Бюджетное общеобразовательное учреждение города Омска

«Средняя общеобразовательная школа № 132»

Центр цифрового образования детей **«IT-куб»**

|  |  |
| --- | --- |
| Принято на заседании педагогического совета  БОУ г. Омска «Средняя общеобразовательная школа №132»  **Протокол №\_\_\_ от \_\_\_\_\_\_\_\_\_ 2024 г.** | Утверждаю  Директор БОУ г. Омска «Средняя общеобразовательная школа №132»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.А. Мишина  **Приказ №\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_ 2024 г.** |

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ**

**ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Программирование на языке Python»

Направленность: техническая

Возраст обучающихся: 14–17 лет

**Автор: Ванслонович Дмитрий Михайлович,**

**педагог дополнительного образования**

**Омск – 2024**

* 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа «Программирование на языке Python» разработана в соответствии с требованиями нормативных документов:

* + - Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020);

- Федеральный закон от 29.12.2012 г. № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации»;

* + - Федеральный закон РФ от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в Российской Федерации»;
    - Паспорт национального проекта «Образование» (утв. президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16);
    - Стратегия развития воспитания в РФ на период до 2025 года (распоряжение Правительства РФ от 29 мая 2015г. № 996-р);
    - Государственная программа Российской Федерации «Развитие образования» (утв. постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 15.03.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования»;
    - Паспорт федерального проекта «Успех каждого ребёнка» (утверждён на заседании проектного комитета по национальному проекту «Образование» 07 декабря 2018г., протокол № 3);
    - Концепция развития дополнительного образования детей до 2030 года, утверждена распоряжением Правительства РФ от 31 марта 2022 г. № 678-р;
    - Приказ Министерства просвещения Российской Федерации от 09.11.2018 № 196

«Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

* + - Приказ Министерства образования и науки Российской Федерации от 03.09.2019

№ 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

* + - Приказ Министерства образования и науки Российской Федерации от 23.08.2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
    - Приказ Министерства просвещения Российской Федерации от 13.03.2019 «114

«Об утверждении показателей, характеризующих общие критерии оценки качества условий осуществления образовательной деятельности организациями, осуществляющими образовательную деятельность по основным общеобразовательным программам, образовательным программам среднего профессионального образования, основным программам профессионального обучения, дополнительным общеобразовательным программам»;

* + - Приказ министерства образования и науки Российской Федерации и министерства просвещения Российской Федерации от 05.08.2020г. № 882/391 «Об организации и осуществлении образовательной деятельности по сетевой форме реализации образовательных программ»;
    - Профессиональный стандарт «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)» (ред. от 16.06.2019 г.) (приказ Министерства труда и социальной защиты РФ от 18.10.2013 № 544н, с изм., внесёнными приказом Министерства труда и соцзащиты РФ от 25.12.2014 № 1115н и от 05.04.2016 № 422н);
    - Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (приказ Министерства труда и социальной защиты РФ от 05.05.2018 № 298н

«Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»);

* + - Постановление Главного государственного санитарного врача РФ от 28.09.2020 года № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно- эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодёжи»;
    - Постановление Главного государственного санитарного врача РФ от28.01.2021 №

2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685- 21 «Гигиенические нормативы и требования к обеспечению безопасности и(или) безвредности для человека факторов среды обитания»;

* Методические рекомендации по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учётом их особых образовательных потребностей. (Письмо Министерства образования и науки РФ № ВК- 641/09 от 26.03.2016);
* Примерная программа воспитания. Утверждена на заседании Федерального учебно-методического объединения по общему образованию 02.06.2020г.;
* Методические рекомендации по созданию и функционированию центров цифрового образования «IT-куб» (утв. распоряжением Министерства просвещения Российской Федерации от 10.11.2021 № ТВ-1984/04).
* Устав МБОУ — ЛИЦЕЯ, утверждённый администрацией Татарского района Новосибирской области № 306 от 16.12.2015 г.

Содержание программы предусматривает развитие сфер личности ребенка: социально-нравственной, эмоционально - экспрессивной, познавательной, художественно- эстетической, активизации психических процессов, раскрытию творческих способностей.

