

**РЕГИОНАЛЬНЫЙ ЦЕНТР ВЫЯВЛЕНИЯ, ПОДДЕРЖКИ И РАЗВИТИЯ
СПОСОБНОСТЕЙ И ТАЛАНТОВ ДЕТЕЙ И МОЛОДЕЖИ
СТАВРОПОЛЬСКОГО КРАЯ «СИРИУС 26»**

СОГЛАСОВАНО

Экспертным советом регионального центра
выявления, поддержки и развития
способностей и талантов детей и молодежи
Ставропольского края «Сириус 26»,
протокол № 1/2025 от 03.02.2025 г.

УТВЕРЖДЕНО

Директором Центра «Поиск»
Томилиной О.А.

приказ № 13/1 от 04.02.2025 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«РАЗРАБОТКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»

Направленность: техническая

Возраст обучающихся: 13-16 лет

Объем программы: 90 часов

Срок освоения: 2 месяца

Форма обучения: очная с использованием дистанционных
образовательных технологий

Автор программы: Пономаренко Елена Александровна, руководитель
структурного подразделения - методического
объединения информационных технологий Центра
«Поиск»;
Решетняк Ольга Владимировна, педагог
дополнительного образования центра «Поиск»

Ставрополь
2025

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	2
УЧЕБНЫЙ ПЛАН7	
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК9	
РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «ВВЕДЕНИЕ В ООП»10	
РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «ВВОДНО-ОТБОРЧНЫЙ КУРС: ВВЕДЕНИЕ В ООП»11	
РАБОЧАЯ ПРОГРАММА КУРСА «РАЗРАБОТКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»12	
СОДЕРЖАНИЕ КУРСА «РАЗРАБОТКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»15	
РАБОЧАЯ ПРОГРАММА УЧЕБНО-ТРЕНИНГОВОГО КУРСА «РАЗРАБОТКА ИГР НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»19	
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ.....	22
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	23
КАДРОВОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ.....	29
МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ	30
УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ	31
ОБЕСПЕЧЕНИЕ ПРОГРАММЫ	31

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Основные характеристики программы

В настоящее время, в условиях экономических санкций, российская экономика переживает значительные изменения, особенно в сфере информационных технологий. С уходом зарубежных разработчиков программного обеспечения возникает необходимость в квалифицированных специалистах, способных эффективно решать задачи государства. В данной ситуации особую актуальность приобретает быстрое наращивание числа программистов, владеющих языком Python, который зарекомендовал себя как один из наиболее востребованных языков программирования.

Программирование занимает центральное место в подготовке кадров в области информационных технологий. В рамках федерального проекта «Развитие кадрового потенциала ИТ-отрасли» в национальной программе «Цифровая экономика РФ» подчеркивается исключительная значимость изучения языков программирования, включая Python. Следовательно, обучение программированию, в частности языку Python, следует начинать уже на уровне школьного образования.

1.1. Направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Разработка приложений на Python» имеет техническую направленность и охватывает два ключевых аспекта изучения:

- 1) Технологический: обучение языку Python рассматривается как способ формирования образовательного потенциала, необходимого для освоения современных цифровых технологий.
- 2) Общеобразовательный: программа нацелена на развитие основных познавательных процессов, таких как анализ, выявление закономерностей, создание инструкций и логические выводы.

1.2. Адресат программы

Программа предназначена для учащихся 13-16 лет. Она ориентирована на одаренных школьников 8-10 классов, проявляющих интерес к информатике и желающих развить навыки программирования, получая углубленные теоретические и практические знания по актуальным темам в сфере информационных технологий.

1.3. Актуальность программы

Совершенствование технологий и программного обеспечения привело к сокращению часов, выделяемых для изучения программирования в школьных учебных планах. Современные визуальные и мультимедийные инструменты

часто отвлекают школьников от изучения языков программирования. Простые инструменты для поиска информации в интернете подразумевают более легкий подход, что снижает интерес к углубленному пониманию программных концепций. В большинстве школ отсутствует систематическое обучение языкам программирования.

Дополнительная общеобразовательная общеразвивающая программа «Разработка на языке программирования Python» предоставляет учащимся возможность освоить актуальный язык, формируя аналитические способности и навыки для работы с данными, а также облегчая процесс обучения за счет наглядности и доступности.

1.4. Отличительные особенности/новизна программы

Содержание программы «Разработка на языке программирования Python» разработано с учетом современных требований к подготовке школьников к жизни и работе в цифровом обществе, в рамках ключевых направлений федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика РФ».

Новизна программы «Разработка на языке программирования Python» заключается в предоставлении учащимся возможности изучать один из самых популярных языков программирования, который активно используется в различных областях, таких как веб-разработка, анализ данных, искусственный интеллект и автоматизация. Программа обучает основам синтаксиса языка, принципам объектно-ориентированного программирования и дает возможность применять полученные знания на практике через создание собственных компьютерных приложений и проектов. По окончании курса учащиеся будут готовы продолжить свое образование на более углубленном уровне в области программирования.

Уровень освоения программы – углубленное изучение языка программирования Python.

1.5. Объем и срок освоения программы

Объем программы – 90 часов.

Срок реализации программы – 2 месяца.

1.6. Цели и задачи программы

Цель программы – выявление, развитие и продвижение одаренных детей Ставропольского края в сфере программирования, а также подготовка их к участию в государственных инициативах по поддержке таланта. Программа формирует у школьников базовые знания и навыки в программировании на Python, а также осознание фундаментальных концепций и технологий, необходимых для создания современных программных решений.

Задачи программы

1) Обучающие:

- освоение синтаксиса Python и принципов ООП;
- изучение библиотеки Pygame для создания игр;
- приобретение навыков работы с базами данных SQLite;
- обучение проектированию архитектуры приложений.

2) Развивающие:

- развитие алгоритмического и проектного мышления;
- формирование навыков работы с генеративной ИИ;
- развитие креативности при создании приложений.

3) Воспитательные:

- формирование навыков целеполагания и тайм-менеджмента;
- развитие умений работать в команде;
- ранняя профориентация школьников.

