

Министерство образования и науки Алтайского края

Комитет по образованию администрации Топчихинского района

Муниципальное казённое общеобразовательное учреждение

Чистюнькая средняя общеобразовательная школа

Дополнительная общеобразовательная

(общеразвивающая) программа

Технической напрвленности

**«Основы языка программирования Python»**

Возраст учащихся: 12 - 14 лет

Срок реализации: 1 год

Автор - составитель:

Дрешпан И.В.,

педагог дополнительного образования

с.Чистюнька -2024 г.

**Оглавление**

|  |  |  |
| --- | --- | --- |
| 1. | Комплекс основных характеристик дополнительной общеобразовательной (общеразвивающей) программы | 3 |
|  1.1. | Пояснительная записка | 3 |
|  1.2. | Цель, задачи, ожидаемые результаты | 5 |
|  1.3. | Содержание программы | 6 |
|  1.4. | Планируемые результаты | 9 |
| 2. | Комплекс организационно педагогических условий | 11 |
|  2.1. | Календарный учебный график | 11 |
|  2.2. | Условия реализации программы | 11 |
|  2.3. | Формы аттестации | 11 |
|  2.4. | Оценочные материалы | 12 |
|  2.5. | Методические материалы | 13 |
|  2.6. | Список литературы | 15 |

**1.Комплекс основных характеристик дополнительной общеобразовательной(общеразвивающей) программы**

**1.1. Пояснительная записка**

**Нормативные правовые основы разработки ДООП:**

* Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ».
* Концепция развития дополнительного образования детей до 2030 года (Распоряжение Правительства РФ от 31.03.2022 г. № 678-р).
* Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
* Приказ Министерства просвещения России от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
* Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)».
* Приказ Главного управления образования и молодежной политики Алтайского края от 19.03.2015 № 535 «Об утверждении методических рекомендаций по разработке дополнительных общеобразовательных (общеразвивающих) программ».
* Устав МКОУ Чистюньской СОШ.
* Основная образовательная программа МКОУ Чистюньской СОШ.

**Актуальность:**

Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире.

**Новизна:**

 Заключается в том, что на занятиях учащиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python.

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python. На втором году обучения закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

**Целесообразность:**

Научившись программировать на языке Python, учащиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования. Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а так же они являются фундаментом для дальнейшего совершенствования мастерства программирования.

**Вид программы:** модифицированная

**Направленность программы:** техническая.

**Обучение включает в себя следующие основные предметы:** информатика, математика.

**Язык***,* на котором осуществляется образовательная деятельность – государственный язык Российской Федерации – русский.

**Адресат программы:**

Данная дополнительная общеобразовательная программа предназначена для детей 12-14 лет. Курс служит средством внутри профильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации. Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

**Срок и объем освоения программы:**

Срок реализации программы « Основы языка программирования Python» составляет года (36 недель, 36 часов). Период обучения: сентябрь - май.

**Форма обучения:** очная

**Особенности организации образовательной деятельности:** разновозрастные группы

**Сведения об обеспечении образовательных прав и обязанностей обучающихся:**

* обучающиеся имеют право выполнять индивидуальный учебный план, в том числе посещать предусмотренные учебным планом или индивидуальным учебным планом учебные занятия, осуществлять самостоятельную подготовку к занятиям, выполнять задания, данные педагогом в рамках программы;
* дети с ОВЗ имеют право обучаться по дополнительным общеобразовательным программам с учетом особенностей психофизического развития.

**Режим занятий:**

Занятия проводятся в группах постоянного состава до 15 человек. Для каждой группы занятия проводятся 1 раз в неделю по 1 учебному часу. Продолжительность учебного часа – 30-40 минут (в зависимости от возрастной категории

**1.2. Цель, задачи, ожидаемые результаты**

**Цель:** развитие творческой личности, обогащенной общетехническими знаниями и умениями, развитие индивидуальных творческих способностей, интереса к науке и технике.

**Задачи:**

**Предметные:**

* формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
* приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
* формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройстве.

**Личностные:**

* формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
* формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно- исследовательской, творческой и других видах деятельности.

**Метапредметные:**

* умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
* умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
* умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
* владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
* умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласовании позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение.

**1.3. Содержание программы**

**«Основы языка программирования Python »**- 1 год обучения

Учебный план

|  |  |  |  |
| --- | --- | --- | --- |
| №п/п | Название раздела | Количествочасов | Формы аттестации /контроля |
| Всего | Теория | Практика |
| 1 | Раздел 1. Введение в Python | 2 | 1,5 | 0,5 | Собеседование, практическая работа |
| 2 | Раздел 2. Типы данных и операции | 6 | 1,5 | 4,5 | Практическая работа. Тест, практическая работа (промежуточная аттестация) |
| 3 | Раздел 3. Инструкции и синтаксис | 9 | 2,5 | 6,5 | Практическая работа. Тест, практическая работа(промежуточная аттестация).  |
| 4 | Раздел 4. [Функции и модули в программировании](https://younglinux.info/python/function.php) | 9 | 2 | 7 | Тест по модулю |
| 5 | Раздел 5. Сложные типы данных | 7 | 4 | 3 | Решение задач с использованием строк |
| 6 | Раздел 6. Написание и отладка программ. Самостоятельная работа | 3 | 0 | 3 | Защита проекта |
|  |  | 36 | 20 | 16 |  |

Содержание учебного плана

**Раздел 1. Введение в Python.**

**Теория**

Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

**Практика**

Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

**Раздел 2. Типы данных и операции.**

**Теория**

Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math)

**Практика**

 Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.Решение математических задач с использованием функции import math.

**Раздел 3. Инструкции и синтаксис.**

**Теория**

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

**Практика**

Практическое закрепление знаний по условным операторам. Создание программ. Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

**Раздел 4.** [**Функции и модули в программировании**](https://younglinux.info/python/function.php)**.**

**Теория**

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

**Практика**

Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций. Написание программ. Работа со строками.

**Раздел 5. Сложные типы данных.**

**Теория**

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач. Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python.

Установка Python. Доступ к документации. Вввод и вывод данных. Первая программа на языке Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

**Практика**

 Написание программ. Работа со списками. Создание программ. Практическое закрепление полученных знаний.

Создание программ. Практическое закрепление полученных знаний.

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Создание программ. Практическое закрепление полученных знаний.

Создание программ. Практическое закрепление полученных знаний.

**Раздел 6. Написание и отладка программ. Самостоятельная работа.**

**Практика**

Написание программ. Работа со списками. Создание программ. Практическое закрепление полученных знаний.

**1.4. Планируемые результаты**

|  |  |  |
| --- | --- | --- |
| **Результат (освоенные компетенции)** | **Основные показатели оценки результата** | **Формы и методы контроля и оценки** |
| Личностные компетенции | умение работать в команде: работа в общем ритме, эффективное распределение задач и др. | проектная деятельность в команде, презентации и защиты проектов |
| формирование высокого познавательного интереса учащихся | проектная деятельность |
| формирование критического мышления | проектная деятельность |
| проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности | проектная деятельность, выполнение кейсов |
| Метапредметные компетенции | умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений | проектная деятельность, презентации и защиты проектов, выполнение кейсов |
| способность творчески решать технические задачи | выполнение кейсов |
| готовность и способность применения теоретических знаний по физике, информатике для решения задач в реальном мире | проектная деятельность, выполнение кейсов |
| способность правильно организовывать рабочее место и время для достижения поставленных целей | выполнение практических заданий  |
| Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей | выполнение практических заданий  |
| Предметные компетенции | знание основ и принципов программирования | - проектная деятельность, выполнение кейсов;- участие в конференциях, выставках, конкурсах, соревнованиях и т.п.;- выполнение практических заданий |
| знание и понимание основных алгоритмических конструкций |
| знание основ и овладение практическими базисными знаниями программирования в Python  |
| знание основ и овладение практическими базисными навыками разработки игр в PyGame |

**2.Комплекс организационно - педагогических условий**

**2.1. Календарный учебный график**

|  |  |
| --- | --- |
| **Позиции** | **Сроки реализации** |
| Количество учебных недель | 36 |
| Количество учебных дней | 36 |
| Даты начла и окончания учебного года | 01.09.2023-31.05.2024г. |
| Сроки промежуточной аттестации | 21.11.23; 30.01.2024г; 08.05.24.., сроки итоговой аттестации- 29.05.2024г. |

**2.2. Условия реализации программы**

**Материально-техническое обеспечение и оснащение:**

* компьютерные столы;
* компьютеры с установленной средой программирования Scratch;
* операционная система Windows;
* мультимедийный проектор;
* интерактивная доска;
* локальная сеть;
* доступ к сети Интернет; браузер.

**Информационное обеспечение**

* Материалы и презентации к урокам в LMS Яндекс.Лицея.
* Сайт «Python 3 для начинающих» – pythonworld.ru.
* Сайт «Питонтьютор» – pythontutor.ru.
* Лекции А. В. Умнова, прочитанные в Школе анализа данных Яндекса
* https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2 wVBjh5OpdwBl.
* ЭлектроннаябиблиотечнаясистемаeLIBRARY.RU
* (http://www.elibrary.ru)/

**Кадровое обеспечение**

 Реализация программы обеспечивается педагогическими работниками, отвечающими уровню образования по профилю программы и выполняющим трудовую функцию – Преподавание по дополнительным общеобразовательным программам – согласно приказу Министерства труда и социальной защиты РФ от 5 мая 2018 г. № 298н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых». Без требований к опыту работы.

**2.3. Формы аттестации**

**Формами аттестации являются:**

* Зачет
* Творческая работа
* Проект

**2.4. Оценочные материалы**

При наборе учащихся в объединение по интересам проводится начальная

аттестация, в ходе которой педагог проводит устный опрос и практическая

работа, по результатам которого узнает уровень подготовки учащихся к

занятиям.

 Формы промежуточной аттестации: теоретическая часть - письменный

опрос, практическая часть - практическая работа.

 Письменный опрос состоит из перечня вопросов по содержанию разделов программы, каждому из учащихся предлагается ответить письменно на несколько вопросов. Практическая работа предполагает задания по пройденному материалу.

Оценка теоретических знаний и практических умений и навыков учащихся

по теории и практике проходит по трем уровням: высокий, средний, низкий.

 Высокий уровень - учащиеся должны знать правила техники

безопасности при работе, грамотно излагать программный материал, знать основные блоки команд, уметь выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения и уметь самостоятельно создавать и выполнять программы для решения алгоритмических задач в программе Scratch.

 Средний уровень - учащиеся должны знать основные блоки команд,

уметь выполнять алгоритмы, описанные с использованием конструкций

ветвления и повторения, грамотно и по существу излагать программный

материал, не допуская существенных неточностей в ответе.

 Низкий уровень - учащиеся не знают значительной части материала,

допускают существенные ошибки, с большими затруднениями выполняют

практические задания.

При обработке результатов учитываются критерии для выставления

уровней:

Высокий уровень - выполнение 100% - 70% заданий;

Средний уровень - выполнение от 50% до 70% заданий;

Низкий уровень - выполнение менее 50% заданий.

Формы аттестации учащихся в течение учебного года

|  |  |  |  |
| --- | --- | --- | --- |
| Аттестация | Сроки | Теория | Практика |
| Начальная аттестация | ноябрь | устный опрос | Практическая работа |
| Промежуточная | январь | письменный опрос | Практическая работа |
| Промежуточная | май | письменный опрос | Практическая работа |
| Итоговая аттестация | май | конференция | Практическая работа (представление проектов) |

При проведении промежуточной аттестации Все результаты работы заносятся в таблицу представленную в Приложении 2.

**2.5. Методические материалы**

**Методы обучения:**

* Словесный
* Наглядный
* Частично-поисковый
* Исследовательский

**Формы организации образовательной деятельности:**

* Индивидуально-групповая
* Практическое занятие
* Защита проекта
* Презентация
* Мастер-класс

**Педагогические технологии:**

* Технология индивидуального обучения
* Технология группового обучения
* Технология коллективного взаимодействия
* Технология исследовательской деятельности
* Проектная технология

**Дидактические материалы:**

* Раздаточные материалы
* Инструкции
* Технологические карты

**2.6. Список литературы**

1. Долинский М.С. Решение сложных и олимпиадных задач по программированию **-** Учебное пособие **-** М.: **–** 2006.
2. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
3. Сэнд У., Сенд К. «Hello World! Занимательное программирование на языке Python» **-** М.: **–** 2016.
4. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.
5. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. –М.: МПГУ, 2015.
6. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
7. <http://informatics.mccme.ru/course/view.php?id=156>
8. <https://inf5.ru/podgotovka_k_olympiad/olym_zadachi_s_resheniyami.htm>
9. <http://anngeorg.ru/olimp/materials>
10. <http://kpolyakov.spb.ru/school/probook/python.htm>
11. <http://dist-olimpiada.krasnogorka.edusite.ru/p4aa1.html>

 Приложение 1.

Календарный учебный график

**«Основы языка программирования Python»- 1 год обучения**

|  |  |  |  |
| --- | --- | --- | --- |
| №п/п | Наименование раздела, темы | Количество часов | Формыаттестации/контроля |
| Всего  | Теория | Практика |
|  | **Раздел 1. Введение в Python** | **2** | **1,5** | **0,5** |  |
| 1 | Инструкции и структура программы. | 1 | 1 | 0 | Устный опрос |
| 2 | Установка Python. Вввод и вывод данных.  | 1 | 0,5 | 0,5 | Решение задач |
|  | **Раздел 2. Типы данных и операции** | **6** | **1,5** | **4,5** |  |
| 3 | Переменные. Операторы.  | 1 | 0,5 | 0,5 | Тест по теме |
| 4 | Переменные. Операторы. | 1 | 0 | 1 | Решение задач |
| 5 |  Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math) | 1 | 0,5 | 0,5 | Решение задач |
| 6 | Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math) | 1 | 0,5 | 0,5 | Решение задач |
| 7 | Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math) | 1 | 0 | 1 | Решение задач |
| 8 | Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math) | 1 | 0 | 1 | Решение задач |
|  | **Раздел 3. Инструкции и синтаксис** | **9** | **2,5** | **6,5** |  |
| 9 | Условный оператор if, Альтернативный условный оператор elif, else. | 1 | 1 | 0 | Тестирование |
| 10 | Условный оператор if, Альтернативный условный оператор elif, else. | 1 | 0 | 1 | Решение задач |
| 11 | Условный оператор if, Альтернативный условный оператор elif, else. | 1 | 0,5 | 0,5 | Решение задач |
| 12 | Условный оператор if, Альтернативный условный оператор elif, else. | 1 | 0 | 1 | Решение задач |
| 13 | Условный оператор if, Альтернативный условный оператор elif, else. | 1 | 0 | 1 | Решение задач |
| 14 | Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.  | 1 | 1 | 0 | Тестирование |
| 15 | Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.  | 1 | 0 | 1 | Решение задач с использованием циклов |
| 16 | Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.  | 1 | 0 | 1 | Решение задач с использованием циклов |
| 17 | Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.  | 1 | 0 | 1 | Решение задач с использованием циклов |
|  | **Раздел 4**. [Функции и модули в программировании](https://younglinux.info/python/function.php) | **9** | **2** | **7** |  |
| 18 | Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 1 | 1 | 0 | Тест по теме |
| 19 | Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 1 | 0 | 1 | Решение задач с использованием строк |
| 20 | Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 1 | 0 | 1 | Решение задач с использованием строк |
| 21 | Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 1 | 0 | 1 | Решение задач с использованием строк |
| 22 | Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 1 |  | 1 | Решение задач с использованием строк |
| 23 | Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. | 1 | 1 | 0 | Тест по теме |
| 24 | Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. | 1 | 0 | 1 |  Решение задач с использованием строк |
| 25 | Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. | 1 | 0 | 1 | Решение задач с использованием строк |
| 26 | Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. | 1 | 0 | 1 | Решение задач с использованием строк |
|  | **Раздел 5. Сложные типы данных** | **7** | **4** | **3** |  |
| 27 | Списки. Срезы списков. Решение задач со списками | 1 | 1 | 0 | Тест по теме |
| 28 | Списки. Срезы списков. Решение задач со списками | 1 | 0 | 1 | Решение задач |
| 29 | Матрицы | 1 | 0,5 | 0,5 | Решение задач |
| 30 | Словари | 1 | 0,5 | 0,5 | Решение задач |
| 31 | Множества в языке Python | 1 | 0,5 | 0,5 | Решение задач |
| 32 | Множества в языке Python | 1 | 0 | 1 | Решение задач |
| 33 | Множества в языке Python | 1 | 0 | 1 | Решение задач |
|  | **Раздел 6. Написание и отладка программ. Самостоятельная работа** | **3** | **0** | **3** |  |
| 34 |  Написание и отладка программ. | 1 | 0 | 1 | Самостоятельная работа |
| 35 | Написание и отладка программ. | 1 | 0 | 1 | Самостоятельная работа |
| 36 | Написание и отладка программ. | 1 | 0 | 1 | Самостоятельная работа |
|  | **Итого** | **36** | **10** | **26** |  |

 Приложение 2.

**Карта оценивания учащегося**

**Диагностическая карта**

|  |  |  |  |
| --- | --- | --- | --- |
| **№/п** | **Имя**  | **Входная диагностика** | **Итоговая диагностика** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **Итого**  | **%** | **Н****С****В** |  |

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В). Низкий уровень Учащиеся должны знать/ понимать: − технику безопасного поведения во время занятий; − правила поведения в общественных местах, − понятие программы Python; общую структурупрограммы; − основные типы данных; − оператор присваивания; 37 − назначение условного оператора; − способ записи условного оператора; − логический тип данных; − логические операторы or, and, not; − основные циклы с условием; − основные правила записи циклов условием; − формат записи цикла с параметром; − понятие функции; − основные принципы структурного программирования; − понятие локальных переменных подпрограмм; − способ передачи параметров. − назначение строкового типа данных; − операторы для работы со строками; − операции со строками; − способ описания списка; − способ описания кортежа; − способ описания словаря; − основные операции, выполняемые со списками, кортежами и словарями; − понятие множества; − способы описания множества; − операторы работы с множествами. − что такое стиль программирования; − правила именования объектов; − основные рекомендации при написании программ. Учащиеся должныуметь: − понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога; − работать в паре, малой группе; − выполнить установку программы под руководством педагога; − выполнить простейшую программу в интерактивной среде; − написать комментарии в программе под руководством педагога; − решать задачи на элементарные действия с числами; − использовать условный оператор; − определять вид цикла, наиболее удобный для решения поставленной задачи; − использовать цикл с условием под руководством педагога; − определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога; − создавать и использовать основные функции; − описывать и соединять строки; − находить подстроку в строке с помощью педагога; − находить количество слов в строке; − вводить и выводить элементы списка под руководством педагога; − приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога; − описывать множества под руководством педагога; − определять принадлежность элемента множеству по наводящим вопросам; − определять вид ошибок и находить ошибки в программе под руководством педагога; − составлять элементарные алгоритмы для решения задач; − реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога. Средний уровень Учащиеся должны знать/ понимать: 38 − основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности; − основные приемы взаимодействия в группе сверстников; − понятие программы; − структурупрограммынаPython; − режимы работы сPython. − типы данных; − целые,вещественные типы данных иоперациинадними; − оператор присваивания; − назначение условного оператора; − способ записи условного оператора; − логический тип данных; − логические операторы or, and, not; − циклы с условием и их виды; − назначение и особенности использования цикла с параметром; − формат записи цикла с параметром; − примеры использования циклов различных типов. − понятие функции; − основные способы описания функции; − принципы структурного программирования; − понятие локальных переменных подпрограмм; − понятие формальных и фактических параметров подпрограмм; − способ передачи параметров. − назначение строкового типа данных; − операторы для работы со строками; − процедуры и функции для работы со строками; − операции со строками; − сложные типы данных; − способ описания списка; − способ доступа к элементам списка; − способ описания кортежа; − способ описания словаря; − операции, выполняемые со списками, кортежами и словарями; − понятие множества; − способы описания множества; − операторы работы с множествами. − что такое стиль программирования; − правила именования объектов; − основные рекомендации при написании программ; − основные шаги работы над проектом, его презентации. Учащиеся должныуметь: − уважительно относится к преподавателям и сверстникам; − применять некоторые приемы логического (абстрактное) мышления; − концентрировать внимание на одном или двух объектах; − понимать причины успеха/неуспеха с помощью анализа педагога; − выполнить установку программы; − выполнить простейшую программу в интерактивной среде; − написать комментарии в программе; − решать задачи на элементарные действия с числами; − использовать условный оператор; − создавать сложные условия с помощью логических операторов; − определять вид цикла, наиболее удобный для решения поставленной задачи; 39 − использовать цикл с условием; − определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; − создавать и использовать функции; − использовать механизм параметров для передачи значений; − описывать строки; − соединять строки; − находить длину строки; − вырезать часть строки; − находить подстроку в строке; − находить количество слов в строке; − описывать списки; − вводить элементы списка; − выводить элементы списка; − выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; − использовать вложенные списки; − приводить примеры использования вложенных списков (матриц); − описывать множества; − определять принадлежность элемента множеству; − вводить элементы множества; − выводить элементы множества. − определять вид ошибок и находить ошибки в программе. − составлять алгоритмы для решения задач; − реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; − отлаживать и тестировать программы, написанные на языке Python; − понимать и сохранять цели и задачи учебной деятельности в ходе занятия; − планировать свою деятельность с помощью взрослого; − сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи; − понимать причины успеха/неуспеха учебной деятельности под руководством взрослого; − делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога; − выражать творческие идеи, разработать творческий проект на основе образца; − конструктивно взаимодействовать в составе группы в ходе работы над проектом. Высокий уровень Учащиеся должны знать/ понимать: − основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий; − приемы конструктивного взаимодействия в группе сверстников; − понятие программы; − структурупрограммынаPython; − режимы работы сPython. − общуюструктурупрограммы; − типы данных; − целые,вещественные типы данных иоперациинадними; − оператор присваивания; − назначение условного оператора; − способ записи условного оператора; − логический тип данных; 40 − логические операторы or, and, not; − циклы с условием и их виды; − правила записи циклов условием; − назначение и особенности использования цикла с параметром; − формат записи цикла с параметром; − примеры использования циклов различных типов. − понятие функции; − способы описания функции; − принципы структурного программирования; − понятие локальных переменных подпрограмм; − понятие формальных и фактических параметров подпрограмм; − способ передачи параметров. − назначение строкового типа данных; − операторы для работы со строками; − процедуры и функции для работы со строками; − операции со строками; − сложные типы данных; − способ описания списка; − способ доступа к элементам списка; − способ описания кортежа; − способ описания словаря; − операции, выполняемые со списками, кортежами и словарями; − понятие множества; − способы описания множества; − операторы работы с множествами. − что такое стиль программирования; − правила именования объектов; − основные рекомендации при написании программ; − правила и этапы работы над проектом; − приемы успешной презентации проекта. Учащиеся должныуметь: − выполнить установку программы; − выполнить простейшую программу в интерактивной среде; − написать комментарии в программе; − решать задачи на элементарные действия с числами; − использовать условный оператор; − создавать сложные условия с помощью логических операторов; − определять вид цикла, наиболее удобный для решения поставленной задачи; − использовать цикл с условием; − определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; − создавать и использовать функции; − использовать механизм параметров для передачи значений; − описывать строки; − соединять строки; − находить длину строки; − вырезать часть строки; − находить подстроку в строке; − находить количество слов в строке; − описывать списки; − вводить элементы списка; − выводить элементы списка; 41 − выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; − использовать вложенные списки; − приводить примеры использования вложенных списков (матриц); − описывать множества; − определять принадлежность элемента множеству; − вводить элементы множества; − выводить элементы множества. − определять вид ошибок и находить ошибки в программе. − составлять алгоритмы для решения задач; − реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; − отлаживать и тестировать программы, написанные на языке Python; − понимать учебную задачу, анализировать достижение результата; − делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений; − понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием пдагога; − планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи; − создать проектировочную команду и организовать ее деятельность; − разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы; − самостоятельно подготовить презентацию и защитить проект