**Уровень общеобразовательной программы**: начальный

**Направленность программы:** техническая

# Актуальность программы

Актуальность программы «Программирование на языке Python» обусловлена развитием современных и перспективных технологий, что позволяет сегодня компьютерам успешно выполнять такие функции, которые в прошлом были им недоступны или выполнялись другими силами и средствами.

Благодаря росту возможностей и повышению доступности компьютеров, потенциал использования их в разных сферах деятельности стремительно растёт. Данная

образовательная программа позволяет не только обучить созданию программ, но и подготовить обучающихся к планированию и организации работы над разноуровневыми техническими проектами. Это в дальнейшем поможет осуществить осознанный выбор вида деятельности в техническом творчестве.

# Новизна программы

Новизна программы заключается в том, что для организации учебной деятельности используется новейший инструментальный программный функционал, обеспечивающий наиболее полное погружение в современный мир разработки. Это и язык программирования Python, интегрированная среда разработки PyCharm, система управления версиями Git. Всё это позволяет организовать работу обучающихся таким образом, чтобы выполнять поставленные задачи быстро, комфортно и с минимальной потерей концентрации.

# Отличительные особенности программы

Отличительная особенность данной дополнительной программы от существующих образовательных программ в том, что изучается материал, слабо представленный и не представленный в программе основного курса информатики и ИКТ, материал систематизирован, доступно и логично излагается, подкреплен мощным дидактическим материалом, направлен на практику программирования, на развитие творчества и самостоятельности обучающихся. На занятиях создана структура деятельности, создающая условия для творческого развития воспитанников на различных возрастных этапах и предусматривающая их дифференциацию по степени одаренности.

Основные дидактические принципы программы: доступность и наглядность, последовательность и систематичность обучения и воспитания, учет возрастных и индивидуальных особенностей детей. Например, по мере обучения выполняются всё более и более сложные задания, оттачивается мастерство, исправляются ошибки.

Обучаясь по программе, обучающиеся проходят путь от простого к сложному, с учетом возврата к пройденному материалу на новом, более сложном творческом уровне. Программа основывается на доступности материала и построена по принципу «от простого к сложному».

# Адресат программы

Программа адресована детям от 14 до 17 лет, возрастные особенности которых характеризуются тем, что подростки пытаются найти возможность самовыражения, т.е. принять самостоятельное решение, иметь право выразить свою позицию, мнение, взять ответственность на себя. На данном этапе подростки способны логически мыслить, отделять логические операции от объектов, оперировать абстрактными категориями, классифицировать выражения независимо от их содержания, по их логическому типу.

# Объём и срок освоения программы

Объём программы - 72 часа в год. Программа рассчитана на 1 год обучения.

**Форма обучения:** очная.

# Особенности организации образовательного процесса

Формы реализации образовательной программы*:* групповая. Состав группы обучающихся – постоянный.

# Режим занятий, периодичность и продолжительность занятий

Занятия проводятся один раз в неделю продолжительностью 2 академических часа (1 академический час равен 40 минутам). Перерыв между учебными занятиями 10 минут.

**Количество обучающихся в одной группе** 10-12 человек.

* 1. ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ

**Цель:** создание устойчивых навыков использования языка программирования Python для решения прикладных задач.

# Задачи программы:

**Обучающие:**

* сформировать навыки алгоритмического и логического мышления, грамотной разработки эффективных алгоритмов и программ;
* познакомить с принципами и методами структурного программирования;
* сформировать навыки работы в системе программирования Python;
* научить работать с основными структурами данных и типовыми методами обработки этих структур;
* научить работать с основными коллекциями ЯП и типовыми методами их обработки;
* сформировать навыки разработки эффективных алгоритмов и программ с применением основных коллекций ЯП Python;
* сформировать навыки чтения чужого кода.

# Развивающие:

* развитие интереса к техническим знаниям;
* развитие у обучающихся интереса к программированию;
* развитие внимательности при работе с письменными текстами;
* развитие терпения, самоконтроля;
* развитие логического мышления, самостоятельности, познавательного потенциала обучающегося.

