

Департамент образования и науки Курганской области

Государственное бюджетное профессиональное образовательное учреждение
«Курганский технологический колледж
имени Героя Советского Союза Н.Я. Анфиногенова»

Принята на заседании
методического совета ГБПОУ «КТК»
Протокол № _____

УТВЕРЖДЕНА
приказом врио директора ГБПОУ «КТК»,
от _____ № _____

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Python. Создание игр. Модуль 2»

Направленность: техническая
Уровень программы: углубленный
Возраст обучающихся: 13-17 лет
Срок реализации: 32 часа

г. Курган, 2025 г.

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цель обучения: создание условий для получения знаний, умений и навыков в области программирования на языке Python, знакомство с различными парадигмами программирования (процедурной, функциональной и объектно-ориентированной), знакомство с профессиональными инструментами создания 2D-игр для последующей учебной и профессиональной деятельности в области IT-технологий

Задачи обучения:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- формирование навыков процедурного программирования;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-ориентированного программирования;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- знакомство с профессиональными инструментами создания 2D-игр;
- приобретение навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- приобретение навыков использования внешних библиотек при работе с изображениями;
- приобретение навыков поиска информации в интернете, анализ выбранной информации на соответствие запросу, использование информации при решении задач.

Задачи развития:

- развитие у обучающихся интереса к программированию;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- расширение кругозора обучающихся в области программирования;
- профессиональное ориентирование в области IT-технологий.

Задачи воспитания:

- воспитание упорства в достижении результата;
- подготовка к самостоятельной профессиональной деятельности;
- создание условий для социализации и саморазвития личности обучающихся.

Планируемые результаты обучения

По окончании курса ученик приобретает следующие компетенции:

- знает основы языка программирования Python;
- умеет объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умеет искать и обрабатывать ошибки в коде;
- умеет разбивать решение задачи на подзадачи;
- способен писать грамотный, красивый код;
- способен анализировать как свой, так и чужой код;
- понимает основы представления, кодирования, хранения и передачи информации, логических законов построения компьютеров;
- способен работать с информацией: находить, оценивать и использовать

информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);

– способен грамотно строить коммуникацию, исходя из целей и ситуации.

Категории обучающихся по программе: По программе обучаются дети в возрасте 13-17 лет, имеющие навыки программирования. Проводится входное тестирование на знание основ программирования.

Срок реализации программы: Программа рассчитана на 4 месяца

Трудоемкость программы: Программа рассчитана на 32 часа (16 занятий).

Форма обучения: очная

Режим занятий: Занятия по 2 академических часа 1 раз в неделю (академический час – 40 минут)

Форма итоговой аттестации: тестирование и выполнение проектных работ.

Документ, выдаваемый слушателям после освоения программы: сертификат.

Сведения об использовании сетевой формы реализации образовательной программы: не используется.

Сведения о реализации образовательной программы с применением электронного обучения и дистанционных образовательных технологий: не применяются.

3. РАБОЧАЯ ПРОГРАММА

3.1. Учебно-тематический план

Перечень разделов, подразделов и тем,
количество часов и виды учебных занятий

№ пп	Наименование тем (разделов)	Всего часов	В том числе				Форма контроля
			Теорет ически заняти я	Практ и- ческие заняти я	Самос тоятел ьная работ ы	Аттест ация	
1	Планирование проекта игры и архитектура	2		1	1		
2	Игра «Ловкач» (Catch) на PyGame	2		1	1		
3	Интерактивность и управление состояниями	2		1	1		
4	Работа с данными и сохранение	2		1	1		
5	Продвинутое графические эффекты	2		1	1		

6	Реализация продвинутых графических эффектов	2		1	1		
7	Игра «Угадай число» на PyQt + PyGame	2		1	1		
8	Интерактивная визуализация и дебаг	2		1	1		
9	Финализация проекта. Рефакторинг кода	2		1	1	1	
10	Введение в создание собственного проекта	2		1	1	1	
11	Дизайн и графика для игр	2		1	1		
12	Дизайн и графика для игр	2		1	1		
13	Финальный проект: сборка MVP	2		1	1		
14	Финальный проект: доработка и полировка	2		1	1		
15	Финальный проект: доработка и полировка	2		1	1		
16	Защита проектов	2		1	1		Защита проектов
		32		16	16		

3.2. Содержание программы

Занятие 1 : Планирование проекта игры и архитектура

Инструменты разработки и менеджмент

- Разделение проекта на модули: движок, логика игры, UI, сохранение стан.
- Практика: создание каркаса проекта для будущего финального проекта.
- Git, версия контроля, ветвление для командной работы, README и документация.
- Домашнее задание: создать репозиторий для будущего проекта, настроить .gitignore.

