

**Пояснительная записка**

Информационные технологии (ИТ) – одна из наиболее быстро развивающихся сфер экономики. Происходящие в ней изменения открывают новые и, на первый взгляд, фантастические возможности и в других областях – например, в проектировании, транспортировке, управлении людьми и ресурсами, маркетинге и образовании.

По мере того, как развивается сфера информационных технологий, обществу требуется все больше людей, способных разрабатывать различные ИТ-продукты - программистов. Помимо этого, человек, владеющий навыком программирования, располагает огромными возможностями для самореализации в области его интересов.

Для гармоничного проживания в обществе XXI века человек должен владеть определенными компетенциями. Среди них умение непрерывно учиться новому, использование передовые технологии, работа в команде. Пригодятся также способность к рефлексии и высокий уровень критического мышления.

Всему этому и многому другому дети с любым уровнем подготовки научатся на курсе “Python”. Курс создан в соответствии с компетентностным подходом к обучению.

Курс построен таким образом, чтобы мотивировать учащихся к занятиям программированием и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; научиться общаться с компьютером, который ничего не сможет сделать, если человек не напишет для него соответствующую программу.

Курс нацелен на формирование навыков, которые пригодятся ребенку в будущем: умение анализировать информацию, придумывать новые решения задач, работать в команде, создавать и презентовать свой проект.

Актуальность данной программы заключается в том, что в ней существенно углубляется содержание образовательных школьных программ по информатике и даёт возможность детям освоить прикладные умения по информатике, необходимые в современной жизни.

Программа составлена в соответствии с Федеральным законом от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями); Концепцией развития дополнительного образования детей (утверждена распоряжением Правительства Российской Федерации от 4 сентября 2014 г. № 1726-р); Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден Приказом Министерства образования и науки Российской Федерации от 29 августа 2013 г. № 1008; Письмом Министерства образования и науки Российской Федерации от 11.12.2006 г. № 06-1844 «О примерных требованиях к программам дополнительного образования детей»; Письмом Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)».

**Общая характеристика курса.**

Теоретической основой данной программы являются:

* Компетентностный подход: обучение в системе, развивающей предметные способности. Развитие каждой способности происходит в три шага: формирование учебной мотивации; формирование предметных знаний, умений и навыков; формирование личностных изменений, связанных с усвоением новых знаний. (Хуторской А. В. и др.)
* Личностный подход: обучение в условия развития каждого учащегося как личности (Вербицкий А. А. и др.)

Программа рассчитана на 68 академических часов.

**Цель курса** - научить ребёнка использовать язык программирования Python как инструмент для реализации идей в области своих интересов.

Для этого сначала ученик получает базовые знания (алгоритмы, структурное программирование, использование библиотек в процедурном стиле), затем учится проектной деятельности, после чего реализует проекты в области своих увлечений. При этом базовая часть также подразумевает возможность выразить свои интересы. Проекты выполняются преимущественно индивидуально, чтобы дети сначала научились работать над ними, затем уже - работать в группе. После освоения базовой части можно перейти на второй год.

**Задачи курса:**

* Развитие профессиональных компетенций.
* Развитие коммуникативных способностей и умения работать в команде.
* Развитие навыков планирования и презентации проектов.
* Развитие креативных способностей и умения учиться.
* Формирование мотивации успеха, готовности к действиям в новых условиях и нестандартных ситуациях.

**Особенность курса** заключается в том, что детям не дают образец решения задачи. Ученики сами пытаются составить последовательность действий, ведущую к цели, и запрограммировать её. Полученные знания ребёнок применяет при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

**Возраст детей,** участвующих в реализации данной образовательной программы 11-14 лет.

На курсе возможны следующие **формы организации деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие, конференция.

