

Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

РЕКОМЕНДОВАНО

Заведующий по учебной части

 Н. В. Грицюк

протокол № 9
от «5» сентября 2023 г.

УТВЕРЖДАЮ

 Е. А. Дыптан

Приказ № 02-02-36
от «5» сентября 2023 г.



Дополнительная общеобразовательная общеразвивающая
программа технической направленности

«Основы программирования на Python»

Срок реализации: 1 год

Возраст детей: 9-13 лет

Составители программы: Дзюбенко Е. В.

Норильск, 2023

Содержание

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
Новизна программы ДООП	3
Актуальность программы ДООП	3
Педагогическая целесообразность:.....	3
Цель ДООП.....	4
Возраст обучающихся, участвующих в программе	4
Условия вхождения в программу:.....	4
Срок реализации программы:	5
Режим занятий, формы и методы обучения.....	5
Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов.	5
УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	7
СОДЕРЖАНИЕ ПРОГРАММЫ	9
МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО- МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	12
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	13

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Современное общество переживает активную стадию цифровой трансформации. Все больше сфер жизни людей становятся зависимыми от информационных технологий и электроники. Одной из составляющих информационной компетентности является владение языком программирования, вследствие чего встает вопрос о выборе языка программирования, который отвечает современным требованиям к написанию программ, служит основой для дальнейшего развития и совершенствования навыков программирования и удобен в освоении подростками. Язык программирования Python принято считать одним из самых простых в освоении. Обучаться языку возможно имея лишь базовые навыки использования компьютера, установка необходимого ПО не сложнее установки любой другой программы, а синтаксис языка не перегружен и интуитивно понятен. Данные факторы раскрывают Python максимально эффективным инструментом при обучении детей программированию. При этом Python является современным, востребованным и перспективным языком разработки, позволяющим не только заложить фундамент знаний будущих программистов и инженеров, но получить востребованные и актуальные навыки, необходимые в учебной, проектной и повседневной деятельности.

Новизна программы ДООП

Дополнительная общеобразовательная программа «Основы программирования на Python» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

Актуальность программы ДООП

обусловлена повышенным спросом на изучение языков программирования детьми, в частности языка Python. Python изучается в школьном курсе информатики, необходим для решения олимпиадных и конкурсных заданий, а также сдачи ЕГЭ, имеет прикладной характер и может использоваться для решения повседневных задач. Кроме того, Python является востребованным языком программирования, используемым профессиональными инженерами во многих сферах IT-индустрии, поэтому знание данного языка даже на базовом уровне повышает шансы будущих выпускников на трудоустройство. Отличительная особенность программы заключается в возможности получения обучающимися универсальных компетенций, необходимых при дальнейшем изучении не только информационных технологий, но и предметов гуманитарного и естественно-научного цикла. Также в результате изучения парадигмы объектно-ориентированного подхода к программированию происходит формирование базовых знаний и умений для работы с большинством популярных языков и необходимых при освоении других IT-направлений.

Педагогическая целесообразность:

В связи с простотой синтаксиса в сравнении с другими языками программами (ясность кода, быстрота реализации) при изучении Python есть возможность сформировать у обучающихся представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях). В то же время Python является востребованным языком, он отлично подходит для концепции объектно-ориентированного программирования и активно применяется в различных областях от разработки веб-приложения до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный в

использовании инструмент для решения учебных задач и для создания собственных проектов. Вместе с тем, чистота и ясность его конструкций позволит в дальнейшем с легкостью выучить любой другой язык программирования.

Цель ДООП

Обучить основным принципам и этапам программирования и разработки программного обеспечения на основе языка программирования Python для последующего решения поставленных технических задач, а также для реализации творческих проектов.

Задачи:

- ✓ Формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- ✓ Знакомство с принципами и методами функционального программирования, объективно – ориентированного программирования;
- ✓ Приобретение навыков работы в интегрированной среде разработки на языке Python;
- ✓ Изучение конструкций языка программирования Python;
- ✓ Знакомство с основными структурами данных и типовыми методами обработки этих структур;
- ✓ Познакомить с понятием проекта и алгоритмом его разработки;
- ✓ Формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- ✓ приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач.

Программа обучения представляет собой 5 этапов. Обучение начинается со знакомства с языком Python и его синтаксисом. Затем изучаются множества и способы работы с ними. На третьем этапе учащиеся знакомятся с функциями. Далее изучаются некоторые библиотеки. И на последнем этапе слушатели курса готовят проект по итогам учебного года.

Особенностью данной программы является привлечение детей среднего школьного возраста к изучению программирования при помощи языка Python, т. к. он обладает следующими достоинствами:

- ✓ Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разнообразных программ, от текстовых процессоров до веб-браузеров;
- ✓ Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python несложно;
- ✓ В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;
- ✓ Python используется как язык программирования крупными корпорациями, такими Google.

