Задание 3 (Посетитель)</b></p> <p>Необходимо реализовать структуру данных, управляющую данными, поступившими по методам GET или POST, причем поступить может как одна, так и множество порций данных за раз. Необходимо реализовать различные операторы на этих структурах данных</p>

***Теоретические вопросы для подготовки к зачету***  
***Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения 1 семестр)***

1. Функции модуля math.
2. Инструкция if...else.
3. Инструкция цикла while.
4. Инструкция цикла for.

5. Функция range.
6. Строки. Операции над строками (+, \*, in, доступ по индексу [], получение среза). Функция len.
7. Методы строк: strip, find, count, replace.
8. Списки в Python. Создание: создание пустого списка, методы append, split, функция list. Генераторы списков.
9. Создание копии списка (срезы, функции list и deepcopy).
10. Основные операции над списками (+, \*, in, доступ по индексу [], получение среза). Перебор элементов списка.
11. Добавление и удаление элементов списка (методы append, insert, pop, remove). Методы reverse, join. Функция map.
12. Сортировка списков. Параметры метода sort: key, reverse.
13. Кортежи. Создание. Создание пустого кортежа и кортежа из одного элемента. Операции (+, \*, in, доступ по индексу [], получение среза). Функции tuple, len.
14. Распаковка последовательности.
15. Словари. Создание словаря. Функции dict, zip.
16. Операции над словарями ([], in, del). Функция len.
17. Перебор элементов словаря. Методы get, clear, copy, keys, values.
18. Множества. Создание. Функции set, len.
19. Операции над множествами: in, |, &, -, ^, <=, >=, <, >, ==. Методы add, discard.
20. Создание и вызов функции.
21. Передача аргументов в функцию. Необязательные параметры функций. Функции в качестве аргументов.
22. Глобальные и локальные переменные.
23. Анонимные функции.
24. Создание и использование модулей. Инструкции import и from.
25. Пакеты. Использование пакетов.

### ***Теоретические вопросы для подготовки к зачету***

#### ***Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения 2 семестр)***

1. Инструкция try ... except ... else ... finally.
2. Инструкции raise и assert.
3. Инструкция with ... as.
4. Классы встроенных исключений.

5. Работа с текстовыми файлами. Открытие файла (функция open) и закрытие файла (метод close). Чтение текстового файла (методы read, readline, readlines). Перебор строк файла в цикле for. Запись в текстовый файл (метод write, функция print с параметром file).
6. Сохранение объектов в файл (функции dump и load модуля pickle).
7. Понятие класса и объекта. Определение класса и создание экземпляра класса.
8. Методы класса. Параметр self. Статические методы. Закрытые методы.
9. Атрибуты класса и экземпляра класса. Доступ к атрибуту. Закрытые атрибуты.
10. Свойства. Создание и использование свойства.
11. Наследование. Базовый и производный классы. Переопределение методов.
12. Специальные методы. Перегрузка операторов.
13. Функции map, filter, reduce, any, all.
14. Декораторы функций.
15. Итераторы.
16. Функции-генераторы.
17. Массивы. Использование массивов.
18. Стеки. Использование стеков. Реализация стека.
19. Очереди. Использование очереди. Реализация очереди.
20. Связные списки. Использование связанных списков. Реализация связанных списков.
21. Бинарные деревья. Использование бинарных деревьев. Реализация бинарных деревьев.
22. Бинарный поиск.
23. Обменные сортировки.
24. Сортировка Шелла.
25. Быстрая сортировка.

