

Частное образовательное учреждение высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ  
УПРАВЛЕНИЯ И ЭКОНОМИКИ»

РАССМОТРЕНО И ОДОБРЕНО

УТВЕРЖДАЮ

На заседании кафедры  
информационных технологий и  
математики  
Протокол № 9 от 25.05.2023 г.

Первый проректор  
С.В. Авдашкевич  
28.06.2023

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина:	ФТД.02 Программирование на Python
Направление подготовки:	09.03.03 Прикладная информатика
Направленность (профиль):	Прикладная информатика в экономике
Уровень высшего образования:	Бакалавриат
Форма обучения:	очная, заочная
Разработчики:	Кандидат экономических наук, доцент Таюрская И. С. Кандидат педагогических наук, доцент Уткина О. Н.

Санкт-Петербург  
2023

### 1. Цели и задачи дисциплины:

#### *Цель освоения дисциплины:*

формирование систематизированных знаний и навыков в области программирования; ознакомление студентов с принципами работы современных языков программирования, с актуальными парадигмами программирования.

#### *Задачи дисциплины:*

– изучение основных конструкций современного языка программирования на примере Python; – изучение алгоритмов обработки массивов и других структурированных данных с использованием языка Python.

### 2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы высшего образования

Планируемые результаты освоения ОП ВО (код и содержание компетенций)	Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Примечание
ПК-2 Способен разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1 Знать инструменты и методы модульного тестирования; инструменты и методы прототипирования пользовательского интерфейса; инструменты и методы тестирования нефункциональных и функциональных характеристик ИС; методы оценки качества программных систем; основы программирования; основы современных операционных систем; основы современных систем управления базами данных; основы управления взаимоотношениями с клиентами и заказчиками (CRM); программные средства и платформы инфраструктуры информационных технологий организаций; современные стандарты информационного взаимодействия систем; современные структурные языки программирования; теория тестирования; языки программирования и работы с базами данных; языки современных бизнес-приложений.	06.015 Профессиональный стандарт «Специалист по информационным системам» 06.022 Профессиональный стандарт «Системный аналитик»
	ПК-2.2 Уметь алгоритмизировать деятельность; кодировать на языках программирования; тестировать результаты прототипирования.	
	ПК-2.3 Владеть навыками проведения анализа результатов тестов; координирования и проведения оценки готовых систем; обучения участников рабочей группы методике оценки готовых систем; оформления отчета о степени соответствия готовых систем требованиям; подготовки методике оценки готовых систем на соответствие требованиям; принятия решения о пригодности архитектуры; разработки прототипа ИС в соответствии с требованиями; осуществления сбора, обработки и анализа результатов оценки готовых систем на соответствие требованиям; согласования пользовательского интерфейса с заказчиком; тестирования прототипа ИС на проверку корректности архитектурных решений.	

Планируемые результаты освоения ОП ВО (код и содержание компетенций)	Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Примечание
ПК-3 Способен проектировать ИС по видам обеспечения	<p>ПК-3.1 Знать архитектуру, устройство и функционирование вычислительных систем; возможности ИС; инструменты и методы верификации архитектуры ИС; инструменты и методы верификации структуры программного кода; инструменты и методы проектирования архитектуры ИС; инструменты и методы проектирования и дизайна ИС; коммуникационное оборудование; основы современных операционных систем; основы современных систем управления базами данных; программные средства и платформы инфраструктуры информационных технологий организаций; сетевые протоколы; современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP..., ITIL, ITSM); современные стандарты информационного взаимодействия систем; теорию баз данных; устройство и функционирование современных ИС.</p> <p>ПК-3.2 Уметь проектировать архитектуру ИС; проверять (верифицировать) архитектуру ИС; кодировать на языках программирования; верифицировать структуру программного кода</p> <p>ПК-3.3 Владеть навыками верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС; разрабатывать архитектурную спецификацию ИС; разрабатывать структуру программного кода ИС; согласовывать архитектурную спецификацию ИС с заинтересованными сторонами; устранять обнаруженные несоответствия.</p>	06.015 Профессиональный стандарт «Специалист по информационным системам»

Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Планируемые результаты обучения по дисциплине
ПК-2.1. Знать инструменты и методы модульного тестирования; инструменты и методы прототипирования пользовательского интерфейса; инструменты и методы тестирования нефункциональных и функциональных характеристик ИС; методы оценки качества программных систем; основы программирования; основы современных операционных систем; основы современных систем управления базами данных; основы управления взаимоотношениями с клиентами и заказчиками (CRM); программные средства и платформы инфраструктуры информационных технологий организаций; современные стандарты информационного взаимодействия систем; современные структурные языки программирования; теория тестирования; языки программирования и работы с базами данных; языки современных бизнес-приложений.	Знает основы программирования на Python.
ПК-2.2. Уметь алгоритмизировать деятельность; кодировать на языках программирования; тестировать результаты прототипирования.	Умеет алгоритмизировать деятельность; кодировать на языках программирования; тестировать результаты прототипирования на Python.

Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Планируемые результаты обучения по дисциплине
ПК-2.3. Владеть навыками проведения анализа результатов тестов; ко-ординирования и проведения оценки готовых систем; обучения участников рабочей группы методике оценки готовых систем; оформления отчета о степени соответствия готовых систем требованиям; подготовки методики оценки готовых систем на со-ответствие требованиям; принятия решения о пригодности ар-хитектуры; разработки прототипа ИС в соответствии с требо-ваниями; осуществления сбора, обработки и анализа результатов оценки готовых систем на соответствие требованиям; согла-сования пользовательского интерфейса с заказчиком; тестиро-вания прототипа ИС на проверку корректности архитектурных решений.	Владеет навыками разработки прототипа ИС в соответствии с требованиями на Python.
ПК-3.1. Знать архитектуру, устройство и функционирование вычисли-тельных систем; возможности ИС; инструменты и методы верификации архитектуры ИС; инструменты и методы верифика-ции структуры программного кода; инструменты и методы проектирования архитектуры ИС; инструменты и методы про-ектирования и дизайна ИС; коммуникационное оборудование; основы современных операционных систем; основы современ-ных систем управления базами данных; программные средства и платформы инфраструктуры информационных технологий организаций; сетевые протоколы; современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP..., ITIL, ITSM); современные стандарты информационного взаимодействия систем; теорию баз данных; устройство и функционирование современных ИС.	Знает инструменты и методы верификации структуры программного кода на Python.
ПК-3.2. Уметь проектировать архитектуру ИС; проверять (верифици-ровать) архитектуру ИС; кодировать на языках программиро-вания; верифицировать структуру программного кода	Умеет кодировать на Python.
ПК-3.3. Владеть навыками верификации структуры программного ко-да ИС относительно архитектуры ИС и требований заказчика к ИС; разрабатывать архитектурную спецификацию ИС; разра-батывать структуру программного кода ИС; согласовывать архитектурную спецификацию ИС с заинтересованными сто-ронами; устранять обнаруженные несоответствия.	Владеет навыками верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС на Python.

### 3. Содержание, объем дисциплины и формы проведения занятий

№ п/п	Наименование темы дисциплины	Компетенции	Оценочные средства текущего контроля		
			ЗНАТЬ	УМЕТЬ	ВЛАДЕТЬ
			ПК-2.1 ПК-3.1	ПК-2.2 ПК-3.2	ПК-2.3 ПК-3.3
1	Основы программирования на языке Python.	ПК-2	Конспект №1 (10)	Задача №1 (10) Задача №2 (10) Собеседование, опрос/ Контрольная работа №1 (10)	Деловая и (или) ролевая игра/Кейс-задача №1 (20)

09.03.03 Прикладная информатика, направленность (профиль) "Прикладная информатика в экономике"  
 Рабочая программа дисциплины  
 Дисциплина: ФТД.02 Программирование на Python  
 Форма обучения: очная, заочная  
 Разработана для приема 2021/2022, 2022/2023 учебного года  
 Обновлено на 2023/2024 учебный год

№ п/п	Наименование темы дисциплины	Компетенции	Оценочные средства текущего контроля		
			ЗНАТЬ	УМЕТЬ	ВЛАДЕТЬ
			ПК-2.1 ПК-3.1	ПК-2.2 ПК-3.2	ПК-2.3 ПК-3.3
2	Разработка прикладных программ на Python.	ПК-3	Конспект №2 (10)	Задача №3 (10)	Деловая и (или) ролевая игра/Кейс-задача №2 (20)
<b>Количество баллов (100 баллов):</b>			100		