1.7. Планируемые результаты освоения программы

1. Предметные результаты:

- знание основ программирования на языке Python;
- владение принципами ООП;
- умение создавать приложения с использованием Pygame и SQLite.

2. Метапредметные результаты:

- способность применять генеративный ИИ в программировании;
- умение проектировать архитектуру приложений;
- навыки презентации и защиты проектов.

3. Личностные результаты:

- развитие навыков целеполагания и планирования;
- формирование ответственного отношения к проектной деятельности.

2. Организационно-педагогические условия реализации программы

2.1. Язык реализации программы

Реализация дополнительной общеобразовательной общеразвивающей программы «Разработка на языке программирования Python» осуществляется на государственном языке Российской Федерации (русском языке).

2.2. Форма обучения

Форма обучения – очная с применением дистанционных образовательных технологий.

2.3. Особенности реализации программы

Программа реализуется по модульному принципу с интеграцией дистанционных образовательных технологий.

- 1 модуль – дистанционный учебно-отборочный курс в течение 2-х;
- 2 модуль – очная профильная смена в течение 2-х недель;
- 3 модуль – дистанционный учебно-тренинговый курс в течение 3-х недель.

Основная часть содержания программы реализуется в формате очной профильной смены в течение 2-х недель.

Участие школьников в программе осуществляется на бюджетной основе.

2.4. Условия набора и формирования групп

К обучению допускаются учащиеся 8-10 классов общеобразовательных школ Ставропольского края, имеющие постоянную прописку в Ставропольском крае:

- 1) по результатам конкурсного отбора успешного прохождения учебно-отборочного курса «Введение в ООП»;
- 2) на основании участия в олимпиадах и других интеллектуальных конкурсах регионального и всероссийского уровней.

Условия конкурсного отбора обеспечивают соблюдение прав обучающихся в сфере дополнительного образования и гарантируют зачисление наиболее способных и подготовленных учащихся для освоения программы.

Состав групп формируется по принципу разновозрастные.

2.5. Формы организации и проведения занятий

Занятия организуются в дистанционном формате, в аудиториях (под непосредственным руководством преподавателя) и внеаудиторно (самостоятельная подготовка учащихся к олимпиадам вне учебного плана).

Формы проведения занятий включают комбинированные, теоретические, практические, самостоятельные, репетиционные и контрольные.

Организация деятельности учащихся может принимать различные формы:

- фронтальная: работа преподавателя с группой всех учащихся одновременно;
- групповая: организация работы в малых группах, включая пары, для выполнения конкретных задач, где виден вклад каждого участника;
- коллективная: проблемно-ориентированное взаимодействие, где все участники задействованы одновременно;
- индивидуальная: для коррекции пробелов в знаниях и отработки отдельных навыков.

Режим занятий:

- учебно-отборочный курс: дистанционно 2 недели в удобное для обучающегося время;
- профильная смена проходит в г. Ставрополе: очно, 8 уроков в день в течение 10 учебных дней;

- учебно-тренинговый курс: дистанционно 3 недели в удобное для обучающегося время.

Основные методы и формы реализации содержания программы

Метод двумерной дидактики

В рамках программы обучения широко используется метод двумерной дидактики. Данный метод становится актуальным, когда знаний и навыков, полученных обучающимися на уроках в школе, недостаточно для освоения курса «Язык программирования Python». Этот подход предполагает выбор таких форм обучения, которые позволяют не только решать задачи, но и глубоко понимать терминологию, технологию, а также осмысливать суть и смысл изучаемого материала. Учащиеся учатся оценивать преимущества и недостатки различных решений, предлагать альтернативные варианты и делают всё это в сжатые сроки. Суть метода двумерной дидактики заключается в организации результативного учебного процесса в соответствии с уровнем подготовки учащихся. Систематическое применение данного метода способствует более глубокому усвоению сложного материала, что может привести к опережающему усвоению знаний на несколько лет. Это достигается благодаря спиральному повторению информации, расширению понятийного поля и применению знаний в различных контекстах и с разных перспектив.

Проблемный метод

Проблемный метод охватывает ряд приемов, используемых для выполнения заданий с неоднозначными вариантами решения, особенно в условиях недостатка или избытка информации. Основная цель применения этого метода заключается в развитии аналитических навыков у обучающихся, таких как сравнение, аргументация и выводы, через активное участие в решении сложных задач. Все задания основываются на имеющихся знаниях, однако требуют самостоятельного поиска новых фактов и сведений. Осознание недостатка собственных знаний стимулирует учащихся к поиску новых знаний, что является важным условием для развития творческого мышления. Ключевым достоинством проблемного метода является овладение технологиями принятия решений в ситуациях неопределенности или неоднозначности, что подразумевает разработку различных стратегий решения задач с учетом ограничения информации и ресурсов.

Словесные методы

Лекция с обратной связью представляет собой один из словесных методов, при котором преподаватель периодически задает вопросы во время изложения теоретического материала, чтобы оценить усвоение содержания. Вопросы заранее формулируются для контроля понимания ключевых аспектов.

Эвристическая беседа, вопросно-ответная форма, основывается на греческом слове «эвристика», что означает «отыскать» или «открыть». Суть метода заключается в том, что преподаватель формирует последовательность вопросов, направляющих мысли учащихся в нужное русло. Эвристическая беседа основывается на интуитивных и неявных знаниях, которые учащиеся приобрели на основе самостоятельного опыта. Этот метод может быть особенно эффективен в качестве мотивационной беседы при знакомстве с новой темой.

Метод дизайн-мышления

Метод дизайн-мышления фокусируется на создании продуктов и услуг, которые ориентированы на потребности пользователей. Каждая идея представляет собой решение конкретной потребности человека.

Принципы дизайн-мышления:

- 1) Ошибайся раньше и чаще.
- 2) Создавай прототипы вместо того, чтобы просто рассказывать о продукте.
- 3) Первостепенно фиксируй пожелания пользователей.
- 4) Разрабатывай продукт совместно с пользователем.