**Прогнозируемый результат.**

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Ожидаемый результат | Способ достижения | Как понимаем, что результат достигнут и с помощью кого/чего: (У)ченик, (П)реподаватель, (Р)одитель, (К)омпьютер, (Н)апарник |
|  | **Личностные результаты обучения** | | |
| У0 | Мир удивительный | Через разнообразие проектов в самых разных областях - ребёнок видит, что даже в самой неинтересной для него области есть много занимательного. | Интересуется как разными аспектами своей области интересов, так и другими направлениями. |
| У1 | Компьютер - удивительный инструмент, с помощью которого можно добиться многого | Акцентируем внимание детей, что с помощью компьютера они делают вещи, которые без него делали бы месяцами; добавляем научно-популярные вставки про использование компьютеров. | В тестовом задании ставит компьютер на самую большую отметку (К). |
| У2 | Любой рутинный процесс можно автоматизировать | Прежде чем показать лёгкий способ что-то сделать, предлагаем начать это делать вручную, затем показываем, что это рутинный процесс, и его можно автоматизировать. | В сценарном задании предлагает автоматизировать рутинный процесс (К), при выполнении проекта использует алгоритмы для автоматизации рутинных операций (П). |
|  | **Метапредметные навыки** | | |
| М1 | Проблема? Сейчас решу! | Создаём ситуации успеха при решении незнакомых задач: вместо того, чтобы дать пример, затем повторить, сначала даём задачу и через подсказки помогаем дойти до решения. | При возникновении проблемы предлагает способы её решения или понимает, что текущих знаний для решения проблемы не хватает и предлагает способы эти знания получить (Р). |
| М2 | Ошибки - это инструмент обучения | Часть заданий подразумевают, что надо попробовать несколько методов, один из которых заработает, единственно верный вариант не даётся заранее. | Пробует свои идеи, не ожидая разрешения (П).  Задачи, решение которых подразумевает теорию, которой ещё не было, решают с n-ой попытки (К). |
| М3 | Я не боюсь решать задачи, с которыми раньше не сталкивался | Создаём ситуации успеха при решении незнакомых задач: вместо того, чтобы дать пример, затем повторить, сначала даём задачу и через подсказки помогаем дойти до решения. | Сталкиваясь с новой задачей, ищет решение: задает вопросы, проясняющие структуру задачи; находит знакомые подзадачи и аналогии; разделяет составные части задачи. (П) |
| М4 | Умеет давать обратную связь | Ученики презентуют свои проекты, учитель и другие ученики дают обратную связь, учитель на собственном примере показывает, как лучше это делать. | В ходе комментирования работы другого избегает оценочных суждений, выделяет как сильные, так и слабые стороны (П). |
| М5 | Умеет планировать проекты | В начале выполнения проектов ученики зарисовывают его план, учитель показывает, как это лучше делать. | Раскладывает проект на составные элементы, затем выстраивает их в последовательность шагов (К). |
| М6 | Умеет презентовать свою работу | Ученики презентуют свои проекты, учитель и другие ученики дают обратную связь, учитель также даёт советы, каким образом лучше это делать. | Во время презентации своих проектов пользуется вниманием аудитории (П). |
|  | **Предметные навыки** | | |
| П1 | Умеет создавать работающие программы на языке Python с использованием:   * + переменных   + условных операторов   + циклов   + функций   + модулей   + объектов   + классов   + списков   + словарей | Ученик выполняют задания на платформе с постепенным увеличением сложности. | В свободном проекте без подсказки учителя использует соответствующие инструменты (П).  Для решения поставленной проблемы использует нужный инструмент (К). |
| П2 | Умеет выбрать подходящую задаче структуру данных | В начале проекта при планировании ученики в том числе рассказывают, какие структуры данных собираются применить и доказывают, почему это лучшее решение. | Для выполнения проекта без подсказки учителя выбирает оптимальную из известных структуру данных (П). |
| П3 | Пишет программу следуя стандарту оформления кода | Учим писать код так, как это принято, если это не мешает объяснению понятия. | Использует комментарии, следует общепринятым нормам там, где это не мешает обучению (Н). |
| П4 | Тестирует программу и находит в ней ошибки | Даём задания на поиск ошибок, а также ученики проверяют код друг друга. | Без подсказки учителя умеет находить ошибки в своём и чужом коде, построенном на известных командах (К) |
| П5 | Умеет использовать чужой код: анализировать, изменять | Даём задания с работающим кодом, но дающим результат, отличный от того, который нужен. | В заданиях, где есть готовый код, использует части, которые нужны, изменяет части, которые надо изменить, выкидывает части, которые не нужны (К) |
| П6 | Использует итерационный подход | Сначала даём маленькие кусочки программ, потом объединяем. Также учитель на собственном примере показывает, как проходит разработка программы. | Пишет сначала маленькие кусочки, тестирует их, затем объединяет в большие (П). |

**Механизмы отслеживания результатов.**

* Задания на платформе с автопроверкой (все задания на тему базовых понятий проверяются автоматически системой)
* Ревью кода (выборочная оценка кода учеников на соблюдение стандартов его написания)
* Конференции и круглые столы, на которых возможна взаимопроверка.
* Чек-листы для преподавателей, других учеников (большинство метанавыков оценивается через заполнение преподавателями и учениками анкет)
* Анкета родителей (для оценки появления и развития установок)

**Учебно-дидактическое обеспечение.**

1. Поурочные методические указания.
2. Тематические презентации.
3. Рабочие тетради для учеников.
4. Персональные компьютеры по количеству учащихся в группе.
5. Компьютер, доска, проектор для учителя.

