Подписан: Антонюк Светлана Владимировна
DN: C=RU, S=Красноярский край, L=Зеленогорск, T=Директор, O="МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ""ЦЕНТР ОБРАЗОВАНИЯ ""ПЕРСПЕКТИВА""", СНИЛС=03357285455, ИНН=245300923747, E=julfas@mail.ru, G=Светлана Владимировна, SN=Антонюк, CN=Антонюк Светлана Владимировна
Основание: Я являюсь автором этого документа
Местоположение: место подписания
Дата: 2026.10.29 14:36:31+07'00'
Foxit Reader Версия: 10.1.1

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«ЦЕНТР ОБРАЗОВАНИЯ «ПЕРСПЕКТИВА»

РАССМОТРЕНО
На заседании
педагогического совета
МБУ ДО «ЦО «Перспектива»
Протокол от 03.06.2026 № 3

УТВЕРЖДАЮ
Директор МБУ ДО
«ЦО «Перспектива»
_____С.В. Антонюк
Приказ от 03.06.2026 № 75

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«PYTHON: РАЗРАБОТКА ИГР И ПРИЛОЖЕНИЙ С НУЛЯ»

Возраст обучающихся: 14-18 лет
Срок реализации программы: 1 год
Уровень программы: **базовый**
Направленность программы: **техническая**
Уровень: **базовый**

Автор-составитель:
Шрамков Юрий Глебович, педагог дополнительного образования

Зеленогорск
2026

Раздел I. Комплекс основных характеристик программы

1.1 Пояснительная записка.

Дополнительная общеобразовательная общеразвивающая программа «Python: разработка игр и приложений с нуля» (далее - программа) имеет техническую направленность и разработана для детей 14-18 лет.

Программа направлена развитие у учащихся интереса к техническому образованию, формирование цифровой и технической грамотности, развитие исследовательских, прикладных, инженерных способностей обучающихся в области точных наук и технического творчества с использованием языка программирования Python. Программа разработана в соответствии со следующими нормативно-правовыми актами (выбрать необходимые для конкретной программы документы):

− Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 25.04.2026) «Об образовании в Российской Федерации»;

− Стратегия развития воспитания в Российской Федерации до 2025 года, утвержденная распоряжением Правительства РФ от 29.05.2015 г. № 996-р.;

− Концепция развития дополнительного образования детей до 2030 (Распоряжение Правительства РФ от 31.03.2022 г. № 678-р);

− Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (вступ. в силу с 01.03.2023);

− Указ Президента РФ от 07.05.2024 № 309 «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года», Паспорта национального проекта «Молодежь и дети»;

− Письмо Министерства образования и науки Российской Федерации от 29.03.2016 г. № ВК-641/09 «Методические рекомендации по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей»;

− Письмо Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242 «О направлении методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы);

− Письмо Министерства просвещения Российской Федерации от 31.01.2022 № ДГ-245/06 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями по реализации дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий»);

− Письмо Министерства образования и науки Российской Федерации от 28.08.2015 г. № АК-2563/05 «О методических рекомендациях» (вместе с «Методическими рекомендациями по организации образовательной

деятельности с использованием сетевых форм реализации образовательных программ»);
− Постановление Главного государственного санитарного врача Российской Федерации от 02.06.2026 г. № 19 «Об утверждении СанПиН 2.4.24283-26 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
- Лицензии на осуществление образовательной деятельности;
- Устава МБУ ДО «ЦО «Перспектива»;
- Положения об организации деятельности по дополнительным общеобразовательным программам, и с учетом возрастных и индивидуальных особенностей обучающихся на занятиях технической направленности и спецификой работы учреждения.

Актуальность программы обусловлена потребностью общества в технически грамотных специалистах в области разработки программного обеспечения и растущей популярностью языка программирования Python как универсального инструмента для решения широкого спектра задач в IT-индустрии.

Программа нацелена на решение задач, определенных в Стратегии развития воспитания в Российской Федерации на период до 2025 года, что подчеркивает ее значимость в контексте трудового воспитания и профессионального самоопределения.

Python является одним из самых востребованных языков программирования в мире, широко применяется в разработке игр, веб-приложений, научных вычислениях, машинном обучении, анализе данных и автоматизации. Изучение Python открывает обучающимся путь к множеству современных профессий: разработчик игр, веб-разработчик, специалист по анализу данных, разработчик десктопных приложений.

Программа способствует формированию цифровой и технической грамотности, что является критически важным навыком в современном мире. Позволяет учащимся получить представление о значимости современных информационных технологий, раскрывая особенности новых профессий в IT-области.

Наконец, программа разработана с учетом модульного построения содержания, что обеспечивает индивидуальный образовательный маршрут для каждого учащегося.

Новизна дополнительной общеразвивающей программы «Python: разработка игр и приложений с нуля» заключается в том, что она обеспечивает теоретическое и практическое овладение современными информационными технологиями с использованием языка Python - одного из наиболее доступных и мощных инструментов программирования.

В ходе занятий по программе обучающиеся погружаются в программирование на языке Python с нуля, осваивают различные направления разработки, понимают их особенности и возможности,

выявляют способы применения и определяют наиболее интересные направления для дальнейшего углубления.

Обучающиеся осваивают:

- Основы Python: синтаксис, структуры данных, объектно-ориентированное программирование.
- Разработку 2D-игр: Pygame для создания полноценных игр с графикой и звуком.
- 3D-моделирование в Blender: с использованием Python API для автоматизации.
- Веб-разработку: создание веб-приложений с Flask/Django.
- Разработку GUI-приложений: создание десктопных приложений с Tkinter/PyQt.
- Работу с данными и визуализацией: (NumPy, Pandas, Matplotlib).
- Основы работы с базами данных: (SQLite, PostgreSQL)и процессов, разработки приложений и других смежных областях.

Параллельно с этим, обучающиеся развивают навыки дизайн-мышления, дизайн-анализа и способность создавать новое и востребованное. Синергия методов и технологий, используемых в обучении, дает обучающимся уникальные метапредметные компетенции, полезные в сфере проектирования, моделирования объектов и процессов, разработки приложений и других смежных областях.