# Воспитательные:

* развивать умение грамотно строить коммуникации;
* воспитывать упорство в достижении результата;
* стимулировать интерес к изучению профессии, связанной с программированием.
  1. СОДЕРЖАНИЕ ПРОГРАММЫ Учебный план

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название раздела** | **Количество часов** | |
| **всего** | **из них** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **теория** | **практика** |
| 1 | Введение в программирование.  Линейный алгоритм | 6 | 3 | 3 |
| 2 | Условные  конструкции. Типы данных. Функции. Модули | 16 | 7 | 9 |
| 3 | Циклические конструкции | 10 | 4 | 6 |
| 4 | Решение прикладных задач | 4 | 1 | 3 |
| 5 | Базовые конструкции | 30 | 10 | 20 |
| 6 | Решение прикладных задач | 6 | 2 | 4 |
|  | **Итого** | **72** | **27** | **45** |

Содержание программы

1. Введение в программирование. Линейный алгоритм Задачи:

* Формирование основных понятий система команд, алгоритм, программа, среда разработки, код программы.
* Изучение интерпретатора Python, интегрированной среды разработки PyCharm, знакомство с интерфейсом.

Теория: Знакомство со средой разработки. Создание нового проекта с программным кодом. Запуск и отладка кода программы. Знакомство с системой управления версиями и основными командами. Синтаксис языка. Переменные. Ввод и вывод значений. Комментирование строк. Применение основных математических операторов. Приоритет операций. Операторы присваивания.

Практика: Установка и настройка интерпретатора Python, интегрированной среды разработки PyCharm, знакомство с интерфейсом. Ввод, запуск, отладка учебных примеров. Установка и настройка системы управления версиями Git, регистрация на сайте github.com, создание репозитория, клонирование учебного репозитория. Команды init, clone, add, status, commit, push, branch, checkout. Выполнение лабораторной работы.

1. Условные конструкции. Типы данных. Функции. Модули Задачи:

* Знакомство с основными типами данных языка: целые числа, числа с плавающей точкой, строки.
* Изучение вложенных и каскадных конструкций.

Теория: Знакомство с основными типами данных языка: целые числа, числа с плавающей точкой, строки. Проверка вхождения подстроки в строку. Длина строки. Приёмы написания программ с использованием функций. Стандартная библиотека Python, знакомство с модулями. Функции модуля Random и datetime. Разбор конструкции if-elif- else. Операторы тождественности. Логические операторы для построения условный конструкций. Вложенные и каскадные конструкции.

Практика: Решение задач на явное и неявное преобразование типов данных: с использованием сложных условий, функций и модулей. Выполнение лабораторной работы.

1. Циклические конструкции Задачи:

* Изучение циклических алгоритмических конструкций, в т.ч. вложенных.
* Изучение операторов прерывания и продолжения циклов.

Теория: Знакомство с циклическими конструкциями: циклы с предусловиями While и цикл со счётчиком For. Оператор else в циклических конструкциях. Вложенные циклы. Прерывания и продолжения циклов (операторы break и continue).

Практика: Решение задач с использованием циклов. Выполнение лабораторной работы по циклическим конструкциям.

1. Решение прикладных задач Задачи:

* Отработка навыков решения задач.

Теория: Постановка задачи, выбор темы мини-проекта, обсуждение путей возможного решения.

Практика: Практическая работа над мини-проектом. Демонстрация результатов.

1. Базовые конструкции Задачи:

* Формирование представлений обучающихся о строках как коллекциях и их индексации.
* Знакомство со множествами.

Теория: Продолжение работы со строками. Индексация. Срезы. Функции и методы работы со строками (find(), index(), replace(), isdigit(), isalpha(), islower() и т.д.). Знакомство со списками. Способы создания списка. Индексация. Функции и методы работы со списками (append(), extend(),insert(), count(), max(), min(), sort() и т.д.). Вложенные списки. Списочные выражения. Методы split() (разбиение строки на список строк по произвольному разделителю) и join (сбор строки из списка строк через заданный разделитель). Множества. Отличия множества от списка. Методы для работы с множествами (add(), remove(), copy() и т.д.). Теория множеств. Операции над множествами. Неупорядоченные коллекции пар «Ключ-Значение» (словари). Методы для работы со словарями (keys(), values(), items(), get() и т.д.).

