
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Основы программирования Python»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Медиаискусство

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	6
4. Содержание дисциплины (модуля).....	6
5. Учебно-методическое обеспечение	8
6. Материально-техническое обеспечение	8
7. Методические и оценочные материалы	10

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Основы программирования Python»» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Медиаискусство, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Основы программирования Python» обеспечивает подготовку обучающихся к профессиональной проектной деятельности в условиях цифровой среды, основанной на понимании логики программирования, организации кода и технологических возможностей автоматизации. Освоение курса формирует фундамент для создания эффективных, масштабируемых и совместно разрабатываемых решений в различных прикладных контекстах — от визуального программирования до работы с данными и веб-интерфейсами.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Медиаискусство и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование у обучающихся системного понимания базовых принципов программирования, структуры кода и инструментов разработки на языке Python, а также развитие способности применять полученные знания в осознанной, структурированной и технологически обоснованной проектной практике.

Задачи изучения дисциплины (модуля):

- освоить базовый синтаксис языка Python, включая переменные, условные операторы, циклы, функции, списки, словари и обработку исключений;
- научиться работать с терминалом, файловой системой и организовывать структуру Python-проекта с использованием виртуального окружения;
- овладеть навыками чтения, записи и обработки данных в различных форматах, а также выполнения HTTP-запросов и парсинга веб-данных;
- научиться создавать визуальные алгоритмические композиции с помощью модуля turtle и применять принципы модульности и повторного использования кода;
- освоить базовую работу с системой контроля версий Git и платформой GitHub для версионирования, публикации и совместной разработки проектов.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- базовый синтаксис Python: переменные, условия, циклы, функции, списки, словари, исключения;
- принципы работы с терминалом и файловой системой;
- структуру Python-проекта и назначение виртуального окружения;
- форматы данных, с которыми можно работать при создании программ;
- принципы организации кода по файлам и повторного использования функций
- базовую работу с GitHub.

уметь:

- запускать Python-проект и настраивать среду;
- устанавливать библиотеки через и управлять зависимостями;
- создавать генеративную графику с помощью turtle;
- читать и обрабатывать файлы;
- выполнять простые запросы к интернету и парсить данные;
- писать аккуратный, структурированный код с разделением на файлы;
- использовать git для версионирования и GitHub для публикации;
- отлаживать код и собирать финальный проект с документацией.

владеть:

- навыками работы в терминале и редакторе кода для написания и отладки кода;
- инструментами создания виртуального окружения и управления зависимостями;
- приёмами алгоритмической генерации визуальных паттернов (turtle);
- методами сбора, хранения и преобразования данных;
- навыками организации кода и базовой работы с git/GitHub;
- способностью создавать самостоятельный или групповой проект с краткой презентацией и дальнейшим развитием.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства, для управления и демонстрации объектов медиаискусства	ОПК-3.1.	Знает основные принципы программирования, а также особенности отечественных информационных технологий и сетевых ресурсов в области медиаискусства
		ОПК-3.2.	Умеет разрабатывать прикладные программные средства, используя современные инструменты и технологии, а также интегрировать их с сетевыми ресурсами для решения конкретных задач в области медиаискусства
		ОПК-3.3.	Имеет практический опыт разработки программных средств, используемых при создании моделей и решений в области медиаискусства
ПК-4.	Способен проектировать и сопровождать системы, включающие компоненты искусственного интеллекта, с учётом этических и авторских аспектов в сфере медиаискусства	ПК-4.1.	Знает о возможностях использования технологий искусственного интеллекта в художественных и технических проектах, а также о типовых сценариях их применения в области медиаискусства
		ПК-4.2.	Умеет формулировать требования к качеству, интерпретируемости и контролируемости компонентов искусственного интеллекта в контексте медиаискусства
		ПК-4.3.	Имеет опыт ведения проектной документации, фиксирующей структуру, логику и результаты функционирования

			компонентов искусственного интеллекта, с целью воспроизводимости и сохранения авторского замысла под руководством опытных специалистов
--	--	--	--