Отличительные особенности дополнительной общеразвивающей программы «Python: разработка игр и приложений с нуля» заключаются в следующем:

1. Использование Python как основного языка программирования: Python является одним из самых популярных и доступных языков, что снижает порог входа для начинающих и одновременно открывает широкие возможности для профессионального роста.
2. Техническая направленность: Программа ориентирована на изучение практических технологий разработки программного обеспечения, что делает ее актуальной в контексте современного технологического развития.
3. Междисциплинарность: Программа охватывает различные аспекты технического творчества, включая программирование, разработку игр, веб-разработку, создание десктопных приложений, 3D-моделирование в Blender и работу с данными.
4. Практико-ориентированный подход: Обучение включает в себя выполнение практических заданий и проектов, что позволяет учащимся применять полученные знания и навыки в реальных условиях.
5. Использование современных технологий и инструментов: Программа предполагает работу с современным программным обеспечением (Visual Studio Code, PyCharm, Blender, Pygame, Flask/Django, Tkinter/PyQt), что обеспечивает актуальность и востребованность полученных знаний и навыков.

6. Развитие творческих способностей: Программа способствует развитию творческого мышления, умения решать нестандартные задачи и создавать новые идеи.
7. Формирование командных навыков: Работа над проектами и заданиями в группах способствует развитию коммуникативных навыков, умения работать в команде и распределять роли.
8. Индивидуальный подход: Программа предусматривает возможность индивидуального подхода к обучению, учитывая интересы и способности каждого учащегося.
9. Интеграция с экосистемой Python: Обучающиеся получают доступ к огромному количеству библиотек и фреймворков Python, что позволяет расширять функциональность своих проектов (машинное обучение, анализ данных, веб-разработка, автоматизация).

Модуль - структурная единица образовательной программы, имеющая определённую логическую завершённость по отношению к результатам обучения. (Словарь рабочих терминов по предпрофильной подготовке).

1. Результатом освоения вводного блока является формирование фундаментальных hard и soft skills, необходимых для дальнейшего обучения. Слушатель осваивает основы алгоритмического мышления, получает первичные навыки настройки современного программного обеспечения, а также учится работать в команде, используя облачные сервисы для совместной работы.

Soft Skills: Навыки командной работы, коммуникации, решения практических задач, поиска и анализа информации.

Hard Skills: Базовое понимание логики работы ПК, установка и настройка необходимого ПО (Python, Visual Studio Code/PyCharm, Blender, Pygame), основы работы с облачными сервисами, основы алгоритмического мышления.

2. Результатом освоения Блоков 2 и (Основы Python и Разработка игр и приложений) является создание серии учебных продуктов (кейсов), демонстрирующих сформированность технических компетенций. Каждый кейс - это законченный учебный модуль (например, консольное приложение, 2D-игра, веб-приложение, десктопное приложение с GUI), который представляет собой решение конкретной учебной задачи и служит основой для портфолио.

Кейс по Python: Написание работающих программ на Python, демонстрирующих понимание синтаксиса, ООП и работы с данными.

Кейс по Pygame: Создание 2D-игры с графикой, анимацией и игровой логикой.

Кейс по веб-разработке: Создание веб-приложения с использованием Flask или Django.

Кейс по GUI: Создание десктопного приложения с графическим интерфейсом.

- Результатом освоения проектного блока является законченный проект (индивидуальный или командный), представленный к защите. Проект представляет собой комплексное решение, интегрирующее все полученные знания и навыки: от первичного анализа идеи и планирования до технической реализации на Python, с последующим тестированием и финальной презентацией
- Результат: Функционирующее приложение или игра на Python/
- Защита проекта: Публичная презентация, включающая демонстрацию рабочего продукта, обоснование принятых решений и ответы на вопросы экспертов.

Адресат программы

Программа предназначена для подростков в возрасте 14–18 лет, мотивированных к обучению и обладающих системным мышлением. Формы занятий групповые, количество обучающихся в группе – 10 человек. Состав групп постоянный, разновозрастной.

Место проведения занятий: г. Зеленогорск, ул. Комсомольская д. 17

Возрастные особенности группы

Содержание программы учитывает возрастные и психологические особенности детей 14–18 лет, которые определяют выбор форм проведения занятий с обучающимися. Дети этого возраста отличаются внутренней уравновешенностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия.

Ребят также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценки поступков и действий ребёнка со стороны не только старших, но и сверстников. Ребёнок стремится завоевать в их глазах авторитет, занять достойное место в коллективе. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти своё место в детском коллективе.

Также следует отметить, что дети данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало формированию начального профессионального самоопределения обучающихся.

Режим занятий, объём программы: длительность одного занятия составляет 2 академических часа по 45 минут с десятиминутным перерывом, периодичность занятий - 2 раза в неделю.

Срок освоения программы определяется содержанием программы и составляет 1 года. Годовая нагрузка на ученика составляет 144 часа.

Объём программы: 144 часов.

По уровню освоения программа ***одноуровневая:*** базовый уровень - 1 год обучения.

Формы обучения: очная, очно-заочная.

Виды занятий: беседы, обсуждения, мультимедийные презентации, игровые формы работы, практические занятия, метод проектов. Также программа курса включает групповые и индивидуальные формы работы обучающихся (в зависимости от темы занятия).

Основной тип занятий - комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится педагогом самостоятельно, но с учётом рекомендованного календарно-тематического плана.

Каждая тема курса начинается с постановки задачи – характеристики предметной области или конкретной программы на языке C#, которую предстоит изучить. С этой целью преподаватель проводит презентацию или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в группе. Задания выполняются на компьютере с использованием интегрированной среды разработки (Visual Studio Code, PyCharm или Jupyter Notebook). При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того, чтобы каждый обучающийся получил наилучший результат, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес обучающегося к предмету, его активность и самостоятельность, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики обучения индивидуального и в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний обучающихся. Выполнение контрольных

заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

Реализация программы предполагает использование здоровье сберегающих технологий.

Здоровьесберегающая деятельность реализуется:

- Через создание безопасных материально-технических условий.
- Включением в занятие динамических пауз, периодической смены деятельности обучающихся.
- Контролем соблюдения обучающимися правил работы на ПК.
- Через создание благоприятного психологического климата в учебной группе в целом.

Педагогическая целесообразность программы

Программа является целостной и непрерывной в течение всего процесса обучения и позволяет обучающемуся шаг за шагом раскрывать в себе творческие возможности и самореализоваться в современном мире.

Стартовый уровень предполагает использование и реализацию общедоступных и универсальных форм организации материала, минимальную сложность предлагаемого для освоения содержания программы.

Базовый уровень помимо использования и реализации общедоступных и универсальных форм организации, пройденного на стартовом уровне, материала, предполагает уже более высокий уровень сложности материала для освоения содержания программы второго года обучения.

Осваивая данную программу, обучающиеся будут овладевать навыками уже востребованных специальностей, многие из которых включены в Атлас профессий будущего. Знания и навыки, рассматриваемые в программе, будут полезны для каждой перспективной профессии.

Дополнительная общеобразовательная общеразвивающая программа «Python: разработка игр и приложений с нуля» имеет техническую направленность и ориентирована на изучение программирования на языке Python, разработку игр, веб-приложений и десктопных приложений.

В ходе обучения дети получат навыки командного взаимодействия, «hard» и «soft» компетенций, а также получат знания в области программирования на Python, работы с игровыми библиотеками (Pygame), веб-фреймворками (Flask/Django), создания GUI-приложений (Tkinter/PyQt), работы с данными (NumPy, Pandas, Matplotlib) и компьютерных технологий.

Обучающиеся после окончания курса получат начальные знания об истории развития отечественной и мировой техники, о различных направлениях изучения программирования, веб-разработки, анализа данных, компьютерных технологий; освоят принципы работы с Python и его экосистемой, а также приемы и технологии разработки игр, веб-приложений, десктопных приложений и автоматизации процессов.

Программа рассчитана на обучающихся 14–18 лет.

Цели и задачи программы

Цель программы: Формирование у обучающихся комплексных hard skills в области разработки программного обеспечения с использованием языка программирования Python и soft skills, необходимых для успешной работы в современной IT-индустрии. Программа направлена на подготовку специалистов, способных самостоятельно создавать и реализовывать проекты - от идеи до работающего прототипа.

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

Обучающие задачи:

- Сформировать устойчивые знания базового синтаксиса и принципов объектно-ориентированного программирования на языке Python.
- Обучить работе в средах разработки Visual Studio Code и PyCharm, включая настройку интерфейса, отладку и использование системы контроля версий (Git).
- Дать практические навыки создания 2D-игр с использованием библиотеки Pygame: работа со спрайтами, анимацией, звуком и игровой логикой.
- Сформировать навыки работы с Blender для 3D-моделирования и использования Python API Blender для автоматизации процессов.
- Обучить основам веб-разработки с использованием фреймворков Flask или Django: создание веб-приложений, работа с шаблонами, формами и базами данных.
- Сформировать навыки создания десктопных приложений с графическим интерфейсом с использованием Tkinter или PyQt.
- Познакомить с основами работы с данными: библиотеки NumPy, Pandas, визуализация данных с Matplotlib.
- Обучить работе с базами данных (SQLite, PostgreSQL) и основам SQL.
- Освоить основы проектной деятельности и гибких методологий (Agile/Scrum) для эффективного планирования и выполнения групповых проектов.

Развивающие задачи:

- Развить алгоритмическое и логическое мышление, необходимое для решения задач программирования.
- Развить навыки поиска, анализа и применения информации для самостоятельного решения технических задач.
- Сформировать навыки командной работы, распределения задач и совместной разработки с использованием облачных сервисов (Trello, Notion, GitHub).
- Развить навыки презентации и защиты своих проектов, аргументации выбранных решений.

- Стимулировать интерес к самостоятельному изучению новых инструментов и технологий в сфере разработки программного обеспечения и Python-разработки.

Воспитательные задачи:

- Воспитать этику групповой работы, отношения делового сотрудничества и взаимоуважения в процессе командной разработки.
- Сформировать ответственное отношение к труду, целеустремленность и упорство в достижении результата при решении сложных технических задач.
- Воспитать понимание важности качества кода и пользовательского опыта (UX) для создания комфортных и этичных программных продуктов.
- Сформировать активную жизненную позицию через реализацию собственных проектов и решение практических задач.
- Пропагандировать здоровый образ жизни и формировать понимание правил безопасности при работе с компьютерной техникой.

3. Содержание программы
Учебно-тематический план обучения

№ п/п	Название блока, темы	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Блок 1. Вводный		**25**	**6**	**19**	
1.1	**Введение. Командообразование. Правила безопасности**	**2**	**1**	**1**	Опрос.
1.2	**Основы работы с ПК и облачными сервисами**	**11**	**2**	9	Практические задачи
1.2.1	Настройка ПК под ПО. Установка необходимого софта	6	1	5	
1.2.2	Работа с облачными сервисами.	5	1	4	
1.3	**Основы алгоритмического мышления**	**12**	**4**	**8**	Решение задач
Блок 2. Основы программирования на Python		**42**	**10**	**32**	
2.1	**Знакомство со средой разработки (VS Code/PyCharm)**	**4**	**1**	**3**	Опрос
2.2	**Базовый синтаксис Python (переменные, типы данных, операторы, условия, циклы)**	**12**	**3**	**9**	
2.3	**Работа с данными (списки, словари, кортежи, множества, файлы, исключения)**	**14**	**4**	**10**	
2.2.3	**Работа с данными (массивы, списки, исключения, файлы)**	**12**	**2**	**10**	

Блок 3. Разработка игр и приложений на Python		**53**	**14**	**39**	
3.1	**Разработка 2D-игр с Pygame**	**17**	**5**	**12**	Презентация полигонов
3.1.1	Введение в Pygame. Создание окна, игровой цикл	6	2	4	
3.1.2	Работа со спрайтами, анимацией и звуком	6	2	4	
3.1.3	Обработка событий, коллизии, игровая логика	5	1	4	
3.2	**Веб-разработка с Flask/Django**	**18**	**4**	**14**	Презентация рабочих программ
3.2.1	Основы веб-разработки. Введение в Flask	6	2	4	
3.2.2	Работа с шаблонами, формами и базами данных	6	1	5	
3.2.3	Создание полноценного веб-приложения	6	1	5	
3.3	**Разработка GUI-приложений и работа с данными**	**18**	**5**	**13**	Презентация приложения
3.3.1	Создание GUI с Tkinter/PyQt	4	2	2	
3.3.2	Работа с данными: NumPy, Pandas, Matplotlib	8	2	6	
3.3.3	3D-моделирование в Blender с Python API	6	1	5	
Блок 4. Проектный		**24**	**5**	**19**	
4.1	Этап 1. Постановка проблемы	4	2	2	Презентация и защита итогового проекта
4.2	Этап 2. Концептуальный	2	1	1	
4.3	Этап 3. Планирование	4	2	2	
4.4	Этап 4. Аналитическая часть	2	-	2	
4.5	Этап 5. Техническая и технологическая проработка	10	-	10	
4.6	Этап 6. Тестирование и защита проектов	2	-	2	
Итого		**144**	**35**	**109**	

Содержание учебного плана

Блок 1. Вводный

Тема 1.1. Введение. Командообразование. Правила безопасности (2 часа: теория - 1, практика - 1)

Теория: Знакомство с программой курса, правилами работы и техникой безопасности. Введение в основы командной динамики. Обзор языка Python и его применения в современной индустрии: разработка игр, веб-приложений, анализ данных, автоматизация.

Практика: Игры и упражнения на командообразование. Инструктаж по безопасной работе с компьютерной техникой.

Форма контроля: Опрос по технике безопасности.

Тема 1.2. Основы работы с ПК и облачными сервисами (11 часов: теория - 2, практика - 9)

Тема 1.2.1. Настройка ПК под ПО. Установка необходимого софта (6 часов: теория - 1, практика - 5)

Теория: Требования к аппаратному обеспечению для разработки на Python. Обзор необходимого ПО: Python (последняя стабильная версия), Visual Studio Code/PyCharm, Blender, Pygame, Git.

Практика: Установка и настройка всего программного стека. Настройка виртуального окружения Python (venv). Установка необходимых библиотек через pip.

Тема 1.2.2. Работа с облачными сервисами (5 часов: теория - 1, практика - 4)

Теория: Обзор современных облачных сервисов для collaboration (GitHub/GitLab, Trello, Notion, Google Docs). Принципы эффективного поиска информации в интернете. Работа с документацией Python.

Практика: Создание репозитория для проекта на GitHub. Организация рабочего пространства в Trello/Notion. Решение кейсов по поиску информации в документации Python.

Форма контроля: Практические задачи по настройке и организации.

Тема 1.3. Основы алгоритмического мышления (12 часов: теория - 4, практика - 8)

Теория: Понятие алгоритма. Базовые структуры данных и алгоритмы (линейные, ветвления, циклы). Принципы декомпозиции задачи. Блок-схемы и псевдокод.

Практика: Решение задач на визуальных языках программирования (например, Blockly) или написание псевдокода для типовых ситуаций (управление персонажем, подсчет очков). Первые простые программы на Python.

Форма контроля: Решение задач.

Блок 2. Основы программирования на Python

Тема 2.1. Знакомство со средой разработки (Visual Studio Code/PyCharm) (4 часа: теория - 1, практика - 3)
Теория: Обзор интерфейса IDE: панели инструментов, проводник файлов, окно кода, терминал, отладчик. Настройка расширений для Python. Работа с виртуальными окружениями.
Практика: Создание первого Python-скрипта "Hello, World!". Настройка окружения. Использование базовых функций автодополнения кода (IntelliSense). Запуск и отладка программ.
Форма контроля: Опрос.
Тема 2.2. Базовый синтаксис Python (12 часов: теория - 3, практика - 9)
Теория: Переменные и типы данных (int, float, str, bool). Операторы (арифметические, логические, сравнения). Условные конструкции (if-elif-else). Циклы (for, while). Функции и их параметры.
Практика: Написание консольных программ: калькулятор, конвертер величин, простые игры («Камень-ножницы-бумага»).
Форма контроля: Практические задания.
Тема 2.3. Принципы ООП в Python (14 часов: теория - 4, практика - 10)
Теория: Понятие класса и объекта. Инкапсуляция, наследование, полиморфизм. Конструкторы. Модификаторы доступа (public, private).
Практика: Создание иерархии классов для игры (например, Entity -> Character -> Player/Enemy).
Тема 2.4. Работа с данными (12 часов: теория - 2, практика - 10)
Теория: Списки (list), кортежи (tuple), словари (dict), множества (set). Обработка исключений (try-except-finally). Работа с файлами (чтение/запись текстовых и JSON-файлов). Модули и импорт.
Практика: Создание простой системы сохранения прогресса игры в JSON-файл (например, таблица рекордов). Работа с данными игрока.
Форма контроля: Практические задания.

Блок 3. Разработка игр и приложений на Python

Тема 3.1. Разработка 2D-игр с Pygame (17 часов: теория - 5, практика - 12)
Тема 3.1.1. Введение в Pygame. Создание окна, игровой цикл (6 часов: теория - 2, практика - 4)
Теория: Установка и настройка Pygame. Структура игрового приложения. Игровой цикл (game loop). Обработка событий. Отрисовка примитивов.
Практика: Создание окна игры. Реализация игрового цикла. Отрисовка простых фигур и текста.
Тема 3.1.2. Работа со спрайтами, анимацией и звуком (6 часов: теория - 2, практика - 4)
Теория: Загрузка и отображение изображений. Работа со спрайтами (класс `Sprite`). Анимация спрайтов. Загрузка и воспроизведение звуков и музыки.

Практика: Создание анимированного персонажа. Добавление фоновой музыки и звуковых эффектов.

Тема 3.1.3. Обработка событий, коллизии, игровая логика (5 часов: теория - 1, практика - 4)

Теория: Обработка ввода с клавиатуры и мыши. Определение столкновений (collision detection). Реализация игровой логики (счет, жизни, уровни).

Практика: Создание полноценной 2D-игры (например, «Змейка», «Платформер»).

Форма контроля: Презентация готовой 2D-игры.

Тема 3.2. Веб-разработка с Flask/Django (18 часов: теория - 5, практика - 13)

Тема 3.2.1. Основы веб-разработки. Введение в Flask (6 часов: теория - 2, практика - 4)

Теория: Основы веб-технологий (HTTP, HTML, CSS). Установка и настройка Flask. Создание первого веб-приложения. Маршрутизация (routing).

Практика: Создание простого веб-приложения с несколькими страницами. Работа с маршрутами и представлениями.

Тема 3.2.2. Работа с шаблонами, формами и базами данных (6 часов: теория - 2, практика - 4)

Теория: Шаблонизатор Jinja2. Работа с формами (Flask-WTF). Основы работы с базами данных (SQLite). ORM SQLAlchemy.

Практика: Создание веб-приложения с формами регистрации и авторизации. Сохранение данных в базу данных.

Тема 3.2.3. Создание полноценного веб-приложения (6 часов: теория - 1, практика - 5)

Теория: Архитектура веб-приложений. Работа с сессиями и cookies. Деплой приложения.

Практика: Создание полноценного веб-приложения (например, блог, список задач, интернет-магазин).

Форма контроля: Презентация готового веб-приложения.

Тема 3.3. Разработка GUI-приложений и работа с данными (18 часов: теория - 4, практика - 14)

Тема 3.3.1. Создание GUI с Tkinter/PyQt (8 часов: теория - 2, практика - 6)

Теория: Введение в GUI-программирование. Основы Tkinter: виджеты, компоновка, обработка событий. Альтернатива: PyQt5.

Практика: Создание десктопного приложения с графическим интерфейсом (например, калькулятор, текстовый редактор, менеджер задач).

Тема 3.3.2. Работа с данными: NumPy, Pandas, Matplotlib (6 часов: теория - 1, практика - 5)

Теория: Введение в научные вычисления. Библиотека NumPy для работы с массивами. Pandas для анализа данных. Визуализация данных с Matplotlib.

Практика: Анализ и визуализация данных из CSV-файлов. Создание графиков и диаграмм.

Тема 3.3.3. 3D-моделирование в Blender с Python API (4 часа: теория - 1, практика - 3)
Теория: Введение в Blender. Python API Blender (модуль `bpy`). Автоматизация задач моделирования.
Практика: Написание скриптов для автоматического создания 3D-объектов и сцен.
Форма контроля: Презентация готового приложения.

Блок 4. Проектный

Тема 4.1. Этап 1. Постановка проблемы (4 часа: теория - 2, практика - 2)
Теория: Методы анализа проблем и формирования идеи проекта.
Практика*:* Мозговой штурм, выбор и формулировка темы итогового проекта.
Тема 4.2. Этап 2. Концептуальный (2 часа: теория - 1, практика - 1)
Теория: Принципы целеполагания (SMART).
Практика: Определение целей, задач и финального вида продукта проекта.
Тема 4.3. Этап 3. Планирование (4 часа: теория - 2, практика - 2)
Теория: Основы Agile/Scrum. Создание бэклога продукта. Распределение ролей в команде.
Практика: Разделение проекта на спринты, планирование задач в Trello/Notion.
Тема 4.4. Этап 4. Аналитическая часть (2 часа: практика - 2)
Практика: Анализ аналогов, существующих решений и технологий.
Тема 4.5. Этап 5. Техническая и технологическая проработка (10 часа: практика - 10)
Практика: Активная разработка проекта: программирование на Python, создание интерфейсов, работа с базами данных, отладка.
Тема 4.6. Этап 6. Тестирование и защита проектов (2 часа: практика - 2)
Практика: Финальное тестирование, подготовка презентации, выступление перед комиссией.
Форма контроля: Презентация и защита итогового проекта.

1.4. Планируемые результаты

Предметные результаты

Выпускник будет знать:

- Базовый синтаксис и основные принципы объектно-ориентированного программирования (ООП) на языке Python.
- Структуру и логику работы в средах разработки Visual Studio Code и PyCharm.
- Основы работы с библиотекой Pygame для создания 2D-игр.
- Основы веб-разработки с использованием фреймворков Flask или Django.
- Принципы создания десктопных приложений с графическим интерфейсом (Tkinter/PyQt).

- Основы работы с данными: библиотеки NumPy, Pandas, визуализация с Matplotlib.
- Основы работы с базами данных (SQLite, PostgreSQL) и SQL.
- Основы проектной деятельности и гибких методологий (Agile/Scrum).

Выпускник будет уметь:

- Писать программы на Python, используя базовые конструкции языка и принципы ООП.
- Создавать 2D-игры с использованием Pygame: работать со спрайтами, анимацией, звуком и игровой логикой.
- Создавать веб-приложения с использованием Flask/Django: работать с шаблонами, формами, базами данных.
- Создавать десктопные приложения с графическим интерфейсом с использованием kinter/PyQt.
- Работать с данными: анализировать, обрабатывать и визуализировать данные с помощью NumPy, Pandas, Matplotlib.
- Работать с базами данных: создавать таблицы, выполнять запросы SQL, использовать ORM.
- Использовать систему контроля версий Git и облачные сервисы для совместной работы.
- Планировать этапы разработки проекта, распределять задачи в команде и презентовать готовый продукт.

Личностные результаты

- Формирование устойчивого интереса к программированию на Python и разработке программного обеспечения как к потенциальной области профессиональной деятельности.
- Развитие абстрактно-логического мышления, необходимого для решения задач программирования.
- Повышение самооценки и уверенности в своих силах благодаря созданию работающих программ и завершенных проектов.
- Формирование коммуникативной компетентности и способности к эффективному сотрудничеству со сверстниками в процессе образовательной и проектной деятельности.
- Воспитание ответственного отношения к труду, целеустремленности и упорства в отладке кода и достижении технического результата.
- Формирование осознанного и уважительного отношения к мнению и результатам труда других членов команды.
- Усвоение правил безопасного поведения при работе с компьютерной техникой для сохранения собственного здоровья.

Метапредметные результаты

- Умение самостоятельно находить, анализировать и применять информацию из документации Python, форумов (Stack Overflow) и видеоуроков для решения возникающих технических проблем.

- Способность работать по предложенным инструкциям, а также планировать свою деятельность и действовать самостоятельно для достижения цели проекта.
- Умение ясно излагать свои мысли, аргументировать выбранные решения при обсуждении проектов и давать конструктивную обратную связь.
- Развитие навыков командной работы: эффективно распределять роли и обязанности в рамках проекта, нести ответственность за свой участок работы.
- Способность критически оценивать как собственный проект, так и работы других обучающихся, выделяя сильные стороны и зоны для роста.
- Умение представить публике результаты своего труда: продемонстрировать рабочее приложение, объяснить его идею, обосновать использованные технологии и ответить на вопросы.

II. Комплекс организационно-педагогических условий реализации общеразвивающей программы

2.1. Календарный учебный график на 2026-2027 учебный год

№ п/п	Основные характеристики образовательного процесса	Кол-во
1.	Количество учебных недель	36
2.	Количество учебных дней	72
3.	Количество часов в неделю	4
4.	Количество часов	144
5.	Недель в I полугодии	16
6.	Недель во II полугодии	20
7.	Начало занятий	13 сентября
8.	Выходные дни	31 декабря – 10 января
9.	Окончание учебного года	31 мая

2.2. Условия реализации программы

Материально-техническое обеспечение

Требования к помещению:

- Помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования.
- Качественное освещение.
- Столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

Оборудование:

- Компьютеры и ноутбуки на каждого обучающегося и преподавателя.
- Проекционное оборудование (экраны) – 1 шт.
- Маркерная доска – 1 шт.
- Персональные компьютеры для работы с предустановленной операционной системой и специализированным ПО.