Практика: Практические и лабораторные работы по строкам, спискам, множествам и словарям.

1. Решение прикладных задач Задачи:

* Отработка навыков решения задач.

Теория: Постановка задачи, выбор темы итогового проекта, обсуждение путей возможного решения. Формирование рабочих групп. Оформление готовых решений.

Практика: Практическая работа над итоговым проектом. Демонстрация результатов.

# ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Результативность и способы оценки программы построены на основе компетентностного подхода.

# Предметные результаты:

*Обучающиеся будут знать:*

* понятия «алгоритм», «программа»;
* основные конструкции языка программирования Python (ветвления if, операторы цикла while, for, вспомогательные алгоритмы, функции, модули);

*-* структуру данных языка программирования Python;

*Обучающиеся будут уметь:*

- строить циклические конструкции при работе с коллекциями

**-** владеть основными приёмами составления программ на языке программирования

Python

задач

# Метапредметные результаты:

* формирование умения ориентироваться в системе знаний;
* формирование умения выбирать наиболее эффективные способы решения
* на компьютере в зависимости от конкретных условий;
* формирование приёмов проектной деятельности, включая умения видеть

проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы;

* + формирование умения распределять время;
  + формирование умений успешной самопрезентации.

# Личностные результаты:

* + формирование умения самостоятельной деятельности;
  + формирование умения работать в команде;
  + формирование коммуникативных навыков;
  + формирование навыков анализа и самоанализа;
  + формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

1. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ
   1. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Период обучения 36 недель, 1 раз в неделю. Всего занятий – 72

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п занятия** | **Тема занятия** | **Форма занятия** | **Количество часов** | **Форма контроля** |
| **Введение в программирование. Линейный алгоритм — 6 часов** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Введение в программирование на языке Python. Установка и настройка среды  программирования. Простейшая программа. Команда input() и print(). Параметры sep и end | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 2 | Установка и настройка системы управления версиями. Комментарии. PEP 8 | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 3 | Переменные. Целые числа. Приоритет операций. Целочисленная арифметика | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| **Условные конструкции. Типы данных. Функции. Модули — 16 часов** | | | | |
| 4 | Типы данных int, float, str. Определение длины строки. Вхождение подстроки в строке | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 5 | Условный оператор if-else. Построение сложных условий с помощью логических операций and, or, not | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 6 | Построение сложных условий с помощью логических операций and, or, not | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 7 | Вложенный условный оператор if. Каскадный оператор elif | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 8 | Вспомогательные алгоритмы. Функции | КЗ | 4 | Опрос, проблемная беседа,  самостоятельная работа |
| 9 | Создание, подключение, использование модулей | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 10 | Решение задач по теме  «Условные конструкции» | П | 2 | Самостоятельная работа,  лабораторная работа |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Циклические конструкции — 10 часов** | | | | |
| 11 | Цикл со счётчиком For. Диапазоны для итератора | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 12 | Цикл с предусловием While. Сигнал остановки. Бесконечные циклы | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 13 | Решение задач по теме  «Циклические конструкции» | П | 2 | Самостоятельная работа,  лабораторная работа. |
| 14 | Прерывания и продолжения циклов (операторы break и continue) | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 15 | Вложенные циклические конструкции. Оператор break и continue во вложенных циклах | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| **Решение прикладных задач — 4 часа** | | | | |
| 16 | Мини-проект. Защита мини-  проекта | ПР | 4 | Проектная  деятельность |
| **Базовые конструкции — 30 часов** | | | | |
| 17 | Строки. Индексация. Перебор элементов строки. Срезы строк | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 18 | Функции и методы (find(), index(), replace(), isdigit(),  isalpha(), islower() и т.д.) | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 19 | Решение задач по теме  «Строки» | П | 2 | Самостоятельная работа,  лабораторная работа |
| 20 | Списки. Основы работы со списками: создание списков, индексация, перебор элементов в цикле, вывод списка | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 21 | Функции и методы списков (append(), extend(),max(), min(),  insert(), count(), sort() и т.д.) | КЗ | 2 | Опрос, проблемная беседа, |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | самостоятельная работа |
| 22 | Списочные выражения. Методы split() и join() | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 23 | Вложенные списки. Создание, перебор и вывод элементов  двумерного списка | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 24 | Решение задач по теме  «Списки» | П | 2 | Самостоятельная работа,  лабораторная работа |
| 25 | Множества. Методы для работы со множествами (add(), remove(), copy() и т.д.) | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 26 | Теория множеств. Операции над множествами | КЗ | 2 | Опрос, проблемная беседа,  самостоятельная работа |
| 27 | Решение задач по теме  «Множества» | КЗ | 2 | Самостоятельная работа,  лабораторная работа |
| 28 | Урок-практикум | П | 2 | Практическая  работа |
| 29 | Словари | КЗ | 2 | Самостоятельная работа,  лабораторная работа |
| 30 | Методы работы со словарями (keys(), values(), items(), get() и т.д.) | КЗ | 2 | Самостоятельная работа,  лабораторная работа |
| 31 | Решение задач по темам  «Списки», «Множества»,  «Словари» | П | 2 | Самостоятельная работа,  лабораторная работа |
| **Решение прикладных задач — 6 часов** | | | | |
| 32 | Итоговый проект. Защита  проекта | ПР | 6 | Проектная  деятельность |