Содержание учебного материала, лабораторные работы и практические занятия, курсовая работа
<p><b>Тема 1:</b> Основы программирования на языке Python.                      Язык программирования Python. Структура программы. Типы данных: простые и структурированные. Условный оператор. Оператор выбора. Циклы. Структурированные типы данных.  <b>Практические занятия/самостоятельная работа:</b>                      Язык программирования Python. Структура программы. Типы данных: простые и структурированные. Условный оператор. Оператор выбора. Циклы. Структурированные типы данных.  <b>Лабораторная работа:</b> -</p>
<p><b>Тема 2:</b> Разработка прикладных программ на Python.                      Библиотеки Python. Стандартная библиотека. Сетевые возможности языка Python. Использование языка Python для математических расчётов.  <b>Практические занятия/самостоятельная работа:</b>                      Библиотеки Python. Стандартная библиотека. Сетевые возможности языка Python. Использование языка Python для математических расчётов.  <b>Лабораторная работа:</b> -</p>
<p><b>Курсовая работа:</b>                      не предусмотрено учебным планом</p>

### Очная форма обучения

Вид учебной работы	Всего часов	Семестр 7
Аудиторные занятия (АЗ):	36	36
Лекционные занятия (Лек)	18	18
Лабораторные занятия (Лаб)	0	0
Практические занятия (Пр)	18	18
Самостоятельная работа студента (СР)	33	33
Курсовая работа	0	0
Другие виды самостоятельной работы*	33	33
Контроль самостоятельной работы (КСР)	3	3
Контактная работа (КоР)	39	39
Форма промежуточной аттестации	0	Зачет
Подготовка к экзамену и сдача экзамена (СР, КоР)	0	0
Общая трудоемкость дисциплины, часы/ЗЕТ	72/2	72/2

\* Подготовка к аудиторным занятиям, подготовка к зачету (при наличии)

№	Наименование темы дисциплины	Семестр/Курс	Количество учебных часов				Практическая подготовка
			В том числе по видам аудиторных занятий			СР	
			Лек	Пр	Лаб		
1	Основы программирования на языке Python.	7	6	8	0	15	8
2	Разработка прикладных программ на Python.	7	12	10	0	18	10
Итого:			18	18	0	33	18

\* Практическая подготовка при реализации дисциплин организована путем проведения практических занятий и (или) выполнения лабораторных и (или) курсовых работ и предусматривает выполнение работ, связанных с будущей профессиональной деятельностью.

### Заочная форма обучения

Вид учебной работы	Всего часов	Семестр 7
Аудиторные занятия (АЗ):	8	8
Лекционные занятия (Лек)	4	4
Лабораторные занятия (Лаб)	0	0
Практические занятия (Пр)	4	4
Самостоятельная работа студента (СР)	57	57
Курсовая работа	0	0
Другие виды самостоятельной работы*	57	57
Контроль самостоятельной работы (КСР)	3	3
Контактная работа (КоР)	11	11
Форма промежуточной аттестации	0	Зачет
Подготовка к экзамену/зачету и сдача экзамена/зачета (СР, КоР)	4	4
Общая трудоемкость дисциплины, часы/ЗЕТ	72/2	72/2

\* Подготовка к аудиторным занятиям

№	Наименование темы дисциплины	Семестр/ Курс	Количество учебных часов				Практическая подготовка
			В том числе по видам аудиторных занятий			СР	
			Лек	Пр	Лаб		
1	Основы программирования на языке Python.	7	2	2	0	27	8
2	Разработка прикладных программ на Python.	7	2	2	0	30	10
		Итого:	4	4	0	57	18

\* Практическая подготовка при реализации дисциплин организована путем проведения практических занятий и (или) выполнения лабораторных и (или) курсовых работ и (или) путем выделения часов из часов, отведенных на самостоятельную работу, и предусматривает выполнение работ, связанных с будущей профессиональной деятельностью.

#### 4. Способ реализации дисциплины

Без использования онлайн-курса.