### ***Теоретические вопросы для подготовки к зачету***

#### ***Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения 3 семестр)***

1. Основные принципы объектно-ориентированного проектирования.
2. Шаблон проектирования: интерфейс
3. Шаблон проектирования: делегирование.
4. Шаблон проектирования: фабричный метод



5. Шаблон проектирования: абстрактная фабрика
6. Шаблон проектирования: строитель
7. Шаблон проектирования: адаптер
8. Шаблон проектирования: мост
9. Шаблон проектирования: декоратор
10. Шаблон проектирования: фасад
11. Шаблон проектирования: цепочка обязанностей
12. Шаблон проектирования: команда
13. Шаблон проектирования: посредник
14. Шаблон проектирования: наблюдатель
15. Шаблон проектирования: состояние
16. Шаблон проектирования: стратегия
17. Шаблон проектирования: посетитель
18. Шаблон проектирования: шаблонный метод.
19. Событийно-ориентированное программирование.
20. События и обработчики событий. Виды событий.
21. События мыши и клавиатуры.
22. tkinter, PyQt, PyGTK – особенности и отличия.
23. Главный цикл программы.
24. Основные элементы управления: кнопки, ползунки, поля ввода.
25. Работа с графикой.
26. Работа с путями в Windows и Linux.
27. Основные форматы хранения данных: CSV, XML, JSON.

### ***Теоретические вопросы для подготовки к зачету***

#### ***Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах» (очная, очно-заочная форма обучения 4 семестр)***

1. Получение и разбор HTML-страниц.
2. Библиотеки HTML-парсинга.
3. Сокеты.
4. Клиент-серверные приложения.
5. Обращение к внешним API.
6. Многопоточность.
7. Библиотеки многопоточности и многопроцессности.
8. Отправка и получение электронных писем.
9. Основные виды тестирования программного обеспечения.
10. Модульное и интеграционное тестирование.
11. Понятие регрессии.

12. Библиотеки автоматизированного тестирования.
13. Разработка через тестирование.
14. Методология TDD.
15. Основные приемы документирования программного кода.
16. Соглашения о стиле документирования кода.
17. Языки форматирования документации: ReST, Markdown.
18. Автоматическая сборка документации с использованием Sphinx.

***Теоретические вопросы для подготовки к зачету***  
***Профиль «Высокопроизводительные вычисления в цифровой экономике»***  
***(заочная форма обучения 3 семестр)***

1. Инструкция if...else.
2. Инструкции цикла while и for. Функция range.
3. Строки. Операции над строками (+, \*, in, доступ по индексу [], получение среза). Функция len. Методы строк: strip, find, count, replace.
4. . Создание списка в Python: создание пустого списка, методы append, split, функция list. Генераторы списков. Создание копии списка.
5. Основные операции над списками (+, \*, in, доступ по индексу [], получение среза). Перебор элементов списка.
6. Методы списков: index, insert, pop, remove, reverse, join, sort, count.
7. Создание словаря. Функция dict. Генераторы словарей. Создание копии словаря.
8. Операции над словарями ([], in, del). Функция len.
9. Перебор элементов словаря. Методы get, clear, copy, keys, values.
10. Множества. Создание. Функции set, len.
11. Операции над множествами: in, |, &, -, ^, <=, >=, <, >, ==. Методы add, discard.
12. Создание и вызов функции.
13. Передача аргументов в функцию. Необязательные параметры функций. Функции в качестве аргументов. Глобальные и локальные переменные.
14. Создание и использование анонимных функций.
15. Создание и использование модулей. Инструкции import и from.
16. Понятие исключения. Инструкция try ... except ... else ... finally. Инструкции raise и assert. Классы встроенных исключений.
17. Работа с текстовыми файлами. Открытие файла (функция open) и закрытие файла (метод close). Чтение текстового файла (методы read,

- readline, readlines). Перебор строк файла в цикле for. Запись в текстовый файл (метод write, функция print с параметром file).
18. Сохранение объектов в файл (функции dump и load модуля pickle).
  19. Создание и использование методов класса. Параметр self. Статические методы. Защищенные методы. Специальные методы.
  20. Атрибуты класса и экземпляра класса. Доступ к атрибуту. Защищенные атрибуты.
  21. Свойства. Создание и использование свойства.
  22. Наследование. Базовый и производный классы. Переопределение методов.
  23. Функции map, filter, reduce, any, all.
  24. Декораторы функций.
  25. Итераторы.
  26. Функции-генераторы.
  27. Стек. Использование стека. Реализация стека.
  28. Очереди. Использование очереди. Реализация очереди.
  29. Связные списки. Использование связанных списков. Реализация связанных списков.
  30. Основные алгоритмы сортировки.