Календарный учебный график заполнен с помощью условных обозначений: КЗ — комбинированные занятия, сочетающие элементы теории и практики;

П — проверочное занятие;

ПР — проектная работа (работа над кейсами).

* 1. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ Материально-техническое обеспечение программы:
     + Компьютерное оборудование:
     + Стационарные персональные компьютеры (системный блок, монитор, клавиатура USB, мышь USB) с доступом в интернет - 12 шт.(ноутбуки)
     + интерактивная доска, проектор.
     + Интерпретатор Python v.3.10.x. https:/[/www.p](http://www.python.org/)y[thon.org/](http://www.python.org/)
     + Интегрированная среда разработки PyCharm. https:/[/www.jetbr](http://www.jetbrains.com/)a[ins.com/](http://www.jetbrains.com/)
     + Руководство по PyCharm. https://coderlessons.com/tutorials/python- technologies/uznaite-pycharm/pycharm-kratkoe-rukovodstvo
     + Система управления версиями Git. https://git-scm.com/
     + Система для работы с удалёнными репозиториями GitHub. https://github.com/
     + Тренажёр по работе с Git LearnGitBranching. https://learngitbranching.js.org
     + Электронный учебник для изучения Git Pro Git. https://git-scm.com/book/ru/v2
     + Мобильное приложение для работы с GitHub. https://github.com/mobile
     + Текстовый редактор Sublime Text. https:/[/www.sublim](http://www.sublimetext.com/)e[text.com/](http://www.sublimetext.com/)
     + Тренажёр по работе с Python Питонтьютор. https://pythontutor.ru/
     + Электронный учебник для изучения Python. https://pymanual.github.io/
     + Мобильное приложение для изучения языков программирования SoloLearn. https://sololearn.onelink.me/MfgO/gqu5mku2

# Кадровое обеспечение

1. Требования к кадровому обеспечению деятельности Центра «IT-куб» определяются образовательной организацией самостоятельно с учетом действующего трудового законодательства.
2. Образовательную деятельность по дополнительным общеобразовательным программам на базе Центра «IT-куб» осуществляют педагоги дополнительного образования. В соответствии с пунктом 4 статьи 46 Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» к занятию педагогической деятельностью по дополнительным общеобразовательным программам допускаются лица, обучающиеся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ, и успешно прошедшие промежуточную аттестацию не менее чем за два года обучения. Соответствие образовательной программы высшего образования направленности дополнительной общеобразовательной программы определяется образовательной организацией.
   1. ФОРМЫ АТТЕСТАЦИИ

Формы оценки уровня достижений обучающегося.