Основные компоненты дизайн-мышления:

- 1) Процесс: присутствует алгоритм работы, интерактивность и формирование смешанных команд.
- 2) Подход: акцент на человекоцентричности, эмпатии и культуре быстрого допуска ошибок.
- 3) Среда: наглядный мыслительный процесс и возможность «думать руками».

УЧЕБНЫЙ ПЛАН

№ тем ы	Наименование модуля, учебного курса	Контактная работа обучающихся с преподавателем, часов			Формы контроля / аттестации
		Теория	Практика	Всего	
1.	Дистанционный учебно-отборочный курс «Введение в ООП»	3	3	6	Тест с самопроверкой. Разработка программы.
2.	Учебный курс «Разработка на языке программирования Python»	38	52	80	Разработка и дизайн приложения, разработка игры.
3.	Учебно-тренинговый	2	2	4	Разработка

	курс «Разработка игр на языке программирования Python»				проекта.
Итого:		42	57	90	

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Наименование модуля, учебного курса	Дата начала обучения	Дата окончания обучения	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
Дистанционный учебно-отборочный курс «Введение в ООП»	22.09.2025	08.10.2025	2	14	6	Дистанционное обучение
Учебный курс «Разработка на языке программирования Python»	03.11.2025	15.11.2025	2	10	80	Очное обучение, 5 раз в неделю по 8 часов
Учебно-тренинговый курс «Разработка игр на языке программирования Python»	15.11.2025	07.12.2025	2	4	4	Дистанционное обучение

РАБОЧАЯ ПРОГРАММА УЧЕБНО-ОТБОРОЧНОГО КУРСА «ВВЕДЕНИЕ В ООП»

Курс «Введение в ООП» предназначен для обучающихся 8-10 классов. Освоение языка Python и связанных с ним технологий является важным этапом для тех, кто стремится развивать навыки в программировании, создании приложений и разработке игр.

В рамках курса «Введение в ООП» рассматриваются следующие темы:
Вводно-отборочный курс:

- синтаксис языка Python и его основные конструкции;
- списки;
- кортежи;
- библиотеки;
- декораторы.

В результате успешного освоения курса обучающийся должен:

Знать:

- основы синтаксиса и конструкции языка Python;
- ключевые типы данных и способы их использования.

Уметь:

- формулировать задачи и разрабатывать алгоритмы.

Тематический план курса

№	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, час.		
		Теория	Практика	Всего
1	Синтаксис языка Python и его основные конструкции.	1		1
2	Списки.	1		1
3	Кортежи.		1	1
4	Библиотеки.		1	1
5	Декораторы.		1	1
6	Итоговый тест.		1	1
Итого:		2	4	6

СОДЕРЖАНИЕ УЧЕБНО-ОТБОРОЧНОГО КУРСА «ВВЕДЕНИЕ В ООП»

Курс «Введение в ООП» предназначен для обучающихся 8-10 классов, желающих освоить основы программирования на языке Python и научиться работать с его базовыми конструкциями.

Тема 1: Синтаксис и внешний вид кода.

Теория: Основы синтаксиса Python. Отступы, комментарии, структура программы. Переменные и основные типы данных.

Практика: Написание простых программ, использование отступов и комментариев.

Тема 2: Списки.

Теория: Понятие списка. Создание, индексация, срезы. Методы списков (добавление, удаление, поиск элементов).

Практика: Работа со списками: создание, модификация, обработка.

Тема 3: Кортежи.

Теория: Понятие кортежа. Отличия кортежей от списков. Использование кортежей.

Практика: Создание и использование кортежей в программах.

Тема 4: Библиотеки

Теория: Понятие модуля и библиотеки. Импорт модулей. Стандартные библиотеки Python.

Практика: Использование стандартных библиотек для решения задач.

Тема 5: Декораторы

Теория: Понятие декоратора. Создание и применение декораторов. Синтаксис декораторов.

Практика: Написание простых декораторов для функций.

Ожидаемые результаты.

По окончании курса обучающиеся смогут: понимать синтаксис языка Python и использовать его основные конструкции, работать со списками и кортежами, использовать стандартные библиотеки Python, создавать и применять декораторы.

Форма подведения итогов: тестирование с возможностью самопроверки и написание программного проекта, в котором используются изученные темы.

РАБОЧАЯ ПРОГРАММА КУРСА «РАЗРАБОТКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»

Курс «Разработка приложений на языке программирования Python» предназначен для обучающихся 8-10 классов. В рамках курса школьники научатся использовать язык Python для создания игровых проектов, освоят принципы объектно-ориентированного программирования и получают практический опыт полного цикла разработки — от проектирования до защиты готового продукта. Участники получают навыки работы с современными технологиями, включая игровые движки, базы данных и искусственный интеллект. В курсе рассматриваются следующие темы:

1. Основы игровой разработки и Python:
 - знакомство с Python и игровым движком Pygame;
 - создание игрового окна и игрового цикла;
 - работа с графикой и основы геймдизайна;
 - обработка пользовательского ввода с клавиатуры.
2. Объектно-ориентированное программирование:
 - введение в ООП: классы и объекты;
 - принципы инкапсуляции и защита данных;
 - наследование и композиция;
 - полиморфизм и универсальные методы.
3. Проектирование и разработка приложений:
 - проектирование приложения "Персональный менеджер задач";
 - создание ядра приложения;
 - интеграция логики приложения и графического интерфейса;
 - обработка действий пользователя.
4. Игровые механики и системы:
 - реализация взаимодействия игровых объектов;
 - создание систем счета, здоровья и жизней;
 - разработка искусственного интеллекта для врагов;
 - механики атаки, защиты и анимаций.
5. Работа с данными и базами данных:
 - сохранение данных приложений;
 - введение во встраиваемую БД SQLite;
 - основные команды языка SQL;
 - создание игрового мира с коллизиями.

6. Интерфейсы и визуальное оформление:

- основы графических библиотек;
- создание пользовательских интерфейсов;
- отображение данных и статистики;
- полировка игрового интерфейса.

7. Современные технологии:

- программирование с использованием генеративного ИИ;
- интеграция AI-инструментов в разработку;
- автоматизация процессов разработки.