Информационное обеспечение:

- Python (версия 3.10 или выше).
- Visual Studio Code или PyCharm Community Edition.
- Blender (последняя стабильная версия).
- Pygame (устанавливается через pip).
- Flask или Django (устанавливается через pip).
- Git для контроля версий.
- Библиотеки Python: NumPy, Pandas, Matplotlib, Tkinter (встроена), PyQt5 (опционально).
- SQLite (встроена в Python), PostgreSQL (опционально).

Методическое обеспечение

Методические пособия, разработанные преподавателем с учётом конкретных задач, упражнения, варианты демонстрационных программ, материалы по терминологии Python, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Кадровое обеспечение

Программа реализуется Воробьевой М.С., педагогом дополнительного образования.

2.3. Формы аттестации и оценочные материалы

Система отслеживания результатов обучающихся выстроена следующим образом:

- Определение начального уровня знаний, умений и навыков.
- Промежуточный контроль.
- Итоговый контроль.

Входного контроля при приёме по данной общеразвивающей программе не предусмотрено.

Текущий контроль осуществляется путём наблюдения (Приложение 1), определения качества выполнения заданий (Приложение 2, 3), отслеживания динамики развития обучающегося. Способы проверки уровня освоения тем: опрос, тестирование, выполнение упражнений, наблюдение, оценка выполненных самостоятельных работ. Формы проведения итогов по каждой

теме и каждому разделу общеразвивающей программы соответствуют целям и задачам ДООП.

Итоговая аттестация осуществляется в форме презентации, защиты проекта и оценивается по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 4:

Таблица 4

Баллы, набранные обучающимся	Уровень освоения
0–50 баллов	Низкий
51–70 баллов	Средний
71–100 баллов	Высокий

Индивидуальный/групповой проект оценивается формируемой комиссией или принимаются призовые места на региональных/ всероссийских проектных конкурсах.

Компонентами оценки индивидуального/группового проекта являются (по мере убывания значимости): качество проекта, отзыв руководителя проекта, уровень презентации и защиты проекта. Если проект выполнен группой обучающихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально. Для оценки проекта членам комиссии рекомендуется использовать «Бланк оценки проекта» (Приложение 5).

2.4. Методические материалы

Образовательный процесс осуществляется в очной форме.

В образовательном процессе используются следующие ***методы:***

1. Объяснительно-иллюстративный.
2. Метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой).
3. Метод проектов.
4. Наглядный:

- Демонстрация презентаций, схем, таблиц, диаграмм т. п.
- Использование технических средств.
- Просмотр обучающих видеороликов (RUTUBE).

5. Практический:

- Практические задания;
- Анализ и решение проблемных ситуаций т. д.
- Написание кода на Python.
- Создание игр и приложений.

Дополнительные методы:

- «Вытягивающая модель» обучения.
- ТРИЗ/ПРИЗ.
- SWOT-анализ.
- Data Scouting.
- Кейс-метод.
- Метод Scrum, eduScrum.
- Метод «Фокальных объектов».
- Метод «Дизайн-мышление», «критическое мышление».

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания программы, степени сложности материала, типа учебного занятия. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Формы обучения:

- ***Фронтальная*** **–** предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран. Активно используются Интернет-ресурсы.
- ***Групповая*** – предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом.
- ***Индивидуальная*** – подразумевает взаимодействие преподавателя с одним обучающимся. Как правило данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится

фронтально, затем обучающийся выполняют индивидуальные задания или общие задания в индивидуальном темпе.

- ***Дистанционная*** – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающегося при самостоятельной работе дома. Налаженная система сетевого взаимодействия подростка и педагога, позволяет не ограничивать процесс обучения нахождением в учебной аудитории, обеспечить возможность непрерывного обучения в том числе, для часто болеющих детей или всех детей в период сезонных карантинов (например, по гриппу) и температурных ограничениях посещения занятий.

Формы организации учебного занятия

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения детьми образовательной программы, в соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля: беседа, лекция, мастеркласс, практическое занятие, защита проектов, конкурс, викторина, диспут, круглый стол, «мозговой штурм», воркшоп, глоссирование, деловая игра, квиз, экскурсия.

Некоторые формы проведения занятий могут объединять несколько учебных групп или весь состав объединения, например, экскурсия, викторина, конкурс и т. д.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Педагогические технологии: индивидуализации обучения; группового обучения; коллективного взаимообучения; дифференцированного обучения; разноуровневого обучения; проблемного обучения; развивающего обучения; дистанционного обучения; игровой деятельности; коммуникативная технология обучения; коллективной творческой деятельности; решения изобретательских задач; здоровьесберегающая технология.

Дидактические материалы:

Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ на Python, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

2.5.Список литературы

Список литературы, использованной при написании программы:

1. Mark Lutz / Learning Python, 5th Edition // O'Reilly Media, 2013. – 1648 pp.
2. Al Sweigart / Automate the Boring Stuff with Python, 2nd Edition // No Starch Press, 2019. – 592 pp.
3. Eric Matthes / Python Crash Course, 3rd Edition // No Starch Press, 2023. – 552 pp.
4. Al Sweigart/ Making Games with Python & Pygame // 2012. – 365 pp.
5. Miguel Grinberg / Flask Web Development, 2nd Edition // O'Reilly Media, 2018. – 316 pp.
6. William S. Vincent / Django for Beginners, 4th Edition // 2022. – 332 pp.
7. Wes McKinney / Python for Data Analysis, 3rd Edition // O'Reilly Media, 2022. – 579 pp.
8. Blender Foundation / Blender Python API Documentation // docs.blender.org
9. Выготский Л. С. Собрание сочинений в 6-ти томах М.: Педагогика, 1982-1984.
10. Зимняя И. А. «Педагогическая психология» Учебник для вузов. Изд. второе, доп., испр. и перераб. – М.: Издательская корпорация «Логос», 2000. – 384 с.
11. Исаев Е. И., Слободчиков В. И. «Психология образования человека. Становление субъективности в образовательных процессах». Учебное пособие. – Изд-во ПСТГУ, 2013.

Электронные ресурсы:

1. Python Official Documentation [Электронный ресурс] // URL: https://docs.python.org/3/ (дата обращения: 10.11.2025).
2. Pygame Documentation [Электронный ресурс] // URL: https://www.pygame.org/docs/ (дата обращения: 10.11.2025).
3. Flask Documentation [Электронный ресурс] // URL: https://flask.palletsprojects.com/ (дата обращения: 10.11.2025).
4. Django Documentation [Электронный ресурс] // URL: https://docs.djangoproject.com/ (дата обращения: 10.11.2025).

5. Real Python Tutorials [Электронный ресурс] // URL: https://realpython.com/ (дата обращения: 10.11.2025).
6. Pandas Documentation [Электронный ресурс] // URL: https://pandas.pydata.org/docs/ (дата обращения: 10.11.2025).
7. Matplotlib Documentation [Электронный ресурс] // URL: https://matplotlib.org/stable/contents.html (дата обращения: 10.11.2025).
8. Stack Overflow [Электронный ресурс] // URL: https://stackoverflow.com/questions/tagged/python (дата обращения: 10.11.2025).

Приложение 1

Бланк наблюдения за обучающимися

Группа_____ Педагог _____

№ п/п	ФИО	П О К А З А Т Е Л И					
		Внимателен в течение занятия	Использует базовую систему понятий	Проявляет инициативу, интерес в течение занятия	Идет на деловое сотрудничество	Аккуратно относится к материально-техническим	Р Е З У Л Ь Т А Т
1							
2							
3							
4							
5							

6							
7							
8							
9							
10							

За каждое согласие с утверждением 1 – балл.

Приложение 2

Лист оценки работы обучающихся в **процессе разработки проектного решения**

№ группы: _____ Дата: _____

№ п/п	Фамилия, имя обучающегося	Сложность языка разработки (по шкале от 0 до 5 баллов)	Соответствие написания программного кода поставленной задаче (по шкале от 0 до 5 баллов)	Презентация модели по плану. Степень владения специальным и терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						

9.						
10.						

Приложение 3

Лист оценки работы обучающихся в процессе настройки коммутационного и компьютерного оборудования

№ группы: _____ Дата: _____

№ п/п	ФИО обучающегося	Соответствие построенной конструкции и заданной модели (по шкале от 0 до 5 баллов)	Соответствие написанной программы заданным целям (по шкале от 0 до 5 баллов)	Степень владения специальными терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						

Приложение 4

Пример контрольного тестирования

БАЗОВЫЕ ЗНАНИЯ О PYTHON

Что выведет следующий код?

```
x = 5
y = "10"
print(x + int(y))
```

1. 15
2. 510
3. Ошибка
4. 5

Какой тип данных в Python является изменяемым (можно менять его содержимое)?

1. str (строка)
2. tuple (кортеж)
3. list (список)
4. int (целое число)

Что такое PEP 8?

1. Версия Python
2. Стандарт оформления кода Python (как правильно писать код)
3. Библиотека для работы с данными
4. Программа для написания кода

Какая функция используется для получения длины списка?

1. size()
2. length()
3. len()
4. count()

Что выведет следующий код?

```
for i in range(3):
    print(i)
```

1. 1 2 3
2. 0 1 2
3. 0 1 2 3
4. 1 2

Какая комбинация клавиш используется для запуска программы в Visual Studio Code?

1. Ctrl+S
2. F5
3. Ctrl+R
4. Alt+F4

Что такое переменная в Python?

1. Постоянное число, которое нельзя изменить
2. Имя для хранения данных, которые можно менять
3. Специальная команда для вывода текста

4. Тип файла

Какой знак используется для комментариев в Python?

1. //
2. #
3. /* */
4. --

Что делает функция print() в Python?

1. Сохраняет файл на диск
2. Выводит текст или данные на экран
3. Удаляет переменную
4. Запускает программу

Какое расширение имеют файлы с кодом на Python?

1. .txt
2. .py
3. .exe
4. .html

Что такое цикл в программировании?

1. Ошибка в программе
2. Повторение одних и тех же действий несколько раз
3. Способ сохранения данных
4. Тип переменной

Какая команда используется для создания функции в Python?

1. function
2. def
3. create
4. func

Что такое список (list) в Python?

1. Набор команд для выполнения
2. Коллекция элементов, которые можно менять
3. Тип цикла
4. Способ вывода текста

Какой оператор используется для проверки условия "равно" в Python?

1. =
2. ==
3. !=
4. <>

Что такое ООП (объектно-ориентированное программирование)?

1. Способ написания программ с использованием объектов и классов
2. Тип ошибки в коде
3. Программа для создания игр
4. Язык программирования

Что такое класс в Python?

1. Группа учеников
2. Шаблон для создания объектов

3. Тип переменной
4. Команда для вывода текста

Какая команда используется для установки библиотек в Python?

1. install
2. pip install
3. download
4. get

Что делает команда input() в Python?

1. Выводит текст на экран
2. Получает данные от пользователя с клавиатуры
3. Сохраняет файл
4. Удаляет переменную

Какой тип данных используется для хранения текста в Python?

1. int
2. float
3. str
4. bool

Что такое отладка программы?

1. Удаление программы
2. Поиск и исправление ошибок в коде
3. Запуск программы
4. Сохранение программы

PYTHON В IT - СФЕРЕ

Для чего используется библиотека Pygame?

1. Для создания веб-сайтов
2. Для создания 2D-игр
3. Для работы с базами данных
4. Для анализа данных

Какой фреймворк используется для создания веб-приложений на Python?

1. Unity
2. Blender
3. Flask или Django
4. Pygame

Что такое Git?

1. Язык программирования
2. Система контроля версий (для сохранения истории изменений кода)
3. Библиотека для создания игр
4. Программа для 3D-моделирования

Для чего используется библиотека NumPy?

1. Для создания игр
2. Для работы с числовыми массивами и вычислений
3. Для создания веб-сайтов
4. Для работы со звуком

Что такое виртуальное окружение (venv) в Python?
1. 3D-мир в игре
2. Изолированная среда для установки библиотек для конкретного проекта
3. Программа для создания виртуальной реальности
4. Тип переменной

Какая библиотека используется для создания графиков и диаграмм в Python?
1. Pygame
2. Flask
3. Matplotlib
4. Tkinter

Что такое Tkinter?
1. Библиотека для создания графических интерфейсов (окон с кнопками)
2. Игровой движок
3. Система управления базами данных
4. Язык программирования

Для чего используется SQLite?
1. Для создания игр
2. Для хранения данных в базе данных
3. Для создания 3D-моделей
4. Для работы с изображениями

Что такое API в Blender?
1. Способ управления Blender с помощью кода на Python
2. Тип 3D-модели
3. Программа для рендеринга
4. Формат файла

РАБОТА С PYGAME

Что такое игровой цикл (game loop)?
1. Ошибка в игре
2. Постоянное повторение действий: обработка событий, обновление и отрисовка
3. Тип переменной
4. Способ сохранения игры

Что такое спрайт в Pygame?
1. Тип переменной
2. Графический объект (картинка персонажа или предмета)
3. Звуковой эффект
4. Тип цикла

Какая функция используется для создания окна игры в Pygame?
1. pygame.window()
2. pygame.display.set_mode()
3. pygame.create_window()
4. pygame.screen()

Что такое коллизия (collision) в играх?

1. Ошибка в коде
2. Столкновение двух объектов в игре
3. Тип анимации
4. Способ сохранения игры

Какой метод используется для загрузки изображения в Pygame?

1. pygame.load()
2. pygame.image.load()
3. pygame.get_image()
4. pygame.open()

ВЕБ – РАЗРАБОТКА С PYTHON

Что такое маршрутизация (routing) в Flask?

1. Способ отправки писем
2. Связывание URL-адресов с функциями в коде
3. Тип базы данных
4. Способ создания графики

Что такое шаблон (template) в веб-разработке?

1. Готовый код игры
2. HTML-файл с местами для вставки данных из Python
3. Тип переменной
4. Способ сохранения данных

Что такое форма на веб-странице?

1. Геометрическая фигура
2. Элемент для ввода данных пользователем (логин, пароль и т.д.)
3. Тип цикла
4. Способ вывода текста

Что такое база данных?

1. Программа для создания игр
2. Организованное хранилище информации
3. Тип переменной
4. Язык программирования

Что такое ORM (SQLAlchemy)?

1. Тип игры
2. Способ работы с базой данных через объекты Python (без написания SQL)
3. Библиотека для создания графики
4. Программа для 3D-моделирования

РАБОТА С ДАННЫМИ

Что такое массив в NumPy?

1. Тип игры
2. Набор чисел, организованных в таблицу
3. Способ создания графики
4. Тип переменной

Для чего используется библиотека Pandas?

1. Для создания игр
2. Для анализа и обработки табличных данных

3. Для создания веб-сайтов
4. Для работы со звуком

Что такое CSV-файл?

1. Тип игры
2. Текстовый файл с данными, разделенными запятыми
3. Программа для создания графики
4. Язык программирования

Что делает библиотека Matplotlib?

1. Создает игры
2. Создает графики и диаграммы для визуализации данных
3. Создает веб-сайты
4. Работает с базами данных

ЛОГИЧЕСКИЕ ЗАДАЧИ И АЛГОРИТМЫ

У вас есть список чисел: [1, 2, 3, 4, 5]. Что выведет следующий код?

```
numbers = [1, 2, 3, 4, 5]
print(numbers[2])
```

1. 1
2. 2
3. 3
4. 4

Сколько раз выполнится цикл?

```
for i in range(5):
    print(i)
```

1. 4 раза
2. 5 раз
3. 6 раз
4. Бесконечно

Что выведет следующий код?

```
x = 10
if x > 5:
    print("Больше")
else:
    print("Меньше")
```

1. Больше
2. Меньше
3. 10
4. Ошибка

У вас есть 10 яблок. Вы хотите разделить их поровну между 3 друзьями. Сколько яблок останется?

```
apples = 10
friends = 3
remainder = apples % friends
```

1. 0
2. 1

3. 2
4. 3

Какой результат даст следующее выражение?

```
result = 2 ** 3
print(result)
```

1. 5
2. 6
3. 8
4. 9

Вам нужно написать программу, которая проверяет, является ли число четным. Какое условие нужно использовать?

1. if number / 2 == 0:
2. if number % 2 == 0:
3. if number * 2 == 0:
4. if number + 2 == 0:

У вас есть список имен: ["Анна", "Борис", "Вера"]. Как добавить в конец списка имя "Глеб"?

1. names.add("Глеб")
2. names.append("Глеб")
3. names.insert("Глеб")
4. names.push("Глеб")

Что такое рекурсия в программировании?

1. Ошибка в коде
2. Функция, которая вызывает сама себя
3. Тип цикла
4. Способ сохранения данных

Сколько элементов в следующем словаре?

```
student = {"имя": "Иван", "возраст": 15, "класс": 9}
```

1. 1
2. 2
3. 3
4. 6

Что выведет следующий код?

```
text = "Python"
print(text[0] + text[-1])
```

1. Pn
2. Py
3. Python
4. Ошибка

ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ

Что такое Agile/Scrum?

1. Язык программирования
2. Методология гибкой разработки проектов
3. Библиотека Python

4. Программа для создания игр

Что такое спринт в разработке проекта?

1. Быстрый бег
2. Короткий период времени (1-2 недели) для выполнения части проекта
3. Тип ошибки
4. Способ сохранения кода

Что такое бэклог продукта?

1. Ошибка в программе
2. Список всех задач, которые нужно выполнить в проекте
3. Тип переменной
4. Программа для тестирования

Для чего используется Trello или Notion?

1. Для написания кода
2. Для организации задач и планирования проекта
3. Для создания игр
4. Для работы с базами данных

Что такое MVP (Minimum Viable Product)?

1. Самая дорогая версия продукта
2. Минимальная рабочая версия продукта с основными функциями
3. Тип ошибки
4. Язык программирования

Приложение 5

Проект является одним из видов самостоятельной работы, предусмотренной в ходе обучения по программе. Педагог оказывает консультационную помощь в выполнении проекта.

В комплект обязательных материалов, которые представляются обучающимся, входит: исходный код программы в архиве, презентация проекта.

Бланк оценки индивидуальных/групповых проектов

ФИО члена комиссии _____ Дата _____

Номер	Фамилия имя автора (-ов)	Название проекта	Время защиты	Критерий 1 Актуальность проекта 0-3 балла	Критерий 2 Постановка проблемы 0-3 балла	Критерий 3 Целеполагание 0-3 балла	Критерий 4 Качество результата 0-3 балла	Критерий 5 Практическая реализация 0-3 балла	Критерий 6 Защита проекта 0-3 балла	Итого

_____/_____

подпись расшифровка