**Учебно-тематический план.**

Указана примерная продолжительность каждого модуля, поскольку преподаватель может ускорять и замедлять темп урока для каждой группы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № модуля | Количество академических часов | Темы | Форма проведения | Образовательные результаты | Содержание занятий |
| 1 блок - базовая часть - элементарные конструкции языка Python | | | | | |
| 1 | 8 | Целые числа, переменные, команды ввода-вывода, простые операции со строками | Учебная дискуссия,практические занятия, занятие-проект. | М2, У1, П1, П4, П5 | Задания на платформе и в рабочей тетради на освоение нового материала.  Программа-аватар ученика: приветствует пользователя, интересуется его возрастом, чем увлекается, и рассказывает о том, что она узнала. |
| 2 | 8 | Условный оператор и цикл while. Клавиатурные комбинации. | Учебная дискуссия,практические занятия, занятие-проект. | +П3, П6 | Задания на платформе и в рабочей тетради на освоение нового материала.  Программа-чат-бот в консоли: приветствует пользователя, может загадать ему загадку, поработать калькулятором, решить квадратное уравнение. |
| 3 | 10 | Функции, модули, классы и объекты | Учебная дискуссия,практические занятия, занятие-проект. | +М4 | Задания на платформе и в рабочей тетради на освоение нового материала.  Чат-боты объединяются в сеть: за счёт подключения модулей реализовывается следующая механика: программа ученика А обращается к программе ученика Б и пользователь пытается угадать другую загадку. |
| 2 блок - обучение проектной деятельности, работа в группах | | | | | |
| 4 | 12 | Отработка пройденных тем с помощью модуля turtle. | Учебная дискуссия,практические занятия, занятия-проекты. | +М6, М3, У2 | Задания на платформе и в рабочей тетради на освоение нового материала.  Программа, которая сможет создавать рисунки (узор из звезд на небе, застройку деревни, изображение робота и т.д.)  Пользователь вводит параметры города: сколько в нём домов, деревьев, какая сейчас погода, программа генерирует соответствующий пейзаж. Часть параметров может быть задана случайно или зависеть от ответов пользователя не явно, например, если настроение грустное, то генерируется дождь, а если весёлое, то солнце. |
| 3 блок - выполнение проектов как в группе, так и индивидуально | | | | | |
| 5 | 30 | Выполнение проектов с помощью различных библиотек (сборников подпрограмм или объектов, используемых для разработки программного обеспечения). | Занятия-проекты. | +П2, М5, У1, М1 | Функции индивидуальны для каждой программы, которую создадут дети. Предусматривается выполнение групповых проектов. |

**Содержание программы.**

**Тема 1.1. Python. Знакомство с языком.**

Теория.

Представление языка Python. Понятия «язык программирования», «алгоритм», «программа». Работа с блок-схемами. Знакомство с синтаксисом языка. Понятия «функция», «параметры функции”. Синтаксис написания функций.

Практика.

Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

**Тема 1.2. Переменные.**

Теория.

Понятие данных, переменных, оператора присваивания. Необходимость ввода данных, функция input(), понятие «значение функции».

Практика.

Отработка ввода команд с клавиатуры, свои программы с использованием переменных. Программа-аватар ученика: приветствует пользователя, интересуется его возрастом, чем увлекается, и рассказывает о том, что она узнала.

**Тема 1.3. Выражения. Типы данных.**

Теория.

Понятия «выражение», «приоритет операций», «типы данных». Изменение типов данных с помощью функции int() и str(). Операции со строками и числами. Хранение информации в компьютере.

Практика.

Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

**Тема 2.1. Условный оператор.**

Теория.

Понятия «утверждение», «условие», «условный оператор». Синтаксис условного оператора, уровни вложенности. Операторы сравнения.

Практика.

Выполнение заданий в рабочей тетради на закрепление понятий. Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

**Тема 2.2. Вложенные условные операторы. Клавиатурные комбинации.**

Теория.

Клавиатурные комбинации. Вложенный условный оператор.

Практика.

Написание собственных программ на платформе.

**Тема 2.3. Циклы.**

Теория.

Демонстрация необходимости применения циклов. Обсуждение с учениками, где они могли встречаться в этим понятием. Демонстрация синтаксиса данной конструкции. Разбор примеров программ с использованием цикла. Цикл-счетчик.

Практика.

Решение задач на знакомство с синтаксисом циклов и понимание работы счетчика в тетради. Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

**Тема 2.4. Циклы: продолжение.**

Теория.

Цикл с накоплением результата вычислений в переменной. Цикл с вложенными условными операторами. Целочисленное деление.