Для контроля и самоконтроля за эффективностью обучения применяются методы:

* + - предварительные (анкетирование, наблюдение);
    - текущие (наблюдение);
    - тематические (контрольные вопросы, тесты, промежуточные задания);
    - итоговые (проект).

Формы фиксации образовательных результатов.

Для фиксации образовательных результатов в рамках курса используются:

* + - пакет промежуточных учебных работ, выполненных обучающимися;
    - отзывы обучающихся по итогам занятий и итогам обучения. Формы предъявления и демонстрации образовательных результатов:
    - защита проекта.

Формы подведения итогов реализации программы:

* + - педагогическое наблюдение;
    - педагогический анализ выполнения обучающимися учебных заданий;
    - активность обучающихся на занятиях.
  1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценивание развития учащихся проводится на основе следующего перечня компетенций:

# Технические:

* анализ данных;
* алгоритмическое мышление;
* программирование и написание кода.

# Гибкие:

* критическое мышление;
* работа в коллективе, эффективная коммуникация;
* презентация проекта с точки зрения социального воздействия.

Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

* на каждом занятии: опрос, программирование и написание кода, самоконтроль ученика;
* на уроках-практикумах: тест, программирование и написание кода в интерпретаторе языка программирования Python, взаимоконтроль учеников, самоконтроль ученика;
* при выполнении проектов: программирование и написание кода, презентация проекта.

# Критерии оценивания выполнения практических заданий

|  |  |  |  |
| --- | --- | --- | --- |
| **Оцениваемый результат** | **Недостаточный уровень** | **Средний уровень** | **Высокий уровень** |
| Владение навыками алгоритмического мышления и понимание  необходимости формального  описания алгоритмов | Обучающиеся не способны  определить подходящую алгоритмическую конструкцию для формального описания алгоритма  решения | Обучающиеся  способны определить подходящую алгоритмическую конструкцию для формального описания алгоритма решения  практической задачи | Обучающиеся способны самостоятельно определить подходящую алгоритмическую  конструкцию для формального  описания алгоритма |

|  |  |  |  |
| --- | --- | --- | --- |
|  | практической задачи | при помощи  преподавателя | решения  практической задачи |
| Навыки применения таких структур данных, как число, текст | Обучающиеся не способны определить разницу между указанными типами данных, смысл их использования при решении конкретных задач | Обучающиеся способны  определять при помощи преподавателя типы данных,  необходимых для реализации алгоритма решения  задачи | Обучающиеся способны самостоятельно  определять типы данных,  необходимых для реализации алгоритма решения задачи |
| Навыки построения условных конструкций | Обучающиеся не способны описать, как работает ветвление, его предназначение.  Обучающийся не может самостоятельно использовать условные конструкции для решения  практической задачи. | Обучающиеся способны описать, как работает ветвление, его предназначение.  Обучающийся может с помощью  преподавателя использовать условные конструкции для решения  практической задачи. | Обучающийся может самостоятельно использовать условные конструкции для решения практической задачи, определить  работоспособную  форму построения ветвления. |
| Навыки построения циклических конструкций | Обучающиеся не способны описать, как работает цикл, его предназначение. Обучающийся не может самостоятельно использовать циклические конструкции для решения  практической задачи. | Обучающиеся способны описать, как работает цикл, его предназначение. Обучающийся может с помощью  преподавателя использовать циклические конструкции для решения  практической задачи. | Обучающийся может самостоятельно использовать циклические конструкции для решения практической задачи, определить  работоспособную  форму построения цикла. |
| Навыки разработки программ и их отладки | Обучающийся не может самостоятельно применить пошаговую отладку, найти места ошибок в программном коде,  внести корректные | Обучающийся может самостоятельно применить пошаговую отладку, найти места большинства ошибок в программном коде.  С помощью | Обучающийся может самостоятельно написать  программный код, применить пошаговую отладку, найти места ошибок,  внести корректные |