#### 5. Учебно-методическое обеспечение дисциплины:

*Основная литература:*

1. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ PYTHON 3-е изд., пер. и доп. Учебное пособие для вузов / Федоров Д. Ю. - Санкт-Петербургский государственный экономический университет (г. Санкт-Петербург), 2022 г. - 210 с. - ISBN 978-5-534-14638-7 – Режим доступа: <https://urait.ru/book/programmirovanie-na-yazyke-vysokogo-urovnya-python-492920>

2. Борзунов С. В., Кургалин С. Д. — Алгебра и геометрия с примерами на Python - Издательство Лань, 2021 г. - 444 с. - ISBN 978-5-8114-7961-0 – Режим доступа: <https://e.lanbook.com/book/169808>

3. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON. Учебное пособие для вузов / Чернышев С. А. - Санкт-Петербургский государственный университет промышленных технологий и дизайна (г. Санкт-Петербург); Санкт-Петербургский государственный экономический университет (г. Санкт-Петербург), 2022 г. - 286 с. - ISBN 978-5-534-14350-8 – Режим доступа: <https://urait.ru/book/osnovy-programmirovaniya-na-python-496893>

*Дополнительная литература:*

1. ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ. Учебное пособие для вузов / Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. - Санкт-Петербургский государственный экономический университет (г. Санкт-Петербург), 2022 г. - 235 с. - ISBN 978-5-534-02816-4 – Режим доступа: <https://urait.ru/book/tehnologii-i-metody-programmirovaniya-489920>

2. РАЗРАБОТКА ИНТЕРНЕТ-ПРИЛОЖЕНИЙ. Учебное пособие для вузов / Сысолетин Е. Г., Ростунцев С. Д. ; под науч. ред. Доросинского Л.Г. - Уральский федеральный университет имени первого Президента России Б.Н. Ельцина (г. Екатеринбург), 2022 г. - 90 с. - ISBN

---

978-5-9916-9975-4 – Режим доступа: <https://urait.ru/book/razrabotka-internet-prilozheniy-492224>

3. ПРОГРАММИРОВАНИЕ. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД. Учебник и практикум для вузов / Зыков С. В. - Национальный исследовательский университет «Высшая школа экономики» (г. Москва), 2022 г. - 155 с. - ISBN 978-5-534-00850-0 – Режим доступа: <https://urait.ru/book/programmirovanie-obektno-orientirovannyy-podhod-490423>

**6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения**

1. Операционная система
2. Пакет прикладных офисных программ
3. Антивирусное программное обеспечение
4. LMS Moodle
5. Вебинарная платформа
6. Python
7. PyCharm Community Edition

**7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», информационных справочных систем и профессиональных баз данных, необходимых для освоения дисциплины**

1. [ibooks.ru](http://ibooks.ru) : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://ibooks.ru>. - Текст: электронный
2. Электронно-библиотечная система СПбУТУиЭ : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <http://libume.ru>. - Текст: электронный
3. Юрайт : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://urait.ru>. - Текст: электронный
4. [eLibrary.ru](http://elibrary.ru) : научная электронная библиотека [Электронный ресурс] : профессиональная база данных. - Режим доступа: <http://elibrary.ru>. - Текст: электронный
5. Архив научных журналов НЭИКОН [Электронный ресурс] : профессиональная база данных. - Режим доступа: [arh.neicon.ru](http://arh.neicon.ru). - Текст: электронный
6. КиберЛенинка : научная электронная библиотека [Электронный ресурс] : информационная справочная система. - Режим доступа: <http://cyberleninka.ru>. - Текст: электронный
7. Лань : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://e.lanbook.com>. - Текст: электронный
8. [it-world.ru](https://www.it-world.ru) [Электронный ресурс] : информационная справочная система . - Режим доступа: <https://www.it-world.ru>. - Текст: электронный
9. Connect: IT-технологии : информационная справочная система. - Режим доступа: <https://www.connect-wit.ru/>. - Текст: электронный
10. Виртуальный компьютерный музей [Электронный ресурс] : информационная справочная система . - Режим доступа: <https://www.computer-museum.ru>. - Текст: электронный
11. Компьютерра : информационная справочная система . - Режим доступа: <https://www.computerra.ru/>. - Текст: электронный
12. Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации: профессиональная база данных. - Режим доступа: <https://digital.gov.ru>. - Текст: электронный
13. Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций: профессиональная база данных . - Режим доступа: <https://rkn.gov.ru>. - Текст: электронный
14. [Math-Net.Ru](https://www.mathnet.ru/): профессиональная база данных . - Режим доступа: <https://www.mathnet.ru/>. - Текст: электронный