Практика.

Отработка ввода команд с клавиатуры, написание собственных программ на платформе.

**Тема 2.5. Проект «Чат-бот».**

Теория.

Обсуждение проекта «Чат-бот» в консоли: приветствует пользователя, может загадать ему загадку, поработать калькулятором, решить квадратное уравнение.

Практика.

Выполнение проекта «Чат-бот» на платформе.

**Тема 3.1. Создание функций.**

Теория.

Изучение понятий «функция», «определение функции», «параметры функции», «значение функции», «вызов функции». Синтаксис создания своих функций.

Практика.

Выполнение заданий в рабочей тетради. Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

**Тема 3.2. Создание модулей. Модуль random.**

Теория.

Изучение понятия «модуль», создание модуля, подключение модуля. Модуль random.

Практика.

Написание собственных программ на платформе. Проект «Угадай число».

**Тема 3.3. Модули: продолжение.**

Теория.

Повторение понятия «модуль», создание модуля, подключение модуля.

Практика.

Проект «Чат-бот» - объединение в сеть. Засчёт подключения модулей реализовывается следующая механика: программа ученика А обращается к программе ученика Б и пользователь пытается угадать другую загадку.

**Тема 3.4. Классы и объекты.**

Теория.

Изучение понятий «класс», «объект». Синтаксис создания классов.

Практика.

Написание собственных программ на платформе.

**Тема 4.1. Модуль turtle. Линейные алгоритмы и переменные.**

Теория.

Повторение понятий «алгоритм», «программа», «язык программирования», «переменная». Изучение понятий «исполнитель», «пиксель», «координаты». Знакомство с модулем turtle: рисование линий, повороты, ширина и цвет пера, перемещение пера в указанную точку.

Практика.

Написание собственных программ на платформе.

**Тема 4.2. Модуль turtle. Классы и функции.**

Теория.

Повторение понятий «функция», «определение функции», «параметры функции», «значение функции», «вызов функции», «объект», «класс». Заливка фигур. Разбор базовых фигур для проекта «Город».

Практика.

Написание собственных программ на платформе. Программирование базовых фигур для проекта «Город».

**Тема 4.3. Модуль turtle. Циклы.**

Теория.

Повторение понятия «цикл». Рисование фигур с помощью цикла.

Практика.

Выполнение заданий в рабочей тетради. Написание собственных программ на платформе. Программирование фигур на основе циклов для проекта «Город».

**Тема 4.4. Модуль turtle. Условный оператор.**

Теория.

Повторение понятия «условный оператор». Логические операторы or, and, not.

Практика.

Написание собственных программ на платформе. Создание своей картины с помощью ранее запрограммированных функций для проекта «Город».

**Тема 4.5. Проект “Город”. Завершение.**

Теория.

Завершение проекта «Город».

Практика.

Создание своей картины с помощью ранее запрограммированных функций для проекта «Город». Демонстрация работ.

Следующая тема рассчитана на 30 академических часов.

**Тема 5. Групповые и индивидуальные проекты.**

Теория.

Проекты с помощью различных библиотек (сборников подпрограмм или объектов, используемых для разработки программного обеспечения). Функции индивидуальны для каждой программы, которую создадут дети. Предусматривается выполнение групповых проектов.

Практика.

Выполнение проектов.

**Методическое обеспечение:**

Для реализации программы используются следующие **методы обучения**:

По **источнику полученных знаний**: словесные, наглядные, практические.

По **способу организации** познавательной деятельности:

Развивающего обучения (проблемный, проектный, творческий, частично-поисковый, исследовательский, программированный)

Дифференцированного обучения (уровневые, индивидуальные задания)

Игровые (конкурсы, игры-конструкторы, турниры с использованием мультимедиа, дидактические)

**Условия реализации рабочей программы**

**Материально-технические условия реализации рабочей программы:**

* Персональный компьютер или ноутбук с предустановленным программным обеспечением (Операционная система Windows/Linux, MS Office или аналог)
* Экран
* Колонки
* Средства телекоммуникации - локальная сеть, выход в Интернет
* Учебно-практическое и учебно-лабораторное оборудование
* Презентации и видео к урокам, дидактический материал.
* Специализированная учебная мебель: компьютерные столы, стулья

**Список литературы.**

1. Хуторской А.В. Компетентностный подход в обучении. Научно-методическое пособие. — М.: Издательство «Эйдос»; Издательство Института образования человека, 2013. — 73 с.

# Вербицкий, Ларионова: Личностный и компетентностный подходы в образовании: проблемы интеграции. Монография. М.: Издательство “Логос”, 2017 - 336 с.