|  |  |  |  |
| --- | --- | --- | --- |
|  | изменения и разработать корректную программу. | преподавателя может внести корректные изменения в программный код и разработать корректную  программу. | изменения в программный код. |
| Навыки чтения чужого кода | Обучающийся не может самостоятельно изучить код программы, понять логику разработки алгоритма, методы, применённые в программе, выявить места ошибок,  внести изменения для корректировки программы. | Обучающийся может самостоятельно изучить код программы и выявить методы,  применённые в  программе. С помощью преподавателя  понять логику разработки алгоритма, выявить места ошибок, внести изменения для корректировки  программы. | Обучающийся может самостоятельно изучить код программы, понять логику разработки алгоритма, методы, применённые в программе, выявить места ошибок, внести изменения для корректировки программы,  предложить способы её совершенствования. |
| Выполнение мини- проекта. | Обучающийся не может запрограммировать мини-проект без примера кода. | Обучающийся может запрограммировать мини-проект с минимальными подсказками и примерами кода. | Обучающийся может запрограммировать мини-проект со множеством  дополнительных опций и функций с минимальными подсказками и  примерами кода. |
| Навыки коммуникации и презентации | Недостаточная уверенность, аргументация позиций | Уверенность во время выступления, хороший стиль речи, аргументированность и убедительность. Хорошая визуализация защиты | Уверенность во время выступления, отличный стиль речи, высокая  убедительность и аргументированность  . Качественная  визуализация защиты |
| Навыки применения таких структур данных, как список, множество, словарь | Обучающиеся не способны определить разницу между указанными типами  данных, смысл их | Обучающиеся Способны  определять при  помощи преподавателя типы | Обучающиеся способны самостоятельно  определять типы  данных, |

|  |  |  |  |
| --- | --- | --- | --- |
|  | использования при решении конкретных задач | данных,  необходимых для реализации алгоритма решения  задачи | необходимых для реализации алгоритма решения задачи |
| Навыки применения доступные для конкретных данных методы и функции | Обучающиеся не способны определить разницу между функциями и методами, применимыми к данным, используемым для реализации алгоритма | Обучающиеся способны с помощью преподавателя  определить и правильно реализовать  функции и методы, применимые к данным, используемым для реализации алгоритма | Обучающиеся способны самостоятельно с помощью преподавателя  определить и правильно реализовать функции и методы,  применимые к данным, используемым для реализации  алгоритма |
| Навыки применения условных и циклических конструкций при работе с  коллекциями | Обучающиеся не способны самостоятельно применить условные и циклические алгоритмические конструкции для обработки коллекций, необходимых для решения практической задачи | Обучающиеся способны самостоятельно определить условные и циклические алгоритмические конструкции для обработки коллекций, необходимых для решения практической задачи. С помощью преподавателя применить их при написании  программы | Обучающиеся способны самостоятельно применить условные и циклические алгоритмические конструкции для обработки коллекций, необходимых для решения практической задачи |
| Выполнение итогового проекта | Обучающийся не может написать программу итогового проекта без примера кода. | Обучающийся может написать программу итогового проекта с минимальными подсказками преподавателя и примерами  фрагмента кода. | Обучающийся может самостоятельно написать программу итогового проекта и встроить в него дополнительные возможности, расширяющие  удобство и |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | безошибочность  работы. |

Для оценивания итогового проекта заполняется таблица с критериями, за каждый из которых дается определенное количество баллов. Основные критерии, по которым выставляются баллы:

1. соответствие проекта заданию (0-2 балла);
2. творческий подход (0-3 баллов);
3. сложность проекта (0-5 баллов);
4. качество алгоритмов (0-10 баллов);
5. отсутствие ошибок в проекте (0-5 баллов);
6. качество презентации — содержательность, логичность, креативность представления проекта (0-5 баллов).

Баллы суммируются, и на основании этого делается заключение об уровне сложности и успешности выполненного проекта.

Общая сумма:

14 баллов и меньше — низкий уровень освоения программы; 15-23 баллов — базовый уровень освоения программы; 24-30 баллов — высокий уровень освоения программы. Результаты итогового контроля заносятся в таблицу (приложение 1).