## **8. Материально-техническое обеспечение дисциплины**

1. Учебная аудитория для проведения занятий лекционного типа, семинарского типа - практических занятий, для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оборудованная: рабочими местами для обучающихся, оснащенными специальной мебелью; рабочим местом преподавателя, оснащенным специальной мебелью, персональным компьютером с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; техническими средствами обучения - мультимедийным оборудованием (проектор, экран, колонки) и маркерной доской; лицензионным программным обеспечением

2. Учебная аудитория для проведения занятий семинарского типа - практических занятий – компьютерный класс, оборудованный рабочими местами для обучающихся, оснащенными специальной мебелью, персональными компьютерами с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; рабочим местом преподавателя, оснащенным специальной мебелью, персональным компьютером с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; техническими средствами обучения - мультимедийным оборудованием (проектор, экран, колонки) и маркерной доской; лицензионным программным обеспечением

3. При применении электронного обучения, дистанционных образовательных технологий используются: виртуальные аналоги учебных аудиторий - вебинарные комнаты на вебинарных платформах, рабочее место преподавателя, оснащенное персональным компьютером (планшет, мобильное устройство) с возможностью подключения к сети «Интернет», доступом к электронной информационно-образовательной среде Университета и к информационно-образовательному portalу Университета [imeos.ru](http://imeos.ru), веб-камерой, микрофоном и гарнитурой (в т.ч. интегрированными в устройства), программным обеспечением; рабочее место обучающегося оснащено персональным компьютером (планшет, мобильное устройство) с возможностью подключения к сети «Интернет», доступом к электронной информационно-образовательной среде Университета и к информационно-образовательному portalу Университета [imeos.ru](http://imeos.ru), веб-камерой, микрофоном и гарнитурой (в т.ч. интегрированными в устройства). Авторизация на информационно-образовательном portalе Университета [imeos.ru](http://imeos.ru) и начало работы осуществляются с использованием персональной учетной записи (логина и пароля). Лицензионное программное обеспечение

4. Помещение для самостоятельной работы, оборудованное специальной мебелью, персональными компьютерами с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета, лицензионным программным обеспечением

## **9. Оценочные материалы по дисциплине**

Описание оценочных средств (показатели и критерии оценивания, шкалы оценивания) представлено в приложении к основной профессиональной образовательной программе «Каталог оценочных средств текущего контроля и промежуточной аттестации».

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности приводятся в соответствующих методических материалах и локальных нормативных актах Университета.

Для оценивания учебных достижений студентов в Университете действует балльно-рейтинговая система.

Если оценка, соответствующая набранной в семестре сумме рейтинговых баллов, удовлетворяет студента, то она является итоговой оценкой по дисциплине при проведении промежуточной аттестации в форме экзамена/зачета с оценкой/зачета.

Условием сдачи экзамена/зачета с оценкой/зачета с целью повышения итоговой оценки по дисциплине является сдача студентом экзамена, за который он получает экзаменационные баллы



без учета баллов, полученных за текущий контроль:

### Шкала оценивания учебных достижений по дисциплине, завершающейся зачетом без оценки

Баллы по дисциплине	60 и менее		61-73		74-90		91-100
Итоговая оценка по дисциплине	Незачет		Зачет				
Баллы в международной шкале ECTS с буквенным обозначением уровня	50 и менее	51-60	61-67	68-73	74-83	84-90	91-100
	F	Fx	E	D	C	B	A
Уровень сформированности компетенций	Не сформированы		Пороговый		Высокий		Повышенный

### Шкала оценивания учебных достижений по дисциплине, завершающейся экзаменом/зачетом с оценкой

Баллы по дисциплине	60 и менее		61-73		74-90		91-100
Итоговая оценка по дисциплине	Неудовлетворительно		Удовлетворительно		Хорошо		Отлично
Баллы в международной шкале ECTS с буквенным обозначением уровня	<50	51-60	61-67	68-73	74-83	84-90	91-100
	F	Fx	E	D	C	B	A
Уровень сформированности компетенций	Не сформированы		Пороговый		Высокий		Повышенный

#### 9.1. Типовые контрольные задания для текущего контроля Конспект №1

1. Язык программирования Python.
2. Структура программы.
3. Типы данных: простые и структурированные.
4. Условный оператор.
5. Оператор выбора.
6. Циклы.
7. Структурированные типы данных.

