

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Утверждаю

Проректор по УР

_____ Я.Ю. Григорьев

« _____ » _____ 20____ г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ПРАКТИЧЕСКОЙ
НАПРАВЛЕННОСТИ**

«Python для детей»

Уровень освоения: стартовый

Возраст учащихся: 10-14 лет

Срок реализации: 3 месяца

Составители:

Покровский Виктор Владимирович

г. Комсомольск-на-Амуре

2023 г.

1. Комплекс основных характеристик ДООП

1.1 Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Python для детей» имеет практическую направленность, стартовый уровень.

Программа разработана с учетом следующих нормативно-правовых документов:

– Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

– Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (с изменениями на 30 сентября 2020 года).

– Приказ Минобрнауки России от 23.08.2017 N 816 "Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ" (Зарегистрировано в Минюсте России 18.09.2017 N 48226).

– Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденными 28.09.2020 г. № 28 (регистрационный номер 61573 от 18.12.2020 г.)

Актуальность программы в том, Актуальность программы в том, что программирование Python для детей предоставляет возможность развивать в них навыки компьютерной грамотности и логического мышления, адаптироваться к цифровой эпохе, и, что программирование Python для детей предоставляет возможность развивать в них навыки компьютерной грамотности и логического мышления, адаптироваться к цифровой эпохе, и, в конечном итоге, готовить новое поколение к технологическим вызовам и карьерным возможностям. конечном итоге, готовить новое поколение к технологическим вызовам и карьерным возможностям.

Педагогическая целесообразность. Программа программирования на Python для детей обладает педагогической целесообразностью благодаря способности стимулировать интерес и мотивацию детей, развивать их когнитивные навыки, предоставлять индивидуальный подход, поощрять творчество, готовить к будущим вызовам и улучшать образовательный процесс, что делает ее важным инструментом в сфере образования.

Адресат программы: Возраст обучающихся, участвующих в реализации данной программы: от 12 до 18 лет. Программа не требует специальных знаний и подготовки.

Срок реализации программы: 3 месяца.

Объем программы и режим работы: 72 часа.

Таблица 1 - Объем программы и режим работы

| Период | Продолжительность занятия | Кол-во занятий в неделю | Кол-во часов в неделю | Кол-во недель | Кол-во часов в 3 месяца |
|--------------------|---------------------------|-------------------------|-----------------------|---------------|-------------------------|
| 3 месяца | 3 часа | 2 | 6 ч. | 12 | 72 ч. |
| Итого по программе | | | | | 72 ч. |

Режим организации занятий: Занятия в объединении рекомендуется проводить по 3 часа 2 раза в неделю. Занятия проводятся по 45 минут. Между занятиями 10-минутный перерыв.

Форма обучения: очная

Форма организации занятий: основная форма организации занятий – одиночные, практические занятия.

1.1. Цели и задачи программы

Цели программы: Формирование и развитие навыков программирование обучающихся, через создание программ.

Задачи программы:

1. Развивать практические навыки работы с материалами и инструментами, используемые в программировании на Python;
2. Обучать основам разработки программных конструкций, как для простых, так и для сложных приложений.
3. Содействовать освоению знаний в области истории развития программирования, основных теоретических аспектов и практики создания приложений.
4. Формировать общенаучные и технологические навыки в области проектирования и разработки приложений;
5. Ознакомить с правилами безопасной работы в контексте разработки программных приложений;
6. Развивать способность к творческому подходу в решении задач при создании приложений;
7. Усовершенствовать навыки структурирования мыслей в логической последовательности, защиту своей точки зрения, анализ ситуаций и самостоятельное нахождение ответов на вопросы путем логических рассуждений в контексте программирования на Python и создания приложений.

1.2. Учебный план

| № п/п | Название раздела, темы | Количество часов | | | Формы контроля |
|---|--------------------------------|------------------|--------|----------|----------------|
| | | Всего | Теория | Практика | |
| Программирование на Python (72 часа) | | | | | |
| 1. | Вводное занятие | 3 | 1 | 2 | Анкетирование |
| 2. | Генератор случайных историй | 23 | 6 | 17 | |
| 2.1 | Основы синтаксиса Python | 3 | 1 | 2 | Блиц-опрос |
| 2.2 | Переменные и операторы | 3 | 1 | 2 | Тестирование |
| 2.3 | Работа с текстовыми строками | 3 | 1 | 2 | Блиц-опрос |
| 2.4 | Использование списков и циклов | 3 | 1 | 2 | Тестирование |