### **Теоретические вопросы для подготовки к зачету**

#### ***Профиль «Высокопроизводительные вычисления в цифровой экономике» (заочная форма обучения 4 семестр)***

1. Основные принципы объектно-ориентированного проектирования.
2. Событийно-ориентированное программирование.
3. События и обработчики событий. Виды событий.
4. События мыши и клавиатуры.
5. tkinter, PyQt, PyGTK – особенности и отличия.
6. Главный цикл программы.
7. Основные элементы управления: кнопки, ползунки, поля ввода.
8. Основные форматы хранения данных: CSV, XML, JSON.
9. Получение и разбор HTML-страниц.
10. Библиотеки HTML-парсинга.
11. Сокеты.
12. Библиотеки многопоточности и многопроцессности.
13. Отправка и получение электронных писем.
14. Основные виды тестирования программного обеспечения.

15. Библиотеки автоматизированного тестирования.
16. Разработка через тестирование.
17. Основные приемы документирования программного кода.
18. Соглашения о стиле документирования кода.
19. Языки форматирования документации: ReST, Markdown.

### **Примеры заданий для подготовки к зачету**

#### ***1 семестр (очная и очно-заочная форма обучения), 3 семестр (заочная форма обучения)***

1. Чему будет равно значение переменной A после выполнения оператора `A = [i+10 for i in range(5,0,-1)]`
2. Имеется строка `S="1234567890"`. Чему равно значение S1, если `S1 = S[1:2] + S[4: :-1]`
3. Используя генератор словарей, для заданного натурального числа n создайте словарь D, в котором будет ровно n элементов. Элементами словаря являются `{'s1':10, 's2': 20, 's3': 30, ...}`, т.е. значение равно номеру элемента (нумерация с 1), умноженному на 10, а ключ состоит из символа 's', к которому дописан номер.

#### ***2 семестр (очная и очно-заочная форма обучения), 3 семестр (заочная форма обучения)***

1. Имеется список L вида `[[1, 3, 65], [11, 5, 6], [15, 33, 11]]`. Отсортируйте список по возрастанию последнего значения элемента.
2. После выполнения приведенного кода будет напечатано ...

```
def data(d=[]):  
    d.append(1)  
    return d
```

```
a=data()  
b=data()  
c=data([3])  
print(b,c)
```

- a) `[1] [3]`
- b) `[2] [4]`
- c) `[1,1] [3,1]`
- d) `[1,2] [3,4]`

е) Будет выведено сообщение об ошибке

3. После выполнения приведенного кода будет напечатано ...

```
class Class1:
    def __init__(self, n):
        self._x = n

    def gx(self):
        return self._x+2

x = property(gx)

c=Class1(2)
print(c.x)
```

- a) 2
- b) 4
- c) None
- d) Возникнет ошибка

4. Временная сложность алгоритма – это

- a) время, необходимое для выполнения алгоритма
- b) время, затрачиваемое алгоритмом на его выполнение как функция размера задачи
- c) количество операций, которое выполняет алгоритм за указанное время
- d) время, затраченное на разработку алгоритма

***3 семестр (очная и очно-заочная форма обучения), 4 семестр (все формы обучения)***

1. Приведите пример приватного IP-адреса

- a. 196.168.0.1
- b. 127.0.0.10
- c. 87.250.250.242
- d. 173.194.73.113

2. Приведите пример публичного IP-адреса

- a. 8.8.8.8
- b. 127.0.0.1

- c. 196.168.0.1
- d. 10.38.51.16

3. Соглашение о порядке и способе связи между компьютерами это:
- a. сетевой протокол
  - b. сетевой интерфейс
  - c. коммутация
  - d. адресация
4. Как называется самый распространенный стандарт на архитектуру локальных сетей на основе кабельного соединения
- a. Ethernet
  - b. Wi-Fi
  - c. TCP
  - d. GlobalNet X
5. Приведите пример стандарта из серии IEEE 802
- a. Wi-Fi
  - b. витая пара
  - c. POSIX
  - d. USB
6. Назовите команду, отображающую основную информацию о сетевых подключениях в ОС Linux
- a. ipconfig
  - b. ifconfig
  - c. netstat
  - d. traceroute
7. Назовите команду, отображающую основную информацию о сетевых подключениях в ОС Windows
- a. ifconfig
  - b. ipconfig
  - c. netstat
  - d. tracert
8. Назовите пример сетевой топологии
- a. звезда
  - b. снежинка
  - c. лабиринт

d. круговая

9. Назовите пример стандартного сетевого кабеля

- a. витая пара
- b. медный
- c. экранированный
- d. беспроводной

10. Как называется программно-аппаратный компонент, осуществляющий соединение двух сетей, построенных на разной архитектуре

- a. шлюз
- b. коммутатор (свитч)
- c. концентратор (хаб)
- d. роутер

11. Как называется простейший сетевой компонент, используемый для соединения нескольких узлов сети в одной точке?

- a. концентратор (хаб)
- b. коммутатор (свитч)
- c. роутер
- d. шлюз

12. Как называется сетевой компонент, осуществляющий пересылку пакетов по назначению?

- a. коммутатор (свитч)
- b. концентратор (хаб)
- c. шлюз
- d. роутер

13. Как называется сетевой компонент, служащий для соединения разных подсетей?

- a. роутер
- b. коммутатор (свитч)
- c. концентратор (хаб)
- d. шлюз

14. Какие параметры сети нужно знать, чтобы подключить новый узел к ней (в случае статических IP-адресов)

- a. адрес, маска сети, адрес шлюза или роутера
- b. формат сетевого разъема, версия Ethernet
- c. версию операционной системы

- d. марку и длину сетевого кабеля, уровень помех в помещении.
15. Назовите пример стандартного сетевого протокола транспортного уровня
- a. TCP
  - b. HTTP
  - c. FTP
  - d. POP3
16. Сколько уровней в стандартной модели сетевых протоколов OSI?
- a. 7
  - b. 4
  - c. 10
  - d. 5
17. Назовите самый низкий уровень модели OSI?
- a. физический
  - b. канальный
  - c. транспортный
  - d. прикладной
18. Назовите самый высокий уровень модели OSI?
- a. прикладной
  - b. сетевой
  - c. сеансовый
  - d. транспортный
19. Сколько уровней в семействе протоколов TCP/IP?
- a. 4
  - b. 7
  - c. 10
  - d. 5
20. Чем определяется количество компьютеров в подсети?
- a. маской подсети
  - b. настройками шлюза
  - c. архитектурой сети
  - d. используемыми протоколами

**8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

**а) основная**



1. Жуков, Р. А. Язык программирования Python. Практикум [Электронный ресурс]: учеб. пособие / Р.А. Жуков. — Москва: ИНФРА-М, 2019. — 216 с. — Режим доступа: <http://znanium.com/catalog/product/1000002>.
2. Рамальо, Л. Python. К вершинам мастерства [Электронный ресурс] / Лучано Рамальо; пер. с англ. А.А. Слинкина. — Москва: ДМК Пресс, 2016. — 768 с. — Режим доступа: <http://znanium.com/catalog/product/1028052>

**б) дополнительная:**

3. Дроздов С.Н. Структуры и алгоритмы обработки данных [Электронный ресурс]: учебное пособие / С.Н. Дроздов; Министерство образования и науки РФ, Южный федеральный университет, Инженерно-технологическая академия. — Таганрог : Издательство Южного федерального университета, 2016. — 228 с. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=493032>,  
<http://znanium.com/go.php?id=991928>
4. Буйначев С.К. Основы программирования на языке Python [Электронный ресурс]: учебное пособие / С.К. Буйначев, Н.Ю. Боклаг ; Министерство образования и науки Российской Федерации, Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. — Екатеринбург: Издательство Уральского университета, 2014. — 92 с. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=275962>

**9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>

2. <https://www.python.org/downloads/>
3. <https://docs.python.org/3/>
4. <https://www.anaconda.com/>
5. <http://www.sphinx-doc.org/en/master/>
6. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>  
(<http://library.fa.ru/files/elibfa.pdf>)
7. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
8. Электронно-библиотечная система «Университетская библиотека  
ОНЛАЙН» <http://biblioclub.ru/>
9. Электронно-библиотечная система Znanium <http://www.znanium.com>
10. «Деловая онлайн библиотека» издательства «Альпина Паблишер»  
<http://lib.alpinadigital.ru/en/library>
11. Электронно-библиотечная система издательства «Лань»  
<https://e.lanbook.com/>
12. Электронно-библиотечная система издательства «ЮРАЙТ»  
<https://www.biblio-online.ru/>
13. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>

#### **10. Методические указания для обучающихся по освоению дисциплины**

Самостоятельная работа студентов проходит аудиторно и внеаудиторно. Организации самостоятельной работы служит учебно-тематический план изучения дисциплины. В этом плане указаны темы практических занятий, вопросы и задания для самостоятельного изучения.

При подготовке к практическому занятию необходимо повторить или, если это требуется, изучить соответствующий теоретический материал. Во время занятия нужно точно записывать формулировки решаемых задач, вопросы, указания преподавателя к решению и разбираемые решения. После занятий необходимо просмотреть записанные решения и восстановить в решениях имеющиеся пробелы. В случае затруднений отметить соответствующие задания и обратиться за консультацией к преподавателю. Практические занятия проходят, как правило, в интерактивной форме и преподаватель учитывает активность студентов в процессе решения

предложенных задач и поиска ответов на вопросы. Следует сохранять написанные программы, чтобы использовать их в дальнейшем в качестве образцов при решении других задач или как часть решения более общей задачи.

Домашние задания следует выполнять регулярно при подготовке к практическим занятиям. В большинстве своем задания являются типовыми, и образцы их решения содержатся в рекомендованных пособиях. Если то или иное задание вызвало затруднение, необходимо обратиться к преподавателю на консультации или ближайшем практическом занятии. Регулярность в выполнении домашних заданий — важный фактор освоения дисциплины. Даже небольшие отклонения от графика могут спровоцировать серьезное отставание и в дальнейшем — риск получения неудовлетворительных оценок в ходе текущей и промежуточной аттестации. Контроль выполнения домашних заданий осуществляется в ходе практических занятий в процессе собеседования.

Домашняя контрольная работа является одной из основных форм текущего контроля самостоятельной работы студентов по дисциплине. Сроки выполнения ДКР указываются в учебно-тематическом плане изучения дисциплины. Конкретные сроки сдачи ДКР устанавливаются преподавателем. Оценка за ДКР выставляется по итогам проверки работы и устного собеседования. Эта оценка является существенной компонентой оценки самостоятельной работы студента в течение семестра.

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем**

### **11.1 .Комплект лицензионного программного обеспечения:**

1. Windows, Microsoft Office,
2. Антивирус ESET Endpoint Security.

### **11.2. Современные профессиональные базы данных и информационные справочные системы**

1. Информационно-правовая система «Гарант»,
2. Информационно-правовая система «Консультант Плюс»,
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

### **11.3. Сертифицированные программные и аппаратные средства защиты информации: не предусмотрены.**

### **11.4. Язык программирования Python 3.x в среде Windows.**

11.5. Платформа для научных исследований, основанная на языке программирования Python, Anaconda.

**12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Компьютерный класс, оснащённый системой динамического проецирования.