#### Конспект №2

1. Библиотеки Python.
2. Стандартная библиотека.
3. Сетевые возможности языка Python.
4. Использование языка Python для математических расчётов

#### Собеседование, опрос / Контрольная работа №1

1. Что напечатает эта программа? `sum = 0 n = 5 for i in range(1, n + 1): sum += i print(sum)`
  - a) 15 (+)
  - b) 12
  - c) 0
  - d) 20
2. Укажите НЕПРАВИЛЬНЫЙ способ получения списка из 100 нулей:
  - a) `a = [0] * 100`
  - b) `a = [0 for i in range(100)]`
  - c) `a = [0 for i in range(1,100)]`

d)  $a = [0,0,0\dots,0].\text{len}(100)$  (+)

### Задача №1

1. Какие из представленных выражений можно преобразовать в целое десятичное число за одну операцию:

А) '123e';

Б) '91.4';

В)  $524.345 ** 435345345311145345$ ;

Г)  $'7.1 + 4'$ ;

Д)  $'4' - 2$ ;

Е)  $'4 - 2'$ ;

Ж) '42'

З) -12.12?

2. Напишите функцию `num_sum(a)`, принимающую любое значение.

Если это целое число, то вернуть сумму его чисел.

В противном случае возвращается фраза «Это не целое число».

3. Напишите функцию `pos_add(a, b)`, которая возвращает положительное значение сложения двух целых чисел.

4. Обозначьте порядок вычисления выражения по операциям:

$11 * 2 ** 2 - 13 / 4 + 7$ .

Какое целое число получим в итоге?

5. Дана последовательность случайных цифр любой длины и «волшебное» положительное число, больше нуля.

Напишите функцию `magic()`, принимающую эти аргументы, и выясните, можно ли разделить сумму квадратов последовательности на «волшебное» число без остатка.

В качестве ответа возвращается «Волшебство случается» в случае успеха или «Никакого волшебства», если разделить нельзя.

### Задача №2

1. Напишите функцию `to_dict(lst)`, которая принимает аргумент в виде списка и возвращает словарь, в котором каждый элемент списка является и ключом и значением. Предполагается, что элементы списка будут соответствовать правилам задания ключей в словарях.

2. Иван решил создать самый большой словарь в мире. Для этого он придумал функцию `biggest_dict(**kwargs)`, которая принимает неограниченное количество параметров «ключ: значение» и обновляет созданный им словарь `my_dict`, состоящий всего из одного элемента «first\_one» со значением «we can do it». Воссоздайте эту функцию.

3. Имеется ряд словарей с пересекающимися ключами (значения - положительные числа). Напишите 2 функции, которые делают с массивом словарей следующие операции:

1-ая функция `max_dct(*dicts)` формирует новый словарь по правилу:

Если в исходных словарях есть повторяющиеся ключи, выбираем среди их значений максимальное и присваиваем этому ключу (например, в словаре\_1 есть ключ «a» со значением 5, и в словаре\_2 есть ключ «a», но со значением 9. Выбираем максимальное значение, т. е. 9, и присваиваем ключу «a» в уже новом словаре).

Если ключ не повторяется, то он просто переносится со своим значением в новый словарь (например, ключ «c» встретился только у одного словаря, а у других его нет. Следовательно, переносим в новый словарь этот ключ вместе с его значением). Сформированный словарь возвращаем.

2-ая функция `sum_dct(*dicts)` суммирует значения повторяющихся ключей. Значения остальных ключей остаются исходными. (Проводятся операции по аналогу первой функции, но берутся не максимумы, а суммы значений одноименных ключей). Функция возвращает сформированный

словарь.

4. Создайте словарь с количеством элементов не менее 5-ти. Поменяйте местами первый и последний элемент объекта. Удалите второй элемент. Добавьте в конец ключ «new\_key» со значением «new\_value». Выведите на печать итоговый словарь. Важно, чтобы словарь остался тем же (имел тот же адрес в памяти).

5. Дана строка в виде случайной последовательности чисел от 0 до 9.