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Основа: среда, Python, генеративная графика	6	12		18	Аудиторная работа Проект
2	Данные, файлы, парсинг	6	12		19	Аудиторная работа Проект
3	Структура проекта, ИИ, обратная связь	6	12		19	Аудиторная работа Проект
4	Расширение горизонта	6	13		19	Аудиторная работа Проект
5	Работа над финальным проектом	6	13		19	Документация проекта Проект
	Зачет с оценкой			4		
Итого:		30	62	4	94	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основа: среда, Python, генеративная графика	терминал и навигация по файловой системе; структура проекта; виртуальное окружение; pip, библиотеки, requirements.txt; VS Code, запуск кода; переменные, условия, циклы, функции, списки, словари, исключения;

		turtle как вход в генеративную графику; мини-проект: первая алгоритмическая графика на пен-плоттере.
2	Данные, файлы, парсинг	работа с файлами; JSON, CSV, таблицы; простые запросы в интернет; базовый парсинг; обсуждение хранения данных; мини-проект на основе собранных данных.
3	Структура проекта, ИИ, обратная связь	организация кода по файлам; повторное использование функций; базовые принципы аккуратного кода; работа с GitHub; проект с обратной связью (feedback-loop).
4	Расширение горизонта	speech-to-text / text-to-speech; базы данных обзорно: SQLite, MongoDB; jupyter notebook как инструмент изучения и анализа; браузерная автоматизация обзорно; контейнеры и веб-сервисы обзорно; Blender Python обзорно; начало работы над финальным проектом.
5	Работа над финальным проектом	начало работы над собственным или групповым проектом; доработка структуры и логики; отладка; зависимости и документация; подготовка краткой презентации.

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Окулич, В. И. Введение в информатику и программирование : учебное пособие / В. И. Окулич. - Нижний Новгород : НИУ РАНХиГС, 2022. - 312 с. - ISBN 978-5-00036-285-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2221620>.
2. Гусев, С. А. Математическое программирование : учебное пособие / С. А. Гусев. - Новосибирск : Изд-во НГТУ, 2023. - 134 с. - ISBN 978-5-7782-4996-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2246192>.

Дополнительная литература:

1. Апатова, Н. В. Алгоритмизация и программирование : учебное пособие / Н. В. Апатова, М. А. Бакуменко. – Москва ; Вологда : Инфра-Инженерия, 2025. - 216 с. – ISBN 978-5-9729-2228-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2225324>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья,

оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		

Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Основы программирования Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, проект, документация проекта, аудиторная работа, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Аудиторная работа – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично

фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Проект — исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Документация проекта — видео документация итогового проекта, которая включает в себя концепцию проекта, описание технических особенностей работы, методы и этапы создания, демонстрацию прототипа. Оформлена в едином стиле и опубликована в открытый доступ.

Подготовка документации позволяет студентам систематизировать результаты своей работы, отразить ключевые этапы создания проекта и продемонстрировать его сильные стороны широкой аудитории. Такой формат развивает навыки визуальной коммуникации, структурированного изложения информации и формирования профессионального портфолио.

Самостоятельная работа — работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Основы программирования Python»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине, но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет
4	Удовлетворительно	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		лишь минимальным набором методов исследования.
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.

Дисциплина (модуль) «Основы программирования Python» оценивается следующим образом:

Активность	Вес	Количество	Описание
Проект	30%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Аудиторная работа	30%	1	Активная работа студента на занятии
Документация проекта	20%	1	Видео документация итогового проекта
Зачет с оценкой	20%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Основы программирования Python»: $0,3 \times \text{Проект} + 0,2 \times \text{Документация проекта} + 0,3 \times \text{Аудиторная работа} + 0,2 \times \text{Зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные вопросы для подготовки к занятию

Тема: Основа — среда, Python, генеративная графика

1. Как с помощью командной строки перейти в нужную папку, создать проект и проверить, какие файлы в ней находятся?
2. Зачем при работе над программой создавать отдельное окружение, и как это помогает избежать путаницы с библиотеками?
3. Как в языке Питон записываются переменные, условия и повторяющиеся действия, и зачем они нужны для построения алгоритма?
4. Что представляют собой списки и словари, и в каких случаях удобнее использовать каждый из этих видов хранения данных
5. Как с помощью черепаший графики можно превратить простые команды в сложный рисунок, и как это связано с алгоритмическим мышлением?

Тема: Данные, файлы, парсинг

1. Как прочитать содержимое файла и записать в него новые сведения с помощью языка Питон?
2. В чём различие между хранением данных в виде таблицы и в виде текста с разделителями, и когда что удобнее?

3. Как получить сведения с веб-страницы, не открывая её в браузере, и какие меры нужно соблюдать при этом?
4. Как очистить и упорядочить собранные сведения, чтобы из них можно было сделать выводы или построить рисунок?
5. Как сохранить собранные данные так, чтобы ими можно было воспользоваться позже или передать другому человеку?

Тема: Структура проекта, ИИ, обратная связь

1. Как разделить программу на части, чтобы она была понятной и удобной для доработки?
2. Какие правила стоит соблюдать, чтобы код был чистым, логичным и легко читаемым?
3. Зачем сохранять ход работы над проектом и как это помогает при совместной деятельности?
4. Как программа может менять своё поведение на основе полученных откликов или новых сведений?
5. Как показать и объяснить свой проект так, чтобы другой человек понял, как он устроен и зачем нужен?

Примерное описание задания и критерии оценивания к проекту

Проект. «Голос машины: интерпретация технологического контекста»

Цель проекта:

Осмыслить одну из современных технологических тенденций (голосовые интерфейсы, автоматизация, анализ данных, 3д-генерация) и реализовать её в авторском цифровом проекте с применением расширенных инструментов Python, демонстрируя понимание их возможностей и культурного значения.

Задачи и этапы выполнения:

— Выбрать технологическую тенденцию (например: рост голосовых помощников, автоматизация браузера, анализ текста через Jupyter, генерация 3д-объектов в Blender) и провести краткий анализ её социального, технического и этического контекста.

— Определить, как эта технология влияет на повседневную жизнь, коммуникацию или творчество, и сформулировать авторскую позицию.

— Разработать концепцию небольшого цифрового проекта: аудиоистория с текст-в-речь, автоматизированный сбор данных из веба, визуализация текста в Jupyter, простая 3D-форма в Blender через скрипт.

— Реализовать проект с использованием одного или нескольких расширенных инструментов (например: pyttsx3, speech_recognition, selenium, sqlite3, pymongo, Jupyter Notebook, bry).

— Подготовить презентацию с пояснением кода, демонстрацией результата и аргументированным осмыслением выбранной технологии: как она работает, зачем нужна, какие вызовы несёт.

Критерии оценивания:

— Глубина понимания выбранной технологической тенденции и её контекста.

- Осознанность выбора и применения инструментов Python.
- Работоспособность и логичность кода, чёткая структура проекта.
- Целостность цифрового результата и его соответствие замыслу.
- Аргументированность защиты: способность объяснить, что сделано, как и зачем, включая этические и культурные аспекты.

Примеры оценивания документации проекта:

- полнота раскрытия концепции: документация чётко формулирует художественную идею, постановку проблемы и культурный или социальный контекст проекта;
- ясность описания технической реализации: подробно и доступно изложены используемые технологии, оборудование, программное обеспечение, архитектура системы и этапы сборки;
- структурированность и логичность повествования: видео имеет чёткую композицию — введение, основная часть (этапы создания, методы), демонстрация работы, заключение — и легко воспринимается зрителем;
- качество визуального и звукового оформления: соблюден единый стилистический подход, используются понятные схемы, кадры прототипа, пояснения в кадре или голос за кадром, звук и изображение не искажены.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Укажите тип данных в Python, который используется для хранения упорядоченной изменяемой последовательности элементов.	список	ОПК-3
2.	Назовите ключевое слово, с помощью которого в Python определяется функция.	def	ОПК-3
3.	Укажите оператор, который используется для проверки условия в Python.	if	ОПК-3
4.	Назовите структуру данных, в которой элементы хранятся в виде пар «ключ — значение».	словарь	ОПК-3
5.	Укажите инструмент командной строки для навигации по папкам и файлам в операционной системе.	терминал	ОПК-3
6.	Назовите файл, в котором перечисляются зависимости Python-проекта.	requirements.txt	ОПК-3
7.	Укажите модуль Python, используемый для создания графики с помощью управления «черепахой».	turtle	ОПК-3
8.	Назовите подход к организации кода, при котором программы делятся на отдельные логические файлы.	модульность	ОПК-3
9.	Укажите механизм, позволяющий обрабатывать ошибки в Python.	исключения	ОПК-3
10.	Назовите среду редактирования кода, активно используемую в Python-разработке и поддерживающую расширения.	VS Code	ОПК-3

11.	Укажите формат данных, удобный для хранения вложенной структуры в виде пар ключ-значение.	JSON	ПК-4
12.	Назовите метод получения данных из веб-ресурса с помощью HTTP-запроса в Python.	запрос	ПК-4
13.	Укажите тип файла, в котором хранятся табличные данные с разделителями-запятыми.	CSV	ПК-4
14.	Назовите систему контроля версий, используемую для совместной разработки кода.	Git	ПК-4
15.	Укажите платформу для публикации репозитория и совместной работы над проектами.	GitHub	ПК-4
16.	Назовите процесс автоматического извлечения информации со веб-страниц.	парсинг	ПК-4
17.	Укажите окружение, изолированное от основной системы, для установки библиотек Python.	виртуальное	ПК-4
18.	Назовите инструмент, позволяющий выполнять и визуализировать код по ячейкам.	Jupyter Notebook	ПК-4
19.	Укажите принцип написания кода, при котором он легко читается, понимается и поддерживается.	аккуратный код	ПК-4
20.	Назовите подход, при котором программа реагирует на свои предыдущие результаты, создавая цикл взаимодействия.	обратная связь	ПК-4