| | | | | | |
|---------------------|---|-----------|-----------|-----------|---------------------|
| 2.5 | Создание случайных историй из фраз и слов | 3 | 1 | 2 | Блиц-опрос |
| 2.6 | Использование условных операторов для логики в историях | 3 | 1 | 2 | Тестирование |
| 2.7 | Создание функции для повторного использования кода | 1 | - | 1 | Блиц-опрос |
| 2.8 | Разработка и отладка программы | 2 | - | 2 | Тестирование |
| 2.9 | Добавление дополнительных функций и улучшений | 2 | - | 2 | Демонстрация работы |
| 3. | Чат-бот | 23 | 5 | 18 | |
| 3.1 | Обработка ввода пользователя | 3 | 1 | 2 | Тестирование |
| 3.2 | Ответы на заданные вопросы | 3 | 1 | 2 | Блиц-опрос |
| 3.3 | Создание базы данных с вопросами и ответами | 6 | 1 | 5 | Тестирование |
| 3.4 | Поиск и выбор ответов на основе ввода пользователя | 3 | 1 | 2 | Блиц-опрос |
| 3.5 | Интеграция базы данных с ботом | 3 | 1 | 2 | Тестирование |
| 3.6 | Тестирование и улучшение функциональности | 2 | - | 2 | Блиц-опрос |
| 3.7 | Доработка бота, добавление функциональности | 2 | - | 2 | Тестирование |
| 3.8 | Запуск и проверка бота | 1 | - | 1 | Демонстрация работы |
| 4. | Создание игры «Flappy Bird» | 23 | 4 | 19 | |
| 4.1 | Установка Pygame | 2 | 1 | 1 | Тестирование |
| 4.2 | Создание окна игры | 3 | 1 | 2 | Блиц-опрос |
| 4.3 | Основы рисования и движение объектов | 4 | 1 | 3 | Тестирование |
| 4.4 | Рисование персонажа и препятствий | 4 | 1 | 3 | Блиц-опрос |
| 4.5 | Управление персонажем | 2 | - | 2 | Тестирование |
| 4.6 | Обработка столкновений | 3 | - | 3 | Блиц-опрос |
| 4.7 | Организация игрового цикла | 2 | - | 2 | Тестирование |
| 4.8 | Добавление счета и условий победы/поражения | 1 | - | 1 | Блиц-опрос |
| 4.9 | Доработка игры, добавление анимации | 1 | - | 1 | Тестирование |
| 4.10 | Тестирование и демонстрация игры | 1 | - | 1 | Демонстрация работы |
| Итого часов: | | 72 | 16 | 56 | |

1.3. Содержание учебного плана Программирование на Python (72 часа)

1. Вводное занятие (3ч.).

Организационные моменты. Обзор среды разработки, обсуждение тем, объяснение техники безопасности. (1 ч.)

Практическая работа: Создание среды разработки, выбор версии Python и показ проектов (2 ч.)

2. Генератор случайных историй (23ч.).

2.1 Основы синтаксиса Python.

Изучение основ синтаксиса Python (1 ч.)

Практическая работа: Набор синтаксисов Python (2 ч.)

2.2. Переменные и операторы.

Использование переменных и операторов. (1 ч.)

Практическая работа: Написание первых программ с переменными и операторами. (2 ч.)

2.3. Работа с текстовыми строками

Обсуждение, для чего их используют, когда они нужны. (1 ч.)

Практическая работа: Создание строк, методы изменения регистра, чтение и запись текстовых файлов (2ч.)

2.4. Использование списков и циклов.

Обсуждение списков и циклов, что лучше использовать. (1 ч.)

Практическая работа: Использование списков, циклы, применение списков и циклов. (2 ч.)

2.5. Создание случайных историй из фраз и слов.

Использование списков для хранения фраз и слов, как использовать циклы в генерации. (1 ч.)

Практическая работа: Генерация случайных историй с использованием циклов, добавление случайных элементов в истории. (3 ч.)

2.6. Использование условных операторов для логики в историях.

Введение в условные операторы, объяснение концепции условных выражений. (1ч.)

Практическая работа: Применение условных операторов для создания разных вариантов историй в зависимости от выборов пользователя. (2 ч.)

2.7. Создание функции для повторного использование кода.

Практическая работа: Создание пользовательских функций с использованием ключевого слова, возвращение значений из функций. (1 ч.)

2.8. Разработка и отладка программы

Практическая работа: Разработка и создание базовой структуры программы "Генератор случайных историй". (2 ч.)

2.9. Добавление дополнительных функций и улучшений

Практическая работа: Идентификация дополнительных функций, которые могли бы быть полезными для пользователей. (2 ч.)

3. Чат-бот (23 ч.).

3.1. Обработка ввода пользователя.

Примеры обработки текстового ввода от пользователя. (1 ч.)

Практическая работа: Получение ввода от пользователя с использованием функции. (2 ч.)

3.2. Ответы на заданные вопросы.

Введение в концепцию обработки вопросов и предоставления ответов. (1 ч.)

Практическая работа: Основы структуры базы данных для хранения вопросов и соответствующих ответов. (2 ч.)

3.3. Создание базы данных с вопросами и ответами.

Введение в базы данных и их роль в обработке вопросов и ответов. (1 ч.)

Практическая работа: Разработка структуры базы данных для хранения пар вопрос-ответ. Создание таблицы в базе данных и добавление начальных данных. (5 ч.)

3.4. Поиск и выбор ответов на основе ввода пользователя

Введение в поиск и выбор ответов на основе ввода пользователя. (1 ч.)

Практическая работа: Создание алгоритмов для поиска соответствующих ответов. (2 ч.)