* 1. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

*По источнику передачи и восприятия информации:*

* словесный: рассказ, беседа, лекция;
* наглядный: опыт, иллюстрация, дидактический, наглядный материал,образцы;
* практический: показ, постановка опытов;

*по характеру деятельности:*

* объяснительно-иллюстративный (рассказ, показ, лекция, фильм,карточки и т.п.);
* репродуктивный (воспроизведение, действие по алгоритму);
* проблемный (постановка проблемных вопросов, созданиепроблемных ситуаций);
* исследовательский метод (опыты, лабораторные, эксперименты, опытническая работа);
* проектный метод (разработка проектов, моделирование ситуаций, создание творческих работ);
* метод игры (игры дидактические, развивающие, ролевые, деловые)

1. СПИСОК ЛИТЕРАТУРЫ

# Список литературы для педагога:

1. Поляков К.Ю. Программирование. Python. C++. В 4 ч.: учебное пособие/К.Ю.Пляков. – М.:Бином. Лаборатория знаний, 2019.
2. Прохоренок Н. А., Дронов В.А. Python 3. Самое необходимое – СПб: БХВ-Петербург, 2019.
3. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб- приложения //Пер. с англ.: учебник/Э. Мэтиз. 2-е изд. – Спб.: Питер. – 2018.
4. Лутц М. Python. Карманный справочник. 5-е изд.: Пер. с англ. //М.: ИД Вильямс. – 2015.
5. Бизли Д.М., Г. Ван Россум. Язык программирования Python. Справочник. (пер. с англ.) Киев: ДиаСофт., 2000.
6. Чаплыгин А. Н. Учимся программировать вместе с Питоном. Revision: 226.
7. Васильев Денис Алексеевич Методические особенности изучения языка Python школьниками // Символ науки. 2017. №1. URL: [https://cyberleninka.ru/article/n/metodicheskie-osobennosti-izucheniya-yazyka-python-](https://cyberleninka.ru/article/n/metodicheskie-osobennosti-izucheniya-yazyka-python-shkolnikami) [shkolnikami](https://cyberleninka.ru/article/n/metodicheskie-osobennosti-izucheniya-yazyka-python-shkolnikami) (дата обращения: 15.01.2019).
8. Бухаров Т. А., Нафикова А. Р., Мигранова Е. А. ОБЗОР ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON И ЕГО БИБЛИОТЕК // Colloquium-journal. 2019. №3-

1 (27). URL: [https://cyberleninka.ru/article/n/obzor-yazyka-programmirovaniya-python-i-ego-](https://cyberleninka.ru/article/n/obzor-yazyka-programmirovaniya-python-i-ego-bibliotek) [bibliotek](https://cyberleninka.ru/article/n/obzor-yazyka-programmirovaniya-python-i-ego-bibliotek) (дата обращения: 15.01.2019).

# Список литературы для обучающихся:

1. ПИТОНТЬЮТОР URL: [https://pythontutor.ru](https://pythontutor.ru/) (дата обращения: 15.01.2019).
2. Python: основы и применение // Stepik URL: <https://stepik.org/course/512/> (дата обращения: 15.01.2019).
3. Python. Быстрый старт // GeekBrains: Обучающий IT-портал URL: <https://geekbrains.ru/courses/105>(дата обращения: 15.01.2019)

# Приложение 1

**Оценка проектов по программе**

«PYTHON - ЮНИОРЫ»

При оценке проектов учитываются следующие критерии:

* 1. соответствие проекта заданию (0-2 балла);
  2. творческий подход (0-3 баллов);
  3. сложность проекта (0-5 баллов);
  4. качество алгоритмов (0-10 баллов);
  5. отсутствие ошибок в проекте (0-5 баллов);
  6. качество презентации — содержательность, логичность, креативность представления проекта (0-5 баллов).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | ФИО  обучающегося | Соответствие работы  заданию (0-2 баллов) | Творческий подход  (0-3 баллов) | Сложность проекта  (0-5  баллов) | Качество алгоритма (0-10  баллов) | Отсутствие ошибок в программе (0-5  баллов) | Качество презентации (0-5  баллов) |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |