



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)  
**ШКОЛА ЭКОНОМИКИ И МЕНЕДЖМЕНТА**

«СОГЛАСОВАНО»  
Руководитель ОП

\_\_\_\_\_ Е.А. Тюринa  
(подпись) (Ф.И.О. рук. ОП)

« 15 » \_\_\_\_\_ января 20 21 г.

«УТВЕРЖДАЮ»

Заведующий департамента управления  
на основе данных (Data Driven Management Department)  
(название кафедры)

\_\_\_\_\_ Ю. Д. Шмидт  
(подпись) (Ф.И.О. зав. каф.)

« 15 » \_\_\_\_\_ января 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
Питон (*Python*)  
Направление подготовки 38.04.01 Экономика  
Экономические исследования  
Форма подготовки: очная

курс 1 семестр 2  
лекции час.  
практические занятия 18 час.  
лабораторные работы 0 час.  
в том числе с использованием МАО лек. 0 /пр. 0 /лаб. 0 час.  
всего часов аудиторной нагрузки 18 час.  
в том числе с использованием МАО 0 час.  
самостоятельная работа 18 час.  
в том числе на подготовку к зачету час.  
контрольные работы (количество)  
курсовая работа / курсовой проект не предусмотрены  
зачет 2 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 38.04.01 Экономика, утвержденного приказом Министерства науки и высшего образования российской Федерации от 11.08.2020 № 939.

Рабочая программа обсуждена на заседании департамента управления на основе данных (Data Driven Management Department) ШЭМ ДВФУ № 4 от «15» января 2021 г.

Зав. департамента управления  
на основе данных (Data Driven Management Department),  
д-р экон. наук, проф.  
Составитель  
канд. экон. наук, доцент

Ю.Д. Шмидт

Н.А.Матев

Владивосток  
2021

**I. Рабочая программа пересмотрена на заседании департамента управления на основе данных (Data Driven Management Department)**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании департамента управления на основе данных (Data Driven Management Department):**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## Аннотация дисциплины

### *Питон (Python)*

Общая трудоемкость дисциплины составляет 1 зачётная единица / 36 академических часа. Является дисциплиной части, формируемой участниками образовательных отношений, изучается на 1 курсе и завершается зачетом. Учебным планом предусмотрено проведение практических занятий в объеме 18 часов, а также выделены часы на самостоятельную работу студента – 18 часов.

Язык реализации: русский.

#### **Цель:**

Формирование у студентов навыков программирования на *Python* с целью анализа данных различной природы.

#### **Задачи:**

- сформировать необходимый объем знаний о базовых концепциях программирования на *Python*;
- изучить основы программирования на языке *Python*, типы данных и конструкции языка;
- сформировать специфический понятийный аппарат;
- сформировать современное представление о важности статистических методов изучения в общественной жизни;
- сформировать навыки статистического анализа и оценки экономических процессов на макро и микроуровне посредством языка программирования *Python*.

Для успешного изучения дисциплины у обучающихся должны быть сформированы следующая предварительная компетенция: ПК-3, полученная в результате изучения дисциплин «Теория игр», «Статистика», обучающийся должен быть готов к изучению такой дисциплины, как «Анализ данных в R (продвинутый уровень)», формирующую компетенцию ПК-1.

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют формирование следующих компетенций, индикаторов достижения компетенций:

Наименование категории (группы) компетенций	Код и наименование компетенции (результат освоения)	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
Расчетно-экономические	ПК-1 – Способен вести работу по анализу результатов современных теоретических и эмпирических экономических исследований, поиску противоречий и разрывов, постановке исследовательских вопросов, определяющих перспективные направления научных исследований; разрабатывать и самостоятельно реализовывать программу научного исследования	ПК-1.1 – Формирует дизайн исследовательского проекта и корректно применяет количественные и качественные методы при его реализации	знает основные понятия языка программирования Python; методы описания структур данных на Python; классы задач, формулируемых и решаемых на Python; умеет разрабатывать программы на языке Python; применять изученные методы и структуры данных на практике.

Для формирования вышеуказанных компетенций в рамках дисциплины «Питон (Python)» применяются следующие дистанционные образовательные технологии и методы / активного / интерактивного обучения: работа в малых группах, решение ситуационных задач.

## I. Цели и задачи освоения дисциплины:

Цель: формирование у студентов навыков программирования на *Python* с целью анализа данных различной природы.

Задачи:

- сформировать необходимый объем знаний о базовых концепциях программирования на *Python*;
- изучить основы программирования на языке *Python*, типы данных и конструкции языка;
- сформировать специфический понятийный аппарат;
- сформировать современное представление о важности статистических методов изучения в общественной жизни;
- сформировать навыки статистического анализа и оценки экономических процессов на макро и микроуровне посредством языка программирования *Python*;

Место дисциплины в структуре ОПОП (учебном плане) (пререквизиты дисциплины, дисциплины, следующие после изучения данной дисциплины): для успешного изучения дисциплины у обучающихся должны быть сформированы следующие предварительные компетенции: ПК-3, полученная в результате изучения дисциплин «Теория игр», «Линейная алгебра», «Математический анализ», обучающийся должен быть готов к изучению такой дисциплины, как «Машинное обучение», формирующую компетенцию ПК-3.

Профессиональные компетенции студентов, индикаторы их достижения и результаты обучения по дисциплине

Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
--	--	--

<p>ПК-1 – Способен вести работу по анализу результатов современных теоретических и эмпирических экономических исследований, поиску противоречий и разрывов, постановке исследовательских вопросов, определяющих перспективные направления научных исследований; разрабатывать и самостоятельно реализовывать программу научного исследования</p>	<p>ПК-1.1 – Формирует дизайн исследовательского проекта и корректно применяет количественные и качественные методы при его реализации</p>	<p>знает основные понятия языка программирования Python; методы описания структур данных на Python; классы задач, формулируемых и решаемых на Python; умеет разрабатывать программы на языке Python; применять изученные методы и структуры данных на практике.</p>
--	---	---

## II. Трудоемкость дисциплины и виды учебных занятий по дисциплине

Общая трудоемкость дисциплины составляет 1 зачётная единица (36 академических часа).

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	С е м е с т р	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации***
			Лек	Лаб	Пр	ОК	СР	Конт роль	
1	Тема 1. Введение в <i>Python</i>	7			1				зачет
2	Тема 2. Ветвления и цикл с предусловием	7			2				
3	Тема 3. Циклы перебора	7			3				
4	Тема 4. Работа с текстом	7			3				
5	Тема 5. Регулярные структуры	7			3		18		
6	Тема 6. Модульный аспект	7			3				
7	Тема 7. Нерегулярные структуры	7			3				
Итого:					18		18		зачет

### III. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Теоретическая часть курса не предусмотрена учебным планом.

### IV. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

#### ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1. Введение в *Python*

История создания языка. *Python 2* и *Python 3*. Среды разработки. Области применимости. Ввод-вывод. Арифметические операции. Типы данных.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2. Ветвления и цикл с предусловием

Условная инструкция. Цикл *while*. Решение задач на ветвление и циклы.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3. Циклы перебора

Цикл *for*. Решение задач с вложенными циклами. Решение переборных задач.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4. Работа с текстом

Строки и символы. Некоторые методы у строк. Решение задач на обработку строк.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5. Регулярные структуры

Списки. Двумерные массивы. Решение задач на обработку списков.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6. Модульный аспект

Модули. Функции и рекурсия.

##### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 7. Нерегулярные структуры

Кортежи, словари и множества. Решение задач со словарями, кортежами, множествами.

### V. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Код и наименование индикатора достижения	Результаты обучения	Оценочные средства	
				текущий контроль	Промежуточная аттестация
1	Тема 1. Введение в <i>Python</i>	ПК-1.1	Знает основные среды разработки на языке <i>Python</i> , области	собеседование (УО-1)	–

			применимости языка		
			Умеет отличать типы данных и проводить над ними корректные арифметические действия	решение разноуровневых задач и заданий (ПР-11)	–
2	Тема 2. Ветвления и цикл с предусловием	ПК-1.1	Знает принцип работы условных конструкций и цикла <i>while</i> в <i>Python</i>	собеседование (УО-1)	–
			Умеет решать задачи на ветвление и циклы	решение разноуровневых задач и заданий (ПР-11)	–
3	Тема 3. Циклы перебора	ПК-1.1	Знает принцип работы цикла <i>for</i>	собеседование (УО-1)	–
			Умеет решать задачи с вложенными циклами и задачи на перебор	решение разноуровневых задач и заданий (ПР-11)	–
4	Тема 4. Работа с текстом	ПК-1.1	Знает основные функции и методы для работы со строками	собеседование (УО-1)	–
			Умеет решать задачи на обработку строк	решение разноуровневых задач и заданий (ПР-11)	–
5	Тема 5. Регулярные структуры	ПК-1.1	Знает регулярные структуры в <i>Python</i>	собеседование (УО-1)	–
			Умеет генерировать списки и двумерные массивы, а также решать задачи на обработку списков	решение разноуровневых задач и заданий (ПР-11)	–
6	Тема 6. Модульный аспект	ПК-1.1	Знает синтаксис создания функций	собеседование (УО-1)	–
			Умеет создавать и применять собственные функции	решение разноуровневых задач и заданий (ПР-11)	–
7	Тема 7. Нерегулярные структуры	ПК-1.1	Знает основные нерегулярные структуры, используемые в <i>Python</i>	собеседование (УО-1)	–
			Умеет решать задачи, используя словари, кортежи и множества	решение разноуровневых задач и заданий (ПР-11)	–
8	Зачет				ПР-1

## **VI. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Самостоятельная работа определяется как индивидуальная или коллективная учебная деятельность, осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем. Самостоятельная работа – это познавательная учебная деятельность, когда последовательность мышления студента, его умственных и практических операций и действий зависит и определяется самим студентом.

Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня, что в итоге приводит к развитию навыка самостоятельного планирования и реализации деятельности.

Целью самостоятельной работы студентов является овладение необходимыми компетенциями по своему направлению подготовки, опытом творческой и исследовательской деятельности.

Формы самостоятельной работы студентов:

- работа с основной и дополнительной литературой, Интернет ресурсами;
- самостоятельное ознакомление с лекционным материалом, представленным на электронных носителях, в библиотеке образовательного учреждения;
- подготовка реферативных обзоров источников периодической печати, опорных конспектов, заранее определенных преподавателем;
- поиск информации по теме, с последующим ее представлением в аудитории в форме доклада, презентаций;
- подготовка к выполнению аудиторных контрольных работ;
- выполнение домашних контрольных работ;
- выполнение тестовых заданий, решение задач;
- составление кроссвордов, схем;

- подготовка сообщений к выступлению на семинаре, конференции;
- заполнение рабочей тетради;
- написание эссе, курсовой работы;
- подготовка к деловым и ролевым играм;
- составление резюме;
- подготовка к зачетам и экзаменам;
- другие виды деятельности, организуемые и осуществляемые образовательным учреждением и органами студенческого самоуправления.

## **VII. СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

1. Бизли, Д. *Python*. Книга рецептов / Д. Бизли, Б. К. Джонс ; перевод с английского Б. В. Уварова. – Москва : ДМК Пресс, 2019. – 646 с. – ISBN 978-5-97060-751-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131723>
2. Жуков, Р. А. Язык программирования *Python*. Практикум : учебное пособие / Р. А. Жуков. – Москва : ИНФРА-М, 2019. – 216 с. – (Высшее образование: Бакалавриат). - ISBN 978-5-16-014701-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1000002>
3. Лучано, Р. *Python*. К вершинам мастерства / Р. Лучано ; перевод с английского А. А. Слинкин. – Москва : ДМК Пресс, 2016. – 768 с. – ISBN 978-5-97060-384-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93273>
4. Маккинли, Уэс *Python* и анализ данных / Уэс Маккинли ; перевод А. Слинкина. – 2-е изд. – Саратов : Профобразование, 2019. – 482 с. – ISBN 978-5-4488-0046-7. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/88752.html>
5. Маккинни, У. *Python* и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. – 2-ое изд., испр. и доп. – Москва : ДМК Пресс,

2020. – 540 с. – ISBN 978-5-97060-590-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131721>

6. Сузи, Р. А. Язык программирования *Python* : учебное пособие / Р. А. Сузи. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 350 с. – ISBN 978-5-4497-0705-5. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/97589.html>

#### **Дополнительная литература**

1. Буйначев, С. К. Основы программирования на языке *Python* : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг ; под редакцией Ю. В. Песин. – Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. – 92 с. – ISBN 978-5-7996-1198-9. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/66183.html>

2. Гуриков, С. Р. Основы алгоритмизации и программирования на *Python* : учеб. пособие / С.Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2017. – 343 с. – (Высшее образование: Бакалавриат). - ISBN 978-5-00091-428-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/772265>

3. Дроботун, Н. В. Алгоритмизация и программирование. Язык *Python* : учебное пособие / Н. В. Дроботун, Е. О. Рудков, Н. А. Баев. – Санкт-Петербург : Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020. – 119 с. – ISBN 978-5-7937-1829-5. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/102400.html>

4. Златопольский, Д. М. Основы программирования на языке *Python* / Д. М. Златопольский. – 2-ое изд., испр. и доп. – Москва : ДМК Пресс, 2018. – 396 с. – ISBN 978-5-97060-641-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131683>

5. Коэльо, Л. П. Построение систем машинного обучения на языке *Python* / Л. П. Коэльо, В. Ричарт ; перевод с английского А. А. Слинкин. – 2-е изд. – Москва : ДМК Пресс, 2016. – 302 с. – ISBN 978-5-97060-330-7. – Текст :

электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/82818>

6. Мартин, О. Байесовский анализ на *Python* : руководство / О. Мартин ; перевод с английского А. В. Снастина. – Москва : ДМК Пресс, 2020. – 340 с. – ISBN 978-5-97060-768-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/140585>

7. Саммерфилд, М. *Python* на практике : учебное пособие / М. Саммерфилд ; перевод с английского А. А. Слинкин. – Москва : ДМК Пресс, 2014. – 338 с. – ISBN 978-5-97060-095-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/66480>

8. Федоров, Д. Ю. Программирование на языке высокого уровня *Python* : учебное пособие для вузов / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2021. – 161 с. – (Высшее образование). – ISBN 978-5-534-10971-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/472985>

9. Шарден, Б. Крупномасштабное машинное обучение вместе с *Python* : учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти ; перевод с английского А. В. Логунова. – Москва : ДМК Пресс, 2018. – 358 с. – ISBN 978-5-97060-506-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/105836>

10. Шелудько, В. М. Основы программирования на языке высокого уровня *Python* : учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. – 146 с. – ISBN 978-5-9275-2649-9. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/87461.html>

11. Ян, Э. С. Программирование компьютерного зрения на языке *Python* / Э. С. Ян ; перевод с английского А. А. Слинкин. – Москва : ДМК Пресс, 2016. – 312 с. – ISBN 978-5-97060-200-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93569>

## Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронная библиотека и базы данных ДВФУ.  
<http://dvfu.ru/web/library/elib>
2. Электронно-библиотечная система «Лань» <http://e.lanbook.com>
3. Электронно-библиотечная система «Научно-издательского центра ИНФРА-М» <http://znanium.com>
4. Электронно-библиотечная система БиблиоТех.  
<http://www.bibliotech.ru>
5. Электронный каталог научной библиотеки ДВФУ <http://ini-fb.dvfu.ru:8000/cgi-bin/gw/chameleon>

## Перечень информационных технологий и программного обеспечения

В процессе обучения бакалавров дисциплине «Питон (Python)» используются следующие программные продукты:

- язык *Python*;
- среда разработки *Jupyter Notebook*.

## VIII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Успешное освоение дисциплины предполагает активную работу студентов на всех занятиях аудиторной формы: лекциях и практиках, выполнение аттестационных мероприятий. В процессе изучения дисциплины студенту необходимо ориентироваться на проработку лекционного материала, подготовку к практическим занятиям, выполнение контрольных и индивидуальных работ.

Освоение дисциплины предполагает рейтинговую систему оценки знаний студентов и предусматривает со стороны преподавателя текущий контроль за посещением студентами лекций, подготовкой и выполнением всех практических заданий, выполнением всех видов самостоятельной работы.

Промежуточной аттестацией по дисциплине «Питон (Python)» является

зачет (2 семестр).

Студент считается аттестованным по дисциплине при условии выполнения всех видов текущего контроля и самостоятельной работы, предусмотренных учебной программой.

Критерии оценки по дисциплине для аттестации на экзамене следующие: 86-100 баллов – «отлично», 76-85 баллов – «хорошо», 61-75 баллов – «удовлетворительно», 60 и менее баллов – «неудовлетворительно».

Пересчет баллов по текущему контролю и самостоятельной работе производится по формуле:

$$P(n) = \sum_{i=1}^m \left[ \frac{O_i}{O_i^{max}} \times \frac{k_i}{W} \right],$$

где:  $W = \sum_{i=1}^n k_i^n$  для текущего рейтинга;

$W = \sum_{i=1}^m k_i^n$  для итогового рейтинга;

$P(n)$  – рейтинг студента;

$m$  – общее количество контрольных мероприятий;

$n$  – количество проведенных контрольных мероприятий;

$O_i$  – балл, полученный студентом на  $i$ -ом контрольном мероприятии;

$O_i^{max}$  – максимально возможный балл студента по  $i$ -му контрольному мероприятию;

$k_i$  – весовой коэффициент  $i$ -го контрольного мероприятия;

$k_i^n$  – весовой коэффициент  $i$ -го контрольного мероприятия, если оно является основным, или 0, если оно является дополнительным.

Приступая к изучению дисциплины, студенту необходимо в первую очередь ознакомиться с содержанием РПД. Лекции имеют целью дать систематизированные основы научных знаний о изучаемом предмете, основных методах исследования, применяющихся в экономической статистике.

При изучении и проработке теоретического материала студентам необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;

- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД литературные источники.

- при подготовке к текущему и промежуточному контролю использовать материалы РПД.

Практические занятия проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы над учебной и научной литературой. При подготовке к практическому занятию для студентов очной формы обучения необходимо:

- изучить, повторить теоретический материал по заданной теме;

- изучить материалы практического решения задач по заданной теме, уделяя особое внимание расчетным формулам;

- при выполнении домашних расчетных заданий, изучить, повторить типовые задания, выполняемые в аудитории.

Работа с учебной и научной литературой является главной формой самостоятельной работы и необходима при подготовке к устному опросу на практических, к контрольным работам, тестированию, экзамену. Она включает проработку лекционного материала – изучение рекомендованных источников и литературы по тематике лекций. Конспект лекции должен содержать реферативную запись основных вопросов лекции, предложенных преподавателем схем (при их демонстрации), основных источников и литературы по темам, выводы по каждому вопросу. Конспект должен быть выполнен в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны быть выполнены также аккуратно, содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой

научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным. Объем конспекта определяется самим студентом.

В процессе работы с учебной и научной литературой студент может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);
- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

Работу с литературой следует начинать с анализа соответствующего раздела РПД, в котором перечислены основная и дополнительная литература, учебно-методические издания необходимые для изучения дисциплины и работы на практических занятиях.

Выбрав нужный источник, следует найти интересующий раздел по оглавлению или алфавитному указателю, а также одноименный раздел конспекта лекций или учебного пособия. В случае возникших затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным. Необходимо отметить, что работа с литературой не только полезна как средство более глубокого изучения любой дисциплины, но и является неотъемлемой частью профессиональной деятельности будущего выпускника.

Методические указания по организации внеаудиторной самостоятельной работы на занятии способствуют организации последовательного изучения материала, вынесенного на самостоятельное освоение в соответствии с учебным планом, и имеет такую структуру как:

- тема;
- вопросы и содержание материала для самостоятельного изучения;
- форма выполнения задания;

- алгоритм выполнения и оформления самостоятельной работы;
- критерии оценки самостоятельной работы.

Самостоятельная работа как вид деятельности студента многогранна. В качестве форм самостоятельной работы при изучении дисциплины «Анализ данных в R (продвинутый уровень)» предлагаются:

- работа с научной и учебной литературой;
- подготовка к практическому занятию;
- более глубокое изучение с вопросами, изучаемыми на практических занятиях;
- подготовка индивидуального и групповых заданий
- подготовка к тестированию и экзамену;

Задачи самостоятельной работы:

- обретение навыков самостоятельной научно-исследовательской работы на основании анализа текстов литературных источников и применения различных методов исследования;
- выработка умения самостоятельно и критически подходить к изучаемому материалу.

Технология СР должна обеспечивать овладение знаниями, закрепление и систематизацию знаний, формирование умений и навыков. Апробированная технология характеризуется алгоритмом, который включает следующие логически связанные действия студента:

- чтение текста (учебника, пособия, конспекта лекций);
- конспектирование текста;
- решение задач и упражнений;
- ответы на контрольные вопросы;
- составление планов и тезисов ответа.
- рекомендации по написанию группового домашнего задания

Тема задания выбирается в соответствии с интересами студента. По результатам работы группа студентов сдает преподавателю отчет о

проделанной работе. Важно, чтобы в отчете, во-первых, были освещены как естественнонаучные, так и социальные стороны проблемы; а во-вторых, представлены теоретические положения и конкретные примеры. Особенно приветствуется использование собственных примеров из окружающей студентов жизни.

Отчет должен основываться на проработке нескольких дополнительных к основной литературе источников. Как правило, это научные монографии или статьи. План отчета должен быть авторским. Все приводимые в отчете факты и заимствованные соображения должны сопровождаться ссылками на источник информации. Недопустимо просто скомпоновать реферат из кусков заимствованного текста. Все цитаты должны быть представлены в кавычках с указанием в скобках источника и страницы.

Отчет оформляется в соответствии с требованиями ДВФУ к оформлению курсовых и выпускных квалификационных работ. Сам текст работы желательно подразделить на разделы. Завершают отчет разделы «Заключение» и «Список использованной литературы». В заключении должны быть представлены основные выводы, ясно сформулированные в тезисной форме и, обычно, пронумерованные. Список литературы должен быть составлен в полном соответствии с действующим стандартом (правилами), включая особую расстановку знаков препинания.

## **IX. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Учебные занятия по дисциплине проводятся в помещениях, оснащенных соответствующим оборудованием и программным обеспечением.

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

### **Материально-техническое и программное обеспечение дисциплины**

Наименование специальных помещений и помещений	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения.
--	---	--

для самостоятельной работы		Реквизиты подтверждающего документа
<p>690922, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корп. G, ауд. G718, учебная аудитория для проведения практических занятий; учебная аудитория для текущего контроля и промежуточной аттестации</p>	<p>16 посадочных мест, компьютерный класс с мультимедийным оборудованием, автоматизированное рабочее место преподавателя, переносная магнитно-маркерная доска, Wi-Fi; Подсистема специализированных креплений оборудования CORSA-2007 Tuarex; Подсистема видеокмутации; Подсистема аудиокмутации и звукоусиления; акустическая система для потолочного монтажа SI ЗСТ LP Extron; цифровой аудиопроцессор DMP 44 LC Extron.</p>	<p>Jupyter Notebook – бесплатное ПО</p>
<p>690922, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус А - уровень 10, каб.А 1002, помещение для самостоятельной работы Читальный зал естественных и технических наук с открытым доступом Научной библиотеки</p>	<p>Моноблок Lenovo C360G-i34164G500UDK – 58 шт.</p>	<p>ЭУ0198072_ЭА-667-17_08.02.2018_Арт-Лайн Технолоджи_ПО ADOBE, ЭУ0201024_ЭА-091-18_24.04.2018_Софтлайн Проекты_ПО ESET NOD32, ЭУ0205486_ЭА-261-18_02.08.2018_СофтЛайн Трейд_ПО Microsoft</p>
<p>690922, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус А - уровень 10, каб. А1042, помещение для самостоятельной работы Читальный зал гуманитарных наук с открытым доступом Научной библиотеки</p>	<p>Моноблок Lenovo C360G-i34164G500UDK – 115 шт.  Интегрированный сенсорный дисплей Polymedia FlipBox  Копир-принтер-цветной сканер в e-mail с 4 лотками Xerox WorkCentre 5330 (WC5330C)  Полноцветный копир-принтер-сканер Xerox WorkCentre 7530 (WC7530CPS)  <b>Оборудование для инвалидов и лиц с ограниченными возможностями здоровья:</b>  Дисплей Брайля Focus-40 Blue – 3 шт.  Дисплей Брайля Focus-80 Blue  Рабочая станция Lenovo ThinkCentre E73z – 3 шт.  Видео увеличитель ONYX Swing-Arm PC edition  Маркер-диктофон Touch Memo цифровой  Устройство портативное для чтения плоскочечатных текстов PEarl  Сканирующая и читающая машина для незрячих и слабовидящих пользователей SARA  Принтер Брайля Emprint SpotDot - 2 шт.  Принтер Брайля Everest - D V4  Видео увеличитель ONYX Swing-Arm PC edition</p>	<p>ЭУ0198072_ЭА-667-17_08.02.2018_Арт-Лайн Технолоджи_ПО ADOBE, ЭУ0201024_ЭА-091-18_24.04.2018_Софтлайн Проекты_ПО ESET NOD32, ЭУ0205486_ЭА-261-18_02.08.2018_СофтЛайн Трейд_ПО Microsoft</p>

	Видео увеличитель Topraz 24" XL стационарный электронный Обучающая система для детей тактильно-речевая, либо для людей с ограниченными возможностями здоровья Увеличитель ручной видео RUBY портативный – 2шт. Экран Samsung S23C200B Маркер-диктофон Touch Мемо цифровой	
690091, г. Владивосток, ул. Алеутская 65б, лит. А, А1, Этаж 2, зл.203, помещение для самостоятельной работы. Универсальный читальный зал	Моноблок Lenovo C360G-i34164G500UDK Персональные системы для читальных залов терминала – 12 шт. Рабочее место для медиа-зала HP dc7700 – 2 шт. Персональные системы для медиа-зала в комплекте - 7 шт.	ЭУ0198072_ЭА-667-17_08.02.2018_Арт-Лайн Технологии_ПО ADOBE, ЭУ0201024_ЭА-091-18_24.04.2018_Софтлайн Проекты_ПО ESET NOD32, ЭУ0205486_ЭА-261-18_02.08.2018_СофтЛайн Трейд_ПО Microsoft
690091, г. Владивосток, ул. Алеутская 65б, лит. А, А1, Этаж 2, зл.303, помещение для самостоятельной работы. Зал доступа к электронным ресурсам	Персональные системы для читальных залов терминала – 15 шт.	ЭУ0198072_ЭА-667-17_08.02.2018_Арт-Лайн Технологии_ПО ADOBE, ЭУ0201024_ЭА-091-18_24.04.2018_Софтлайн Проекты_ПО ESET NOD32, ЭУ0205486_ЭА-261-18_02.08.2018_СофтЛайн Трейд_ПО Microsoft

## X. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

№ п/п	Контролируемые разделы / темы дисциплины	Код и наименование индикатора достижения	Результаты обучения	Оценочные средства	
				текущий контроль	Промежуточная аттестация
1	Тема 1. Введение в <i>Python</i>	ПК-1.1	Знает основные среды разработки на языке <i>Python</i> , области применимости языка	собеседование (УО-1)	–
			Умеет отличать типы данных и проводить над ними корректные арифметические действия	решение разноуровневых задач и заданий (ПР-11)	–
2	Тема 2. Ветвления и	ПК-1.1	Знает принцип работы условных	собеседование (УО-1)	–

	цикл с предусловием		конструкций и цикла <i>while</i> в <i>Python</i>		
			Умеет решать задачи на ветвление и циклы	решение разноуровневых задач и заданий (ПР-11)	–
3	Тема 3. Циклы перебора	ПК-1.1	Знает принцип работы цикла <i>for</i>	собеседование (УО-1)	–
			Умеет решать задачи с вложенными циклами и задачи на перебор	решение разноуровневых задач и заданий (ПР-11)	–
4	Тема 4. Работа с текстом	ПК-1.1	Знает основные функции и методы для работы со строками	собеседование (УО-1)	–
			Умеет решать задачи на обработку строк	решение разноуровневых задач и заданий (ПР-11)	–
5	Тема 5. Регулярные структуры	ПК-1.1	Знает регулярные структуры в <i>Python</i>	собеседование (УО-1)	–
			Умеет генерировать списки и двумерные массивы, а также решать задачи на обработку списков	решение разноуровневых задач и заданий (ПР-11)	–
6	Тема 6. Модульный аспект	ПК-1.1	Знает синтаксис создания функций	собеседование (УО-1)	–
			Умеет создавать и применять собственные функции	решение разноуровневых задач и заданий (ПР-11)	–
7		ПК-1.1	Знает основные	собеседование (УО-1)	–

	Тема 7. Нерегулярные структуры		нерегулярные структуры, используемые в <i>Python</i>		
			Умеет решать задачи, используя словари, кортежи и множества	решение разноуровнев ых задач и заданий (ПР- 11)	–
8	Зачет				ПР-1

### I. Текущая аттестация по дисциплине «Питон (Python)»

Текущая аттестация студентов по дисциплине «Питон (Python)» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация по дисциплине «Питон (Python)» проводится в форме контрольных мероприятий (собеседования, решения разноуровневых задач и заданий) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

По каждому объекту дается характеристика процедур оценивания в привязке к используемым оценочным средствам.

#### Оценочные средства для текущего контроля

##### 1. Вопросы для собеседования по всем темам курса

1. История создания языка *Python*.
2. Отличия *Python 2* и *Python 3*.
3. Основные среды разработки.
4. Области применимости *Python*.
5. Ввод-вывод.
6. Базовые арифметические операции.
7. Типы данных в *Python*.
8. Условная инструкция.
9. Цикл *while*.
10. Цикл *for*.
11. Строки и символы. Основные методы у строк.
12. Списки и их методы.
13. Двумерные массивы.
14. Функции и рекурсия.
15. Передача аргументов в функцию.
16. Кортежи и их методы.
17. Словари и их методы.
18. Множества и их методы. Встроенные типы *set* и *frozenset*.

##### Ключи (ответы) на вопросы для собеседования:

Уровень освоения	Критерии оценки результатов обучения	Кол-во баллов
повышенный	Студент глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими	86-100

	видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.	
базовый	Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.	76-85
пороговый	Студент имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при ответах на дополнительные вопросы.	61-75
уровень не достигнут	Студент не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.	0-60

## 2. Комплект типовых заданий для решения разноуровневых задач

### Темы 1 «Введение в Python»

**Задача 1.** Тимофей обычно спит ночью  $X$  часов и устраивает себе днем тихий час на  $Y$  минут. Определите, сколько всего минут Тимофей спит в сутки. Внимание, программа принимает значения  $X$  и  $Y$  из стандартного потока ввода (функция `input`), результат надо выводить в стандартный поток вывода (функция `print`).

**Задача 2.** Максим каждый день ложится спать ровно в полночь и недавно узнал, что оптимальное время для его сна составляет  $X$  минут. Максим хочет поставить себе будильник так, чтобы он прозвенел ровно через  $X$  минут после полуночи, однако для этого необходимо указать время сигнала в формате часы, минуты. Помогите Максиму определить, на какое время завести будильник. Часы и минуты в выводе программы должны располагаться на разных строках (см. пример работы программы).

**Задача 3.** Катя узнала, что ей для сна надо  $X$  минут. В отличие от Максима, Катя ложится спать после полуночи в  $N$  часов и  $M$  минут. Помогите Кате определить, на какое время ей поставить будильник, чтобы он прозвенел ровно через  $X$  минут после того, как она ляжет спать. На стандартный ввод, каждое в своей строке, подаются значения  $X$ ,  $N$  и  $M$ . Гарантируется, что Катя должна проснуться в тот же день, что и заснуть. Программа должна выводить время, на которое нужно поставить будильник: в первой строке часы, во второй – минуты.

**Ключи (ответы) на задания:**

**Задача 1:**

```
X = int(input())
Y = int(input())
print(X*60 + Y)
```

**Задача 2:**

```
X = int(input())
print(X // 60)
print(X % 60)
```

**Задача 3:**

```
X = int(input())
H = int(input())
M = int(input())
W = H * 60 + M + X
print(W // 60, W % 60, sep = "\n")
```

## **Тема 2 «Ветвления и цикл с предусловием»**

**Задача 1.** Напечатайте все нечетные числа от 1 до 100.

**Задача 2.** Посчитайте сумму целых чисел в диапазоне от *a* до *b*.

**Задача 3.** Напишите программу, которая считывает целые числа с консоли по одному числу в строке. Для каждого введённого числа проверить:

- если число меньше 10, то пропускаем это число;
- если число больше 100, то прекращаем считывать числа;
- в остальных случаях вывести это число обратно на консоль в отдельной строке.

**Ключи (ответы) на задания:**

**Задача 1:**

```
i = 1
while i <= 100:
    print(i)
    i += 2
```

**Задача 2:**

```
a = int(input())
b = int(input())
s = 0
i = a
while i <= b:
    s += i
    i += 1
print(s)
```

**Задача 3:**

```
while True:
    a = int(input())
    if 10 <= a <= 100:
        print(a)
    elif a > 100:
        break
```

## **Тема 3 «Циклы перебора»**

**Задача 1.** Пользователь вводит число. Напишите программу, которая проверяет его на четность. В случае, если число четно, на экране должно быть выведено слово «Четное», в противном случае – «Нечетное».

**Задача 2.** Напишите программу, которая считывает с клавиатуры два числа *a* и *b*, считает и выводит на консоль среднее арифметическое всех целых чисел из отрезка [a, b], которые кратны числу 3.

**Ключи (ответы) на задания:**

**Задача 1:**

```
num = int(input("Введи число: "))
if (num % 2) == 0:
    print('Четное')
else:
    print('Нечетное')
```

**Задача 2:**

```
a = int(input())
b = int(input())
s = 0
z = 0
for i in range(a, b + 1):
    if i % 3 == 0:
        s = s + i
        z = z + 1
print(s / z)
```

#### **Тема 4 «Работа с текстом»**

**Задача.** Дана строка «Abrakadabra». Необходимо:

1. Сначала выведите третий символ этой строки.
2. Во второй строке выведите предпоследний символ этой строки.
3. В третьей строке выведите первые пять символов этой строки.
4. В четвертой строке выведите всю строку, кроме последних двух символов.
5. В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).
6. В шестой строке выведите все символы с нечетными индексами, то есть начиная со второго символа строки.
7. В седьмой строке выведите все символы в обратном порядке.
8. В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.
9. В девятой строке выведите длину данной строки.

**Ключи (ответы) на задания:**

1. r
2. r
3. Abrak
4. Abrakadab
5. Arkdba
6. baaar
7. arbadakarbA
8. abdkrA
9. 11

#### **Тема 5 «Регулярные структуры»**

**Задача 1.** Напишите программу, на вход которой подается одна строка с целыми числами. Программа должна вывести сумму этих чисел.

**Задача 2.** Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящийся на противоположном конце этого списка. Например, если на вход подаётся список "1 3 5 6 10", то на выход ожидается список "13 6 9 15 7" (без кавычек). Если на вход пришло только одно число, надо вывести его же. Вывод должен содержать одну строку с числами нового списка, разделёнными пробелом.

**Задача 3.** Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящийся на противоположном конце этого списка. Например, если на вход подаётся список "1 3 5 6 10", то на выход ожидается список "13 6 9 15 7" (без кавычек). Если на вход пришло только одно число, надо вывести его же. Вывод должен содержать одну строку с числами нового списка, разделёнными пробелом.

**Ключи (ответы) на задания:**

**Задача 1:**

Sample Input: 4 -1 9 3

Sample Output: 15

**Задача 2:**

Sample Input 1: 1 3 5 6 10

Sample Output 1: 13 6 9 15 7

Sample Input 2: 10

Sample Output 2: 10

**Задача 3:**

Sample Input 1: 4 8 0 3 4 2 0 3

Sample Output 1: 0 3 4

Sample Input 2: 10

Sample Output 2:

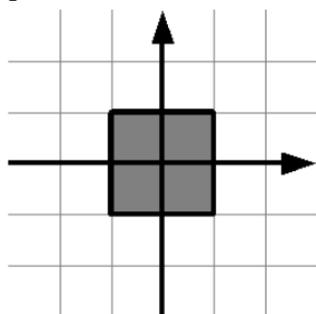
Sample Input 3: 1 1 2 2 3 3

Sample Output 3: 1 2 3

## Тема 6 «Модульный аспект»

**Задача 1.** Напишите функцию  $\text{min4}(a, b, c, d)$ , вычисляющую минимум четырех чисел.

**Задача 2.** Даны два действительных числа  $x$  и  $y$ . Проверьте, принадлежит ли точка с координатами  $(x, y)$  заштрихованному квадрату (включая его границу). Если точка принадлежит квадрату, выведите слово *YES*, иначе выведите слово *NO*. На рисунке сетка проведена с шагом 1.



Решение должно содержать функцию  $\text{IsPointInSquare}(x, y)$ , возвращающую *YES*, если точка принадлежит квадрату и *NO*, если не принадлежит.

**Ключи (ответы) на задания:**

**Задача 1:**

Sample Input: 4 -1 9 3

Sample Output: -1

**Задача 2:**

Sample Input: 4

1

Sample Output: NO

**Тема 7 «Нерегулярные структуры»**

**Задача 1.** Даны два словаря: `dictionary_1 = {'a': 300, 'b': 400}` и `dictionary_2 = {'c': 500, 'd': 600}`. Объедините их в один при помощи встроенных функций языка Python.

**Задача 2.** Создайте словарь, в котором ключами будут числа от 1 до 10, а значениями эти же числа, возведенные в куб.

**Задача 3.** Создайте словарь из строки 'pythonist' следующим образом: в качестве ключей возьмите буквы строки, а значениями пусть будут числа, соответствующие количеству вхождений данной буквы в строку.

**Ключи (ответы) на задания:****Задача 1:**

```
dictionary_1 = {'a': 100, 'b': 200}
dictionary_2 = {'x': 300, 'y': 200}
dictionary_3 = dictionary_1.copy()
dictionary_3.update(dictionary_2)
print(dictionary_3)
```

**Задача 2:**

```
my_dict = {i : i ** 3 for i in range(1, 11)}
print(my_dict)
```

**Задача 3:**

```
str1 = 'pythonist'
my_dict = {i: str1.count(i) for i in str1}
print(my_dict)
```

**Критерии оценки решения разноуровневых задач**

Оценка	Описание
<i>Отлично / зачтено</i>	Задания выполнены полностью и абсолютно правильно.
<i>Хорошо / зачтено</i>	Задания выполнены полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки.
<i>Удовлетворительно / зачтено</i>	Задания выполнены не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
<i>Неудовлетворительно / незачтено</i>	Задания не выполнены или задания выполнены частично (менее 50 процентов), имеются грубые ошибки.

## II. Промежуточная аттестация по дисциплине «Питон (Python)»

Промежуточная аттестация студентов. Промежуточная аттестация студентов по дисциплине «Питон (Python)» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

### Оценочные средства для промежуточного контроля

#### 1. Банк тестовых заданий

**Вопрос 1.** Какие ошибки допущены в коде ниже?

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n - 1)
```

```
print(factorial(5))
```

- Функция не может вызывать сама себя
- Необходимо указать тип возвращаемого значения
- Функция всегда будет возвращать 1
- В коде нет никаких ошибок

**Вопрос 2.** Как получить данные от пользователя?

- Использовать метод readLine()
- Использовать метод input()
- Использовать метод read()
- Использовать метод cin()
- Использовать метод get()

**Вопрос 3.** Что покажет этот код?

```
for j in 'Hi! I\'m mister Robert':
```

```
    if j == '\':
```

```
        print("Найдено")
```

```
        break
```

```
    else:
```

```
        print ("Готово")
```

- "Готово"
- "Найдено"
- "Найдено" и "Готово"
- Ошибку в коде

**Вопрос 4.** Сколько библиотек можно импортировать в один проект?

- Не более 10
- Не более 23
- Не более 3
- Не более 5
- Неограниченное количество

**Вопрос 5.** Где правильно создана переменная?

- var num = 2
- num = float(2)
- Нет подходящего варианта
- \$num = 2
- int num = 2

**Вопрос 6.** Что будет результатом этого кода?

```
x = 23
```

```
num = 0 if x > 10 else 11
print(num)
```

- a. 23
- b. Ошибка
- c. 0
- d. 11
- e. 10

**Вопрос 7.** Какая библиотека отвечает за время?

- a. Time
- b. time
- c. clock
- d. localtime

**Вопрос 8.** Что покажет этот код?

```
for i in range(5):
    if i % 2 == 0:
        continue
    print(i)
```

- a. Числа: 1 и 3
- b. Числа: 1, 3 и 5
- c. Ошибку, так как i не присвоена
- d. Ошибку из-за неверного вывода
- e. Числа: 0, 2 и 4

**Вопрос 9.** Имеется кортеж вида T = (4, 2, 3). Какая из операций приведёт к тому, что имя T будет ссылаться на кортеж (1, 2, 3)?

- a. T[0] = 1
- b. T = (1) + T[1:]
- c. T = (1,) + T[1:]
- d. T.startswith(1)

**Вопрос 10.** Для чего в Python используется встроенная функция enumerate()?

- a. Для определения количества элементов последовательности.
- b. Для одновременного итерирования по самим элементам и их индексам.
- c. Для сортировки элементов по значениям id.

**Вопрос 11.** Что выведет интерпретатор для следующей программы (версия Python 3.6+)?

```
def get_name_and_decades(name, age):
    print(f"My name is {name} and I'm {age / 10:.5f} decades old.")
```

```
get_name_and_decades("Leo", 31)
```

- a. My name is Leo and I'm 31.00000 decades old.
- b. My name is Leo and I'm 3.1 decades old.
- c. Исключение: перед строкой стоит лишняя буква f.
- d. My name is Leo and I'm 3.10000 decades old.
- e. My name is {name} and I'm {age / 10:.5f} decades old.

**Вопрос 12.** Необходимо собрать и вывести все уникальные слова из строки рекламного текста. Какой из перечисленных типов данных Python подходит лучше всего?

- a. кортеж (tuple)

- b. список (list)
- c. множество (set)
- d. словарь (dict)

**Вопрос 13.** Учёт зверей в зоопарке ведётся с помощью приведённого ниже списка словарей. Какая из строчек кода выведет структуру, отсортированную в порядке увеличения возрастов животных?

```
animals = [
    {'type': 'penguin', 'name': 'Stephanie', 'age': 8},
    {'type': 'elephant', 'name': 'Devon', 'age': 3},
    {'type': 'puma', 'name': 'Moe', 'age': 5},
]
```

- a. sorted(animals, key='age')
- b. Ни один вариант не является верным, два словаря нельзя сравнивать друг с другом.
- c. sorted(animals, key=lambda animal: animal['age'])
- d. sorted(animals)

**Вопрос 14.** Какой результат выведет следующий код?

```
def f(a, *pargs, **kargs): print(a, pargs, kargs)
f(1, 2, 3, x=4, y=5)
```

- a. 1, 2, 3, {'x': 4, 'y': 5}
- b. 1 (2, 3) {'x': 4, 'y': 5}
- c. Будет вызвано исключение, после двоеточия обязательно нужно перейти на новую строку.
- d. 1, 2, 3, 'x=4', 'y=5'
- e. 1, 2, 3, 4, 5

**Вопрос 15.** Как вывести список методов и атрибутов объекта x?

- a. help(x)
- b. info(x)
- c. ?x
- d. dir(x)

**Вопрос 16.** Как можно более кратко представить следующую запись?

```
if X:
```

```
    A = Y
```

```
else:
```

```
    A = Z
```

- a. A = Y if Z else Y
- b. A = Y if X else Z
- c. A = X if Z else Y
- d. A = X if Y else Z

**Ключи (ответы) к тестам:**

1-d, 2-b, 3-b, 4-e, 5-b, 6-c, 7-b, 8-a, 9-c, 10-b, 11-d, 12-c, 13-c, 14-b, 15-d, 16-b

## 2. Вопросы к зачету

1. История создания языка *Python*.
2. Отличия *Python 2* и *Python 3*.
3. Основные среды разработки.
4. Области применимости *Python*.

5. Ввод-вывод.
6. Базовые арифметические операции.
7. Типы данных в Python.
8. Условная инструкция.
9. Цикл *while*.
10. Цикл *for*.
11. Строки и символы. Основные методы у строк.
12. Списки и их методы.
13. Двумерные массивы.
14. Функции и рекурсия.
15. Передача аргументов в функцию.
16. Кортежи и их методы.
17. Словари и их методы.
18. Множества и их методы. Встроенные типы *set* и *frozenset*.

**Ключи (ответы) на вопросы для собеседования на экзамене:**

Уровень освоения	Критерии оценки результатов	Кол-во баллов
повышенный	Оценка «отлично» / зачтено выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.	86-100
базовый	Оценка «хорошо» / зачтено выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.	76-85
пороговый	Оценка «удовлетворительно» / зачтено выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.	61-75
уровень не достигнут	Оценка «неудовлетворительно» / не зачтено выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.	0-60