3.5. Интеграция базы данных с ботом.

Введение в интеграцию базы данных с чат-ботом. (1 ч.)

Практическая работа: Настройка подключения бота к базе данных. (2 ч.)

3.6. Тестирование и улучшение функциональности.

Практическая работа: Запуск тестов для проверки правильной работы функциональности бота. (2 ч.)

3.7. Доработка бота, добавление функциональности.

Практическая работа: Анализ обратной связи пользователей и выявление областей для улучшения. (2 ч.)

3.8. Запуск и проверка бота.

Практическая работа: Запуск бота и его развертывание на выбранной платформе или сервере. (1 ч.)

4. Создание игры «Flappy Bird» (23 ч.).

4.1. Установка Pygame.

Что такое Pygame, для чего его используют. (1 ч.)

Практическая работа: Установить Pygame, подготовить рабочее место к созданию игры. (1 ч.)

4.2 Создание окна игры.

Назначение окна игры, выбор разрешения окна. (1 ч.)

Практическая работа: Инициализация Pygame, масштабирование окна. (2 ч.)

4.3 Основы рисования и движение объектов

Выбор какие будут персонажи. (1 ч.)

Практическая работа: Бесконечный цикл, управление объектами и определение коллизии. (3 ч.)

4.4 Рисование персонажа и препятствий.

Рисование персонажа, что такое спрайты и их отображения. (1 ч.)

Практическая работа: Создание групп спрайтов, отображение спрайтов. (3 ч.)

4.5 Управление персонажем

Практическая работа: Обработка клавиш и событий. (2 ч.)

4.6 Обработка столкновений

Практическая работа: Определение и обработка столкновений. (3 ч.)

4.7 Организация игрового цикла

Практическая работа: Отрисовка на экране, обновление экрана. (1 ч.)

4.8 Добавление счета и условий победы/поражения

Практическая работа: Создание переменной для счета и отображение счета на экране. (1 ч.)

4.9. Доработка игры, добавление анимации

Практическая работа: Загрузка кадров анимации (1 ч.)

4.10 Тестирование и демонстрация игры

Практическая работа: Тестирование на работоспособность, оптимизация и добавление пасхалки (1 ч.)

1.4. Планируемые результаты

Реализация программы предполагает достижение обучающимися следующих личностных и предметных результатов:

Предметные:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

2. Комплекс организационно – педагогических условий

2.1. Условия реализации программы

- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

2.2. Формы контроля

- **Входящий контроль** осуществляется при комплектовании группы в начале учебного курса. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.
- *Форма контроля:* тестирование.
- **Текущий контроль** осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.
- **Промежуточный контроль** осуществляется в конце каждого месяца курса.
- Форма контроля: тест, решение практических задач средствами языка программирования.
- **Итоговый контроль** осуществляется в конце учебного курса.
- Форма контроля: решение практических задач средствами языка программирования.
- Отслеживание личностного развития учащихся осуществляется методом наблюдения.

2.3 Формы представления результатов

Учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.

2.3. Оценочный материал *Приложение №1*

- Диагностируются результаты различным образом и на различных этапах

деятельности. Для определения уровня знаний и умений обучающихся проводятся: в начале курса - входной контроль, в середине курса - промежуточный контроль, в конце курса — итоговый контроль. Основным способом оценки – самооценка выполнения программ на языке программирования. Эта оценка присутствует на всех этапах. Одной из форм диагностики результатов является зачет или контрольная работа.

- **Входящий контроль:** Тестирование
- **Промежуточный контроль:** Тестирование, решение задач
- **Итоговый контроль:** учащимся предлагается самостоятельно создать игру для тестирования и демонстрация

Система контроля результативности

| Вид контроля | Время проведения | Цель проведения контроля | Формы и средства выявления результата | Формы фиксации предъявления результата |
|------------------------|------------------|---|---------------------------------------|--|
| Стартовая диагностика | Ноябрь | Оценка исходного уровня знаний на первом месяце обучения в начале учебного курса | Опрос | |
| Текущий контроль | Декабрь | Оценка усвоения учащимися содержания конкретной программы (темы, разделы и т.д) | Опрос | |
| Промежуточный контроль | Декабрь | Оценка качества усвоения учащимися содержания конкретной программы по итогам учебного курса (четверть курса) | Тестирование | Проверка работоспособности проектов |
| Итоговый контроль | Январь | Оценка уровня достижения учащихся, в образовательных программах, по окончании курса образовательной программы | Тестирование | Итоговое занятие. Выставка проектов. |

Методическое обеспечение программы.

Методы обучения:

Выбор метода обучения зависит от содержания занятий, уровня подготовки и опыта

обучающихся.

1. Традиционные методы развития интереса обучающихся к техническому творчеству можно объединить в группы, каждая из которых наиболее продуктивна на том или ином этапе работы. 1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Педагогические технологии, применяемые в реализации программы:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

Список источников

Литература для обучающихся:

1. Сайт/справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
 2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
 3. Сайт/Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
 4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.
1. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net , свободный.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.