Требуется создать словарь, который в качестве ключей будет принимать данные числа (т. е. ключи будут типом int), а в качестве значений – количество этих чисел в имеющейся последовательности. Для построения словаря создайте функцию count\_it(sequence), принимающую строку из цифр. Функция должна вернуть словарь из 3-х самых часто встречаемых чисел.

### Задача №3

1. Напишите функцию read\_last(lines, file), которая будет открывать определенный файл file и выводить на печать построчно последние строки в количестве lines (на всякий случай проверим, что задано положительное целое число).

Протестируем функцию на файле «article.txt» со следующим содержимым:

Вечерело

Жужжали мухи

Светил фонарик

Кипела вода в чайнике

Венера зажглась на небе

Деревья шумели

Тучи разошлись

Листва зеленела

2. Выберите любую папку на своем компьютере, имеющую вложенные директории.

Выведите на печать в терминал ее содержимое, как и всех подкаталогов при помощи функции print\_docs(directory).

3. Документ «article.txt» содержит следующий текст:

Вечерело

Жужжали мухи

Светил фонарик

Кипела вода в чайнике

Венера зажглась на небе

Деревья шумели

Тучи разошлись

Листва зеленела

Требуется реализовать функцию longest\_words(file), которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

4. Требуется создать csv-файл «rows\_300.csv» со следующими столбцами:

– № - номер по порядку (от 1 до 300);

– Секунда – текущая секунда на вашем ПК;

– Микросекунда – текущая миллисекунда на часах.

На каждой итерации цикла искусственно приостанавливайте скрипт на 0,01 секунды.

5. При помощи библиотеки Pillow в директории circles (создайте ее во время выполнения функции) нарисуйте и сохраните 100 кругов радиусом 300 пикселей случайных цветов в формате jpg на белом фоне (каждый круг - отдельный файл). Для этого напишите функцию circles\_generator(num\_of\_circles=100).

### Деловая и (или) ролевая игра / Кейс-задача №1

1. Напишите программу, которая последовательно запрашивает с клавиатуры три целых числа

(после ввода каждого числа пользователь нажимает *Enter*— иными словами, каждое число вводится на отдельной строке) и выводит их произведение.

2. Дан список станций желтой ветки московского метро (упорядочены как на карте):

`line = ["Третьяковская", "Марксистская", "Площадь Ильича", "Авиамоторная", "Шоссе Энтузиастов", "Перово", "Новогиреево", "Новокосино"]`

Считайте, что движение происходит от Третьяковской до Новокосино. Напишите программу, которая запрашивает у пользователя название текущей станции метро (без лишнего пробела на конце) и выводит на экран сообщение вида

Следующая станция: станция.

Считайте, что пользователь знает, какая станция конечная, и не будет запрашивать станцию, следующую после Новокосино.

3. В списке  $L$  содержатся целые числа. Создать новый список  $M$ , содержащий удвоенные элементы  $L$ . Список  $L$  при этом не должен измениться. Например, для списка  $L=[12, 4, 16, 19, 1]$ , после выполнения программы должно выполняться  $M==[24, 8, 32, 38, 2]$ .

4. Напишите функцию *dummy(f)*, которая принимает на вход список значений индекса Freedom House  $f$  и возвращает список из 0 и 1, где 1 соответствует свободным странам (статус *Free*, значения индекса от 1.0 до 2.5 включительно). Функция не должна ничего печатать (выводить на экран).

5. Скачайте базу данных, содержащую результаты выборов в Государственную Думу 2016 года по всем регионам России (csv-файл). Сохраните базу данных в переменную *df*. Создайте переменную *turnout* (явка на выборы). Явка определяется как сумма действительных и недействительных бюллетеней на выборах. Создайте переменную *turnout\_perc* (процент явки на выборы). Процент явки на выборы считается как показатель явки, деленный на число зарегистрированных избирателей.

### Деловая и (или) ролевая игра / Кейс-задача №2

1. Дан список целых чисел. Требуется “сжать” его, переместив все ненулевые элементы в левую часть списка, не меняя их порядок, а все нули - в правую часть. Порядок ненулевых элементов изменять нельзя, дополнительный список использовать нельзя, задачу нужно выполнить за один проход по списку. Распечатайте полученный список. Пример ввода: 4 0 5 0 0 3 2 0 5 0 Пример вывода: 4 5 3 2 5 0 0 0 0

2. Составить программу, моделирующую алгоритм группового сжатия данных RLE – замену нескольких подряд стоящих одинаковых элементов списка – парой вида: элемент, количество, а также программу обратного преобразования.