8. Психология и презентация проектов:

- тренинги по целеполаганию и тайм-менеджменту;
- подготовка к защите проектов;
- публичная презентация результатов;
- финальная сборка и оптимизация.

По окончании курса обучающийся должен:

Знать:

- основы языка Python и его применение в игровой разработке;
- принципы объектно-ориентированного программирования;
- методы разработки игр с использованием библиотеки Pygame;
- основы работы с базами данных SQLite и языком SQL;
- принципы создания пользовательских интерфейсов и обработки ввода;
- основы генеративного искусственного интеллекта и его применение;
- методы проектирования архитектуры приложений;
- принципы тестирования и отладки программного обеспечения.

Уметь:

- разрабатывать игровые приложения с использованием Pygame;
- применять принципы ООП для создания структурированного кода;
- проектировать и реализовывать игровые механики и системы;
- работать с базами данных для хранения и управления информацией;
- создавать графические интерфейсы для различных типов приложений;
- интегрировать инструменты искусственного интеллекта в разработку;
- проводить тестирование и отладку собственных проектов;
- эффективно планировать время и ставить цели в проектной работе;

- презентовать и защищать готовые решения перед аудиторией;
- анализировать требования и проектировать архитектуру приложений;
- работать с графикой, анимациями и визуальными эффектами;
- реализовывать взаимодействие между различными компонентами приложения.

Тематический план

№ темы	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, часов		
		Теория	Практика	Всего
1	Знакомство с Python и игровым движком. Основы Pygame.	2	2	4
2	Введение в ООП. Проектирование приложения "Персональный менеджер задач".	2	4	6
3	Графика и основы геймдизайна. Обработка клавиатуры.	2	4	6
4	Классы и объекты. Создание ядра приложения.	2	4	6
5	Игровая механика. Создание системы счета. Создание системы здоровья/жизней. Добавление анимаций.	2	4	6
6	Инкапсуляция. Защита данных.	2	4	6
7	Наследование и композиция. Полиморфизм. Универсальный вывод информации.	2	4	6
8	Введение во встраиваемую БД SQLite. Основные команды языка SQL.	2	4	6
9	Создание игрового мира. Препятствия.	2	2	4
10	Основы графических библиотек. Создание интерфейса приложения. Отображение данных.	2	4	6
11	Работа над индивидуальными проектами.	4	8	12
12	Программирование на Python с	2	2	4

	использованием генеративного искусственного интеллекта.			
13	Психологические тренинги «Целеполагание» и «Тайм-менеджмент».	2	6	8
Итого		28	52	80

СОДЕРЖАНИЕ КУРСА «РАЗРАБОТКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»

Тема 1. Знакомство с Python и игровым движком. Основы Pygame.

Теория. Введение в язык Python: синтаксис, переменные, основные конструкции. Понятие игрового движка и его компонентов. Библиотека Pygame: установка, основные модули и возможности. Структура игрового приложения: инициализация, основной цикл, завершение. Создание игрового окна: настройка размеров, заголовка, режимов отображения.

Практика. Установка Python и библиотеки Pygame. Создание базового игрового окна с заданными параметрами. Реализация основного игрового цикла с обработкой событий. Написание простой программы с отображением базовой графики.

Тема 2. Введение в ООП. Проектирование приложения "Персональный менеджер задач". Классы и объекты. Создание ядра приложения.

Теория. Основные принципы объектно-ориентированного программирования. Понятия класса, объекта, атрибута, метода. Конструкторы и инициализация объектов. Проектирование структуры приложения "Менеджер задач". Принципы модульности и повторного использования кода.

Практика. Создание классов Task, TaskList для управления задачами. Реализация методов добавления, удаления и редактирования задач. Разработка базового интерфейса для взаимодействия с задачами. Создание ядра приложения с основной бизнес-логикой.

Тема 3. Графика и основы геймдизайна. Обработка клавиатуры.

Теория. Работа с графикой в Pygame: поверхности, цвета, примитивы. Загрузка и отображение изображений (спрайтов). Основы композиции и визуального дизайна в играх. Обработка событий клавиатуры: нажатие, отпускание, удержание. Система координат и преобразования в игровом пространстве.

Практика. Создание графических элементов интерфейса. Загрузка и отображение спрайтов персонажа и фона. Реализация управления персонажем с помощью клавиатуры. Добавление визуальных эффектов и анимаций.

Тема 4. Игровая механика. Создание системы счета. Создание системы здоровья/жизней. Добавление анимаций.

Теория. Проектирование игровых механик и взаимодействий между объектами. Системы счета и прогрессии в играх. Механики здоровья, урона и жизней. Принципы создания анимаций: кадровая анимация, трансформации. Обработка столкновений и взаимодействий между игровыми объектами.

Практика. Реализация системы подсчета очков за сбор предметов. Создание системы здоровья с визуальным отображением. Разработка механики получения урона и восстановления. Добавление анимаций для персонажей и игровых объектов.

Тема 5. Инкапсуляция. Защита данных. Наследование и композиция. Полиморфизм. Универсальный вывод информации.

Теория. Принцип инкапсуляции: приватные и защищенные атрибуты. Методы доступа (геттеры и сеттеры) и валидация данных. Наследование классов: создание иерархий объектов. Композиция как альтернатива наследованию. Полиморфизм: единый интерфейс для различных типов объектов.

Практика. Рефакторинг существующих классов с применением инкапсуляции. Создание иерархии классов игровых объектов через наследование. Реализация композиции для сложных игровых entities. Разработка универсальных функций для работы с различными типами объектов.

Тема 7. Введение во встраиваемую БД SQLite. Основные команды языка SQL. Создание игрового мира. Препятствия.

Теория. Основы реляционных баз данных и SQLite. Структура SQL-запросов: SELECT, INSERT, UPDATE, DELETE. Создание таблиц и определение схемы данных. Проектирование игровых миров: карты, локации, препятствия. Системы коллизий и физики в игровом мире.

Практика. Создание базы данных для хранения прогресса игры. Реализация сохранения и загрузки состояния игры. Разработка системы препятствий с обработкой коллизий. Создание многоуровневой структуры игрового мира.

Тема 8. Основы графических библиотек. Создание интерфейса приложения. Отображение данных.

Теория. Обзор графических библиотек Python: Tkinter, PyQt, Kivy. Принципы проектирования пользовательских интерфейсов. Компоновка элементов интерфейса: менеджеры геометрии. Визуализация данных: графики, диаграммы, таблицы. Обработка пользовательского ввода в графических интерфейсах.

Практика. Создание главного меню игры с использованием графической библиотеки. Разработка интерфейса для отображения статистики и достижений. Реализация системы настроек игры с графическим интерфейсом. Визуализация игровых данных в виде графиков и диаграмм.

Тема 9. Программирование на Python с использованием генеративного искусственного интеллекта.

Теория. Введение в генеративный искусственный интеллект. Обзор доступных AI-инструментов и API для Python. Принципы prompt-инжиниринга и работы с языковыми моделями. Этические аспекты использования ИИ в программировании. Интеграция AI-компонентов в традиционные приложения.

Практика. Написание кода с использованием AI-ассистентов для генерации фрагментов. Создание простого чат-интерфейса с интеграцией языковой модели. Использование ИИ для генерации игрового контента (диалоги, описания). Автоматизация рутинных задач программирования с помощью AI-инструментов.

Тема 10. Работа над индивидуальными проектами. Психологические тренинги «Целеполагание» и «Тайм-менеджмент».

Теория. Методы планирования и управления проектами. Принципы постановки SMART-целей. Техники тайм-менеджмента: матрица Эйзенхауэра, Pomodoro. Стратегии преодоления прокрастинации в разработке. Принципы самоорганизации и мотивации.

Практика. Постановка целей для индивидуального проекта по SMART-критериям. Разработка плана работы над проектом с временными рамками. Применение техник тайм-менеджмента в ежедневной работе. Рефлексия прогресса и корректировка планов разработки.

Тема 10. Защита индивидуальных проектов.

Теория. Структура успешной презентации проекта. Принципы эффективного публичного выступления. Подготовка демонстрационных материалов. Стратегии ответов на вопросы и получения обратной связи. Анализ и рефлексия проделанной работы.

Практика. Подготовка презентации собственного проекта. Репетиция выступления с таймингом. Демонстрация рабочего проекта и его функционала. Ответы на вопросы аудитории и получение обратной связи. Коллективное обсуждение результатов.

РАБОЧАЯ ПРОГРАММА УЧЕБНО-ТРЕНИНГОВОГО КУРСА «РАЗРАБОТКА ИГР НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»

Курс «Разработка игр на языке программирования Python» предназначен для обучающихся, заинтересованных в разработке качественных игровых решений.

В курсе «Разработка игр на языке программирования Python» рассматриваются основные принципы проектирования игровых приложений и основы тестирования программного обеспечения.

В результате освоения учебного курса обучающийся должен:

Знать:

- основные архитектурные паттерны десктопных приложений (такие как MVC, MVVM и т.д.);
- этапы жизненного цикла разработки программного обеспечения;
- принципы организации кода при разработке приложений;
- основы тестирования программного обеспечения, включая различные методологии тестирования и написание тестов;
- виды тестирования: юнит-тестирование, интеграционное тестирование, функциональное тестирование;
- принципы объектно-ориентированного программирования и их применение в игровой разработке;
- основы работы с графическими библиотеками и создания пользовательских интерфейсов;
- методы хранения и управления данными в игровых приложениях.

Уметь:

- проектировать архитектуру игровых приложений с использованием выбранных архитектурных паттернов;
- разрабатывать функциональные модули, соблюдая лучшие практики программирования;
- проводить тестирование программного обеспечения, применяя различные методы и инструменты;
- писать юнит-тесты для проверки отдельных модулей и компонентов приложения;
- анализировать и исправлять ошибки на основе результатов тестирования, обеспечивая качество конечного продукта;
- реализовывать игровую логику с применением принципов объектно-ориентированного программирования;
- создавать пользовательские интерфейсы для игровых приложений;

- организовывать взаимодействие между различными компонентами игровой системы;
- работать с базами данных для сохранения игрового прогресса и статистики.

Тематический план

№ темы	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, часов		
		Теория	Практика	Всего
1	ООП-структура игрового приложения.		1	1
2	Графика и пользовательский интерфейс в играх.		1	1
3	Работа с данными и сохранение игрового прогресса.		1	1
4	Основы тестирования игровых приложений.		1	1
Итого			4	4

СОДЕРЖАНИЕ КУРСА «РАЗРАБОТКА ИГР НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON»

Тема 1. ООП-структура игрового приложения.

Теория. Обзор основных архитектурных паттернов игровых приложений (MVC, MVVM и др.), а также этапы жизненного цикла разработки игрового ПО. Знакомство с основами организации кода и принципами SOLID в контексте игровой разработки. Принципы объектно-ориентированного программирования и их применение в создании игровых объектов.

Практика. Проектирование архитектуры игрового приложения с использованием выбранного паттерна и разработка основных игровых модулей (игрок, враги, система счета) в соответствии с принятыми стандартами. Создание системы взаимодействия игровых объектов.

Форма подведения итогов: тестирование реализации игровой архитектуры и разработки кода с самопроверкой.

Тема 2. Графика и пользовательский интерфейс в играх.

Теория. Основы геймдизайна и создания пользовательского опыта. Принципы работы с графическими библиотеками Pygame. Организация игрового интерфейса (HUD), создание меню и экранов игры. Основы анимации и визуальных эффектов.

Практика. Разработка игрового интерфейса с отображением счета, здоровья, времени. Создание системы меню (стартовый экран, пауза, завершение игры). Реализация анимации игровых объектов и визуальной обратной связи.

Форма подведения итогов: оценка качества и функциональности игрового интерфейса с самопроверкой.

Тема 3. Работа с данными и сохранение игрового прогресса.

Теория. Введение в базы данных SQLite для хранения игровых данных. Основные команды языка SQL для работы с игровой статистикой. Принципы организации хранения игрового прогресса, таблиц рекордов и настроек. Методы сериализации данных в игровых приложениях.

Практика. Создание системы сохранения и загрузки игрового прогресса. Реализация таблицы рекордов с сохранением в БД. Организация хранения статистики игрока и настроек приложения.

Форма подведения итогов: тестирование работы системы сохранения данных и корректности работы с базой данных.

Тема 4. Основы тестирования игровых приложений.

Теория. Введение в тестирование игрового ПО, виды тестирования (юнит-тестирование игровых механик, интеграционное тестирование взаимодействия систем, функциональное тестирование игрового процесса). Обзор инструментов для тестирования и отладки игрового кода. Методы тестирования графических компонентов и пользовательского интерфейса.

Практика. Написание юнит-тестов для проверки функциональности игровых модулей (движение, столкновения, подсчет очков). Интеграционное тестирование взаимодействия игровых систем. Функциональное тестирование игрового процесса. Анализ и отладка ошибок на основе результатов тестирования.

Форма подведения итогов: оценка качества написанных тестов и проведенной отладки с самопроверкой.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

№ п/п	Название раздела, темы	Формы учебного занятия	Формы, методы, приемы обучения. Педагогические технологии	Материально- техническое оснащение, дидактико- методический материал	Форма подведения итогов
1	Тема 1. Разработка игр	Комбинированная	Объяснительно- иллюстративный. Частично-поисковый. Исследовательский..	Проекционное оборудование, ПК. Доступ к сети Интернет.	Разработка проекта
2	Тема 2. Персональный менеджер задач	Комбинированная	Объяснительно- иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Разработка проекта
3	Тема 3. Программирование на Python с использованием генеративного искусственного интеллекта	Комбинированная	Объяснительно- иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
4	Тема 4. Личностное развитие	Комбинированная	Объяснительно- иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
5	Тема 5. Защита индивидуальных проектов	Комбинированная	Объяснительно- иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Защита проекта

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

В процессе обучения проводятся разные виды контроля результативности усвоения программного материала.

1) Входной контроль

Входной контроль проводится целью выявления первоначального уровня знаний и умений, возможностей обучающихся. Входной контроль проводится с каждым обучающимся индивидуально по следующим параметрам – теоретическим и практическим.

Формы: тестирование и проектная деятельность.

Отборочный тест проводится в рамках дистанционного учебно-отборочного курса с целью отбора участников очной профильной смены.

1. Отборочный тест состоит из 25 заданий разного уровня сложности из разделов «Введение в ООП» и «Разработка на языке программирования Python», направлен на проверку основных понятий, рассматриваемых тем.
2. Разработка творческого проекта с целью определения уровня умений и навыков самостоятельной работы, критического мышления и творческого подхода обучающихся на основе имеющихся знаний.

Во время проведения входной диагностики педагог заполняет информационную карточки «Результаты входной диагностики», пользуясь шкалой «Оценка параметров входного контроля».

Оценка параметров входного контроля

Наименование уровня	Результат диагностики, %
Элементарный уровень	0 - 54%
Низкий уровень	55 - 69%
Средний уровень	70 - 84%
Высокий уровень	85 - 100%

Примерные задания:

1. Что такое PEP в мире Python?
 - А) Простое руководство для начинающих
 - В) Документ с предложениями по улучшению Python

- C) Название встроенного модуля
D) Специальный тип данных
2. Какой из этих способов создания списка является правильным?
A) `list = (1, 2, 3)`
B) `list = {1, 2, 3}`
C) `list = [1, 2, 3]`
D) `list = <1, 2, 3>`
3. Что такое кортеж (tuple)?
A) Изменяемая последовательность элементов
B) Неупорядоченная коллекция элементов
C) Упорядоченная неизменяемая последовательность
D) Словарь без ключей
4. Для чего используется метод `list.append(x)`?
A) Для удаления элемента `x` из списка
B) Для добавления элемента `x` в начало списка
C) Для добавления элемента `x` в конец списка
D) Для поиска элемента `x` в списке
5. Какой оператор используется для распаковки кортежа?
A) `+`
B) `*`
C) `=`
D) `,`
6. Что делает конструкция `from math import sqrt`?
A) Импортирует весь модуль `math`
B) Импортирует только функцию `sqrt` из модуля `math`
C) Импортирует все функции, кроме `sqrt`
D) Создает псевдоним для модуля `math`
7. Какие из этих утверждений о списках и кортежах верны?
(Выберите ВСЕ правильные варианты)
A) Список можно изменить после создания
B) Кортеж можно изменить после создания
C) Кортеж занимает меньше памяти, чем список
D) Список можно использовать как ключ словаря
8. Что выведет код: `print((1, 2) + (3, 4))`?
A) `(1, 2, 3, 4)`
B) `(4, 6)`

- C) (1, 2)(3, 4)
- D) Ошибку

9. Какой декоратор превращает метод класса в свойство, которое можно вызывать без скобок?

- A) `@classmethod`
- B) `@property`
- C) `@staticmethod`
- D) `@decorate`

10. Что такое декоратор в Python?

- A) Функция, которая рисует узоры в консоли
- B) Специальный синтаксис для оформления строк
- C) Функция, которая изменяет поведение другой функции
- D) Инструмент для работы с база данных

11. Какие методы есть и у списков, и у кортежей? (Выберите ВСЕ правильные варианты)

- A) `.append()`
- B) `.count()`
- C) `.index()`
- D) `.sort()`

12. Для чего используется виртуальное окружение (venv)?

- A) Для запуска программ в облаке
- B) Для изоляции зависимостей проекта
- C) Для создания графического интерфейса
- D) Для ускорения выполнения кода

13. Что означает запись `*args` в определении функции?

- A) Любые именованные аргументы
- B) Любые позиционные аргументы
- C) Только строковые аргументы
- D) Аргументы по умолчанию

14. Какие из этих типов данных являются изменяемыми? (Выберите ВСЕ правильные варианты)

- A) Кортеж (tuple)
- B) Список (list)
- C) Строка (string)
- D) Словарь (dict)

15. Что делает функция `dir(module)`?

- A) Удаляет модуль
- B) Показывает список методов и атрибутов модуля
- C) Устанавливает модуль
- D) Запускает модуль на выполнение

16. Как создать копию списка `my_list`?

- A) `new_list = my_list`
- B) `new_list = my_list.copy()`
- C) `new_list = copy(my_list)`
- D) `new_list = my_list.clone()`

17. Что такое импорт с псевдонимом?

- A) `import math as m`
- B) `import math from m`
- C) `from math import m`
- D) `import m from math`

18. Какие из этих декораторов существуют в стандартной библиотеке Python? (Выберите ВСЕ правильные варианты)

- A) `@property`
- B) `@lru_cache`
- C) `@timer`
- D) `@dataclass`

19. Для чего используется декоратор `@classmethod`?

- A) Для создания метода, который работает с экземпляром класса
- B) Для создания метода, который работает с самим классом
- C) Для создания приватного метода
- D) Для создания статического метода

20. Что выведет код: `print([x for x in 'hello' if x != 'l'])`?

- A) `['h', 'e', 'o']`
- B) `['hello']`
- C) `['h', 'e', 'l', 'l', 'o']`
- D) Ошибку

21. Какие из этих модулей входят в стандартную библиотеку Python? (Выберите ВСЕ правильные варианты)

- A) `math`
- B) `datetime`
- C) `requests`
- D) `json`

22. Что делает метод `list.insert(i, x)`?
- A) Удаляет элемент на позиции `i`
 - B) Вставляет `x` на позицию `i`
 - C) Заменяет элемент на позиции `i` на `x`
 - D) Ищет `x` на позиции `i`
23. Какой декоратор рекомендуется использовать, чтобы сохранить имя и документацию исходной функции?
- A) `@save_meta`
 - B) `@functools.wraps`
 - C) `@keep_name`
 - D) `@documentation`
24. Какие из этих утверждений о декораторах верны? (Выберите ВСЕ правильные варианты)
- A) Могут изменять поведение функции
 - B) Всегда должны возвращать число
 - C) Могут быть применены к классам
 - D) Исполняются во время импорта модуля
25. Что такое генератор списка (list comprehension)?
- A) Функция для генерации случайных списков
 - B) Способ создания списка на основе другого списка
 - C) Модуль для работы со списками
 - D) Метод для сортировки списка

2) *Текущий контроль* проводится в рамках очной профильной смены на занятиях в виде наблюдения за успехами каждого обучающегося, процессом формирования компетенций. Текущий контроль успеваемости служит для определения педагогических приемов и методов для индивидуального подхода к каждому обучающемуся, корректировки плана работы с группой.

Формы:

- устные и письменные работы;
- индивидуальный опрос.

Практические задания, домашние работы, учащиеся выполняют в форме устной или письменной речи. Оценка основывается на ясности выражения мыслей и использовании предметных знаний.

Текущий контроль успеваемости служит для определения педагогических приемов и методов для индивидуального подхода к

каждому обучающемуся, корректировки плана работы с группой. Осуществляется в форме наблюдения, тестирования, контрольного опроса (устного или письменного), собеседования, психологического мониторинга.

Варианты примерных заданий.

1. Что такое игровой движок и каковы его основные компоненты?
2. Как создать базовое игровое окно с использованием библиотеки Pygame?
3. Что такое игровой цикл и почему он важен в разработке игр?
4. Каковы основные принципы объектно-ориентированного программирования?
5. Как создать класс и объект в Python? Приведите пример.
6. Что такое инкапсуляция и как она реализуется в Python?
7. В чем разница между наследованием и композицией в ООП?
8. Что такое полиморфизм и как его использовать в разработке игр?
9. Как спроектировать архитектуру приложения "Персональный менеджер задач"?
10. Как работать с графикой в Pygame? Основные методы и функции.
11. Что такое геймдизайн и каковы его основные принципы?
12. Как обрабатывать пользовательский ввод с клавиатуры в Pygame?
13. Как создать систему взаимодействия игровых объектов?
14. Как реализовать систему счета в игре?
15. Как разработать систему здоровья и жизней для игрового персонажа?
16. Что такое анимации в играх и как их создавать в Pygame?
17. Как работает искусственный интеллект в играх? Основные алгоритмы.
18. Что такое генеративный искусственный интеллект и как его использовать в программировании?
19. Как интегрировать AI-инструменты в процесс разработки на Python?
20. Что такое SQLite и чем она отличается от других баз данных?
21. Каковы основные команды языка SQL для работы с данными?

22. Как создать и организовать игровой мир с препятствиями?
23. Как работают коллизии в играх и как их реализовать?
24. Какие графические библиотеки Python вы знаете и в чем их различия?
25. Как создать пользовательский интерфейс для приложения?

Промежуточная аттестация. Проводится в форме теста с самопроверкой.

Итоговая аттестация. Завершает второй модуль, проводится в виде индивидуального итогового тестирования и создание творческого проекта.

Формы отслеживания результатов: наблюдение, тестирование, контрольный опрос (устный или письменный), психологический мониторинг.

Формы фиксации результатов: аналитическая справка, оценочные материалы, результаты психологического мониторинга, отчёт.

Документальной формой подтверждения итогов реализации отдельного курса программы является документ об обучении «Сертификат» (без оценки) установленного Центром «Поиск» образца.

КАДРОВОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

Обеспечение реализации программы, нацеленной на предоставление высокого качества обучения, планируется за счет штата, состоящего из высококвалифицированных специалистов, обладающих определенными компетенциями и выполняющими определенный функционал. Из них:

- учитель информатики высшей квалификационной категории - 2 чел.;
- педагог-психолог высшей квалификационной категории - 1 чел.;
- педагог-организатор высшей квалификационной категории - 1 чел.

МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ

Требования к зданию/помещению

Для реализации программы помещение должно удовлетворять строительным, санитарным и противопожарным нормам.

Учебные кабинеты укомплектованы удобными рабочими местами за ученическими столами в соответствии с ростом обучающихся, состоянием их зрения и слуха.

Кабинеты информатики оборудованы в соответствии с гигиеническими требованиями, предъявляемыми к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы с ними. Используемые цифровые образовательные ресурсы, инструменты учебной деятельности (программные средства) лицензированы для использования во всём учреждении или на необходимом количестве рабочих мест. В работе используются комплекты лицензионного или свободно распространяемого программного обеспечения.

В целях организации антитеррористической защищённости охрана здания учреждения должна быть обеспечена системой наружного видеонаблюдения, пропускным режимом и штатными охранниками. Территория учреждения должна иметь периметральное ограждение и наружное освещение в темное время суток.

Учебно-методическое и информационное обеспечение программы

Аудитории:

- аудитория для теоретических занятий с необходимой ученической мебелью, пластиковой доской;
- компьютерный класс на 12 ученических и 1 учительское место;
- коворкинг-зона.

Технические средства и оборудование:

- проекционное оборудование;
- персональные компьютеры с выходом в сеть интернет и необходимым для стандартного функционирования программным обеспечением;
- обучающие и демонстрационные файлы;
- черно-белый лазерный принтер;
- белая бумага для стандартной печати формата А4;

- маркеры для пластиковой доски;
- сплитсистема.

Лицензионное программное обеспечение:

- Операционная система Linux;
- Среда разработки Visual Studio Code;
- Офисный пакет LibreOffice.

Средства защиты:

- антибактериальные салфетки;
- антибактериальный спрей;
- огнетушитель;
- рециркулятор.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

1. Перечень литературы, необходимой для освоения программы

1.1. Перечень литературы, использованной при написании программы

1. Лутц, М. (2013). «Изучаем Python» (4-е изд.). – Москва: Издательство «Вильямс». – 1648 с. Оригинальное название: Learning Python.
2. Свейгарт, А. (2017). «Создание игр на Python с Pygame». – Москва: Издательство «Вильямс». – 384 с. Оригинальное название: Making Games with Python and Pygame.
3. МакГрат, М. (2020). «Python программирование для начинающих». – Москва: Эксмо. – 192 с. Оригинальное название: Python Programming for Beginners.
4. Роули, К. (2018). «Python и создание игр». – Москва: Бомбора. – 400 с. Оригинальное название: Python Crash Course.
5. Шауэрэм, Д. (2019). «Pygame для разработки игр». – Санкт-Петербург: Питер. – 288 с. Оригинальное название: Game Development with Pygame.
6. Бизли, Д. (2020). «Python. Книга рецептов». – Москва: ДМК Пресс. – 448 с. Оригинальное название: Python Cookbook.

1.2. Перечень литературы, рекомендованной обучающимся

1. Гарленд, М. (2021). «Разработка игр на Python и Pygame. От идеи до реализации». – Санкт-Петербург: Питер. – 352 с.
Оригинальное название: Python Game Development.
2. Миллз, К. (2022). «Объектно-ориентированное программирование на Python для начинающих». – Москва: Эксмо. – 256 с.
3. Чен, В. (2023). «SQLite для разработчиков игр». – Москва: ДМК Пресс. – 320 с. Оригинальное название: SQLite for Game Developers.
4. Патерсон, Д. (2023). «Искусственный интеллект в играх на Python». – Санкт-Петербург: Питер. – 304 с. Оригинальное название: AI for Python Game Development.
5. Харрисон, К. (2022). «Проектирование и архитектура игровых приложений». – Москва: Бомбора. – 416 с.

1.3. Перечень литературы, рекомендованной родителям

1. Кови С. «7 навыков высокоэффективных людей. Мощные инструменты развития личности» – Альпина Паблишер, 2019.
2. Ицхак Пинтусевич «Действуй! 10 заповедей успеха» – изд. Эксмо, 2018 г.
3. Стивен Кови «Восьмой навык. От эффективности к величию» – «Альпина Паблишер», 2020 г.
4. Дэвид Аллен «Как привести дела в порядок. Искусство продуктивности без стресса» – Манн, Иванов и Фербер, 2021 г.
5. Кэрл Дуэк «Гибкое сознание. Новый взгляд на психологию развития взрослых и детей» – Манн, Иванов и Фербер, 2022 г.
2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения программы:
 1. Официальная документация по Python
Сайт: python.org/doc/
Описание: Официальная документация по языку Python на английском языке.
 2. Документация по Pygame
Сайт: pygame.org/docs/
Описание: Полная документация по библиотеке Pygame с примерами и руководствами.
 3. Stepik - Разработка игр на Python
Сайт: stepik.org/course/161139
Описание: Бесплатный курс по созданию игр на Python с использованием Pygame.
 4. Real Python - Game Development

Сайт: realpython.com/tutorials/game-development/

Описание: Статьи и tutorиалы по разработке игр на Python.

5. SQLite Tutorial

Сайт: sqlitetutorial.net

Описание: Руководство по работе с SQLite на английском языке.

6. GeeksforGeeks - Python Game Development

Сайт: geeksforgeeks.org/python-game-development/

Описание: Статьи и примеры кода по разработке игр на Python.

7. Python Programming - Game Development

Сайт: pythonprogramming.net/game-development-tutorials/

Описание: Видеоуроки и статьи по созданию игр на Python.

8. OpenAI API Documentation

Сайт: platform.openai.com/docs

Описание: Документация по использованию генеративного ИИ в приложениях.

9. GitHub - Pygame Projects

Сайт: github.com/topics/pygame

Описание: Открытые проекты и примеры кода на Pygame.

10. Stack Overflow - Pygame

Сайт:

stackoverflow.com/questions/tagged/pygame

Описание: Сообщество разработчиков, где можно задать вопросы по Pygame.

Дополнительные образовательные ресурсы:

1. KidsCanCode - Python Game Programming

Сайт: kidscancode.org/python

Описание: Ресурсы по обучению детей программированию игр на Python.

2. Invent with Python

Сайт: inventwithpython.com

Описание: Бесплатные книги и tutorиалы по созданию игр на Python.

3. CodeCombat

Сайт: codecombat.com

Описание: Интерактивная платформа для обучения программированию через создание игр.

4. Coding Games

Сайт: codingame.com

Описание: Платформа для обучения программированию через игровые задачи.

5. Russian Python Community

Сайт: python.ru

Описание: Русскоязычное сообщество Python-разработчиков с форумом и статьями.