Занятие 2: Игра «Ловкач» (Catch) на PyGame

- Ловушки и ловцы: объекты, которые надо поймать или уклоняться.
- Домашнее задание: добавить простую физику движения объектов и очки за пойманные.

Занятие 3: Интерактивность и управление состояниями

- Управление состоянием игры: меню, пауза, игра, окно результатов.
- Пример: реализация окон меню в PyQt и PyGame.
- Домашнее задание: реализовать переход между состояниями.

Занятие 4: Работа с данными и сохранение

- Сохранение прогресса, настройки пользователем: JSON, файл конфигурации.
- Домашнее задание: реализовать сохранение статистики по игре.

Занятие 5: Продвинутые графические эффекты

- Тени, градиенты, анимации переходов.
- Домашнее задание: добавить плавные переходы между меню и игровым полем.

Занятие 6: Реализация продвинутых графических эффектов.

- Реализовать на примере разных игр
- Домашнее задание: добавить анимации игрока и перехода в меню и игровое поле.

Занятие 7: Игра «Угадай число» на PyQt + PyGame

- Комбинирование GUI и графического окна: как работать с несколькими окнами.
- Домашнее задание: собрать мини-проект «Угадай число» с графическим вводом и визуальными подсказками.

Занятие 8: Интерактивная визуализация и дебаг

- Как тестировать игры, писать тесты на логику, использовать логи и окна сообщений.
- Домашнее задание: добавить простую систему логирования в одну из игр.

Занятие 9: Финализация проекта. Рефакторинг кода

- Рефакторинг кода, улучшение структуры проекта.
- Домашнее задание: выбрать одну из реализованных игр и сделать её «готовой к публикации» (полный файл, инструкции).

Занятие 10: Введение в создание собственного проекта

- Обсуждение концепций, целей, жанра и целевой аудитории.
- Домашнее задание: придумать идею финального проекта и написать техническое задание (TER).

Занятие 11, 12: Дизайн и графика для игр

- Принципы UI/UX в играх, использование графических ресурсов, звуковое оформление.
- Домашнее задание: подобрать элементы оформления для финального проекта и сделать стиль.

Занятие 13: Финальный проект: сборка MVP

- Ученики начинают реализовывать минимально жизнеспособный продукт своего проекта.
- Поддержка: консультации, ревью кода, устранение ошибок.

Занятие 14, 15: Финальный проект: доработка и полировка

- Доработка геймплея, исправление багов, добавление документации.
- Подготовка к защите проекта: презентация, демонстрация игрового процесса.

Занятие 16: Защита проектов и обзор курса

- Каждому ученику защита проекта: цели, архитектура, что получилось и что можно улучшить.
- Итоговый обзор: какие навыки освоены, примеры кода, как продолжать учиться самостоятельно.

Дополнительные идеи игр для расширения плана

- Игра «Головоломка-сдвигалка» (пазл-игра, перегон пазла в PyGame).
- Игра-платформер простой на PyGame с передвижением героя и препятствиями.
- Игра-тайм-менеджер: управлять ресурсами за ограниченное время.

- «Игра-лабиринт» с генерацией уровней и простым ИИ для поиска выхода.
- «Рой насекомых»: симуляция колонии насекомых с базовой логикой AI.
- «Музыкальная кнопка»: память мелодии и реакция на ноты.
- «Симулятор магазина» или «Управление налогами» для бизнес-логики и структур данных.

Учебные цели на каждом занятии

- Понимание базовых концепций: переменные, типы данных, циклы, условия.
- Освоение и применение PyQt для оконных приложений и PyGame для графических игр.
- Работа с графикой: загрузка изображений, рисование, анимации.
- Введение в обработку событий: ввод пользователя, мышь, клавиатура.
- Архитектура программ: модули, классы, повторное использование кода, тестирование.
- Работа с внешними данными: сохранение настроек, прогресса в JSON.
- Основы дизайна и разработки финального проекта: планирование, версия контроля, документация.

3.3 Организационно-педагогические условия

Кадровое обеспечение

Работать преподавателями дополнительной образовательной программы могут:

- учителя информатики, математики и других технических и естественно-научных дисциплин;
- преподаватели дополнительного образования технического и естественно-научного направлений;
- студенты профильных вузов.

Материально-техническое обеспечение

Для работы группы дополнительного образования требуются классы с компьютерами для всех учащихся и преподавателя, проектором, маркерной доской, столами и стульями.