3. Игра «Цепочка». В заданном списке слов найти все цепочки слов, в которых каждое слово (кроме первого) отличается от предыдущего ровно одним символом: Пример ввода: [муха, дуб, муза, кус, куб, сук, лупа, луза, вино, лето, кэб, лото]. Пример вывода: [[муха, муза, луза, лупа], [дуб, куб, кус, кэб], [сук], [вино], [лето, лото]].

4. Описать графом городскую маршрутную сеть и составить правила поиска оптимального выбора маршрутов общественного транспорта для поездки между двумя заданными пунктами.

## 9.2. Примерный перечень тем курсовой работы

Не предусмотрено учебным планом

## 9.3. Типовые контрольные задания для промежуточной аттестации: зачет

### Примерный перечень теоретических вопросов к зачету

1. Язык программирования Python.
2. Структура программы.
3. Типы данных: простые и структурированные.

4. Условный оператор.
5. Оператор выбора.
6. Циклы.
7. Структурированные типы данных.
8. Библиотеки Python.
9. Стандартная библиотека.
10. Сетевые возможности языка Python.
11. Использование языка Python для математических расчетов.

### Примерный перечень практических заданий к зачету

Выполнить задание.

1. Ознакомиться с интерактивным и сценарным режимами работы среды разработки программ на языке Python (IDLE).

2. Разработать программу на языке Python, которая выполняет следующее:

- с помощью встроенной системы помощи (help>) получает информацию о встроенной функции или операции, указанной в колонке «Help» табл. 5 (если объем данных большой – привести фрагмент полученных данных):

- 1 – + ;
- 2 – - ;
- 3 – / ;
- 4 – // ;
- 5 – % ;
- 6 – \*\* ;
- 7 – abs() ;
- 8 – divmod() ;
- 9 – pow() ;
- 10 – round() ;
- 11 – & ;
- 12 – | ;
- 13 – ^ ;
- 14 – ~ ;

- выполняет ввод двух аргументов – первый является целым десятичным числом, а второй задается колонкой «Аргумент» табл. 5:

- 1 – восьмеричное число;
- 2 – шестнадцатеричное число;
- 3 – число с плавающей точкой;

- после ввода двух аргументов – выполняет над ними арифметическую операцию, указанную колонкой «Операция»/«арифметическая» табл. 5;

- проверяет тип полученного результата. Если он имеет значение float – преобразовать результат в целое число;

- Вывести результат операции в окно среды разработки в системе счисления по основанию, указанному колонкой «Основание» табл. 5:

- выполняет ввод двух аргументов – каждый представляет собой целое число, заданное в двоичной системе и имеющее длину 8 битов;

- осуществляет над этими аргументами побитовую операцию, указанную колонкой «Операция»/«побитовая» табл. 5, выполненную с помощью соответствующего специального метода;

- вывести значения операндов и результат побитовой операции в окно среды.

Таблица 5 – Перечень индивидуальных заданий

№ п/п	Help	Аргумент	Основание	Операция
-------	------	----------	-----------	----------

09.03.03 Прикладная информатика, направленность (профиль) "Прикладная информатика в экономике"

Рабочая программа дисциплины

Дисциплина: ФТД.02 Программирование на Python

Форма обучения: очная, заочная

Разработана для приема 2021/2022, 2022/2023 учебного года

Обновлена на 2023/2024 учебный год

				Арифмитическая	Побитовая
1	1	1	16	+	&
2	2	2	8	-	
3	3	3	2	*	^
4	4	1	16	/	~
5	5	2	8	//	>>
6	6	3	2	%	<<
7	7	1	16	**	&
8	8	2	8	+	
9	9	3	2	-	^
10	10	1	16	*	~
11	11	2	8	/	>>
12	12	3	2	//	<<
13	13	1	16	%	&
14	14	2	8	**	
15	1	3	2	+	^
16	2	1	16	-	~
17	3	2	8	*	>>
18	4	3	2	/	<<
19	5	1	16	//	&
20	6	2	8	%	