Технические требования к компьютерам совпадают с требованиями к компьютерам для проведения уроков информатики.

Педагогические технологии

При реализации программы применяются следующие педагогические технологии:

- информационно-коммуникативные;
- деятельностные;
- проектные с элементами исследовательской деятельности;
- дифференцированные (индивидуальная траектория обучения);
- модульное обучение.

Используются следующие методы и формы преподавания:

- наглядные;
- словесные;
- с применением технических средств;
- практические;
- проблемные

3.3.1. Перечень основной учебной литературы

1. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И.Г. Семакина и Е.К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
3. Прохоренок Н.А., Дронов В.А. Python 3. Самое необходимое. - С-Пб: Изд-во BHV, 2016 г – 462 с.
4. МакГрат М. Программирование на Python для начинающих. – М.: Эксмо, 2015 г. – 192 с.
5. Чан У. Python: создание приложений. Библиотека профессионала, 3-е изд. - М.: ООО "И.Д. Вильямс", 2015 г. - 816 с.
6. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. – М.: Изд-во НОУ "Интуит", 2016 г. – 178 с.

3.3.2. Перечень дополнительной учебной литературы

1. К.Ю. Поляков, Е.А. Еремин. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.
2. Богатов Ф.Г. Практикум по информатике. Word - Excel - Access - Изд-во: Щит-М, 2007 г. – 264 с.
3. Голицына О.Л., Партыка Т.Л., Попов И. Программное обеспечение. Учебное пособие - Изд-во: Форум, 2008 г. – 448 с.
4. Захарова И.Г. Информационные технологии в образовании: Учебное пособие. 7-е изд., перераб.и доп. - Изд-во: ИЦ Академия, 2011 г.
5. Киселев Г.М., Бочкова Р.В. Информационные технологии в педагогическом образовании. – Изд-во: Дашков и К, 2011 г.
6. Лупин С.А., Колдаев В. Архитектура ЭВМ - Изд-во: Форум, 2008 г. – 384 с.
7. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах: Учебное пособие для вузов.– 4-е издание, стереотипное. – Изд-во: Горячая линия-Телеком, 2011 г. – 146 с.
8. Лутц М. Изучаем Python, 4-е издание. – СПб.: Символ-Плюс, 2011. – 1280 с.
9. Свейгарт Э. Автоматизация рутинных задач с помощью Python. - М.: ООО "И.Д. Вильямс", 2016 г. - 574 с.
10. Любанович Б. Простой Python. Современный стиль программирования. – С-Пб: Изд-во Питер, 2016 г. – 480 с.

3.3.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. it.kgsu.ru - Сайт кафедры ИТ и МПИ «Шаг за шагом»
2. http://www.pc-pensioneru.ru/word_uroki.htm Уроки и видеоматериалы по изучению Microsoft Word
3. http://www.urokpk.ru/obuchenie_word.html - Справочник по программе Word
4. <http://komputs.ru/obuchenie-programmam/izuchaem-powerpoint.html> - Пошаговые инструкции по работе с PowerPoint
5. <http://www.edu.ru/> - Федеральный портал «Российское образование»
6. <http://distancionnoeobuchenie.com/> - Общие сведения о дистанционном обучении
7. <https://www.python.org/> - Официальный сайт по языку программирования Python
8. <https://www.python.ru/> - Сайт русскоязычного сообщества по языку программирования Python
9. <http://zope.net.ru/> - Сайт посвящен работе русскоязычной группы пользователей языка программирования Python и web-платформы Zope.

10. <https://ru.wikipedia.org/wiki/Python> - Страница языка Python в Википедии
11. <https://pythonworld.ru/> - Язык программирования Python 3 для начинающих

3.3.4. Методические рекомендации слушателям

Самостоятельная работа обучающихся при освоении курса складывается из изучения лекционного материала и самостоятельной работы при выполнении итоговой работы. Регулярное посещение занятий.

3.3.5. Перечень используемых информационных технологий, программного обеспечения, информационных справочных систем

При чтении лекций используются слайдовые презентации в формате .pdf.

Необходимое программное обеспечение, используемое в учебном процессе:

- ПО Python IDLE для написания программ.

3.3.6. Описание материально-технической базы, необходимой для освоения программы

Мультимедийное оборудование

- Мультимедийный проектор;
- Интерактивная доска
- Персональный компьютер

Компьютерный класс для проведения лабораторных работ

12 персональных компьютеров, оснащенных лицензионным программным обеспечением.