Возраст обучающихся, участвующих в программе

Программа «Python 3 для начинающих» ориентирована на обучающихся от 9 до 13 лет. Занятия проводятся в группе из 12 человек.

Условия вхождения в программу:

Набор на Программу осуществляется в соответствии с Положением о наборе в Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске «Центр цифрового образования детей IT-Куб г. Норильск».

Поступающий на программу должен владеть базовыми умениями работы на компьютере (создание и удаление файлов; умение работать в простейшем текстовом и графическом редакторе; запуск, просмотр презентаций и видеороликов).

Срок реализации программы:

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет **144 часа**.

Режим занятий, формы и методы обучения

Учебные занятия проходят по очной форме обучения. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час 40 минут) с обязательным перерывом.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются следующие формы работы:

- ✓ демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- ✓ фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- ✓ самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия;
- ✓ повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов;
- ✓ закрепление знаний, умений и навыков через постановку задачи и самостоятельную работу обучающегося под руководством педагога;
- ✓ применение полученных знаний и навыков через прикладную работу обучающегося, использующего на практике приобретенные компетенции.

Ожидаемые образовательные результаты и эффекты, способы предьявления и отслеживания результатов.

Учащиеся будут уметь:

- ✓ Объяснять и использовать на практике как простые, так и сложные структуры данных и конструкций для работы с ними;
- ✓ Искать и обрабатывать ошибки в коде;
- ✓ Разбирать решение задач на подзадачи;
- ✓ Писать грамотный и красивый код;
- ✓ Находить, оценивать, использовать информацию из различных источников, необходимую для решения профессиональных задач, в том числе на основе системного подхода;
- ✓ Грамотно работать в команде, в зависимости от целей и ситуации.

Учащиеся будут знать:

- ✓ Основные классические алгоритмы и способы их реализации;
- ✓ Синтаксис языка программирования Python;
- ✓ Основные библиотеки (random, tkinter и др.);
- ✓ Основные элементы программирования: ввод/вывод информации, данные, операции с данными, условное выполнение, циклы, подпрограммы.

Сформированные навыки:

- ✓ соблюдение требований техники безопасности;
- ✓ работы в изучаемых программных средах;
- ✓ навыки составления алгоритмов;
- ✓ применения на практике основных команд и операторов изучаемых языков;
- ✓ разработки, тестирования и отладки несложных программ;
- ✓ навыки работы в сети Интернет для поиска информации.

Личностные результаты:

- ✓ учащиеся приобретут навыки самостоятельной организации своей деятельности; формирования основ саморазвития и самовоспитания;
- ✓ у обучающихся сформируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты:

- ✓ Умение планировать, умение анализировать, алгоритмизировать.
- ✓ Коммуникативные навыки:
 - Умение договариваться с другими людьми;
 - Работать в команде;
 - Аргументировать свою позицию;
 - Развить эмоциональный интеллект – способность понимать чужие чувства и контролировать свои.
- ✓ Навыки самоорганизации и тайм-менеджмент;
- ✓ Нестандартное мышление, креативные навыки;
- ✓ Умение работать с информацией
 - Анализ информации;
 - Компьютерная грамотность;
- ✓ Стрессоустойчивость.

Опыт:

Проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; информационной деятельности в различных сферах; эффективного применения информационных образовательных ресурсов в учебной деятельности, в том числе самообразовании; эффективной организации индивидуального информационного пространства.

Механизм оценки результативности

Контроль качества образования осуществляется в форме тестов, выполнения практических и проектных работ.

В конце каждой главы обучающиеся выполняют задания по написанию программ, на различные темы, связанные с пройденной главой. При полном прохождении курса оценка знаний будет проводиться по итогам выполнения обучающимися индивидуальных или групповых проектов, по темам, которые выбирают сами слушатели исходя из своих предпочтений.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Введение в программирование	60	21	39	
1.1	Введение. Устройство языка Python.	2	1	1	Тест
1.2.	Данные. Обработка данных	2	1	1	Практическая работа
1.3.	Построение командной работы.	4	2	2	Практическая работа
1.4.	Условный оператор	12	4	8	Практическая работа
1.5.	Циклы	8	3	5	Практическая работа
1.6.	Команды циклов остановок и прерывания циклов.	8	3	5	Практическая работа
1.7.	Циклы с подсчетом «range».	12	4	8	Практическая работа
1.8.	Вложенные циклы.	8	3	5	Практическая работа
1.9.	Групповая работа. Решение задач.	4	1	3	Практическая работа
2	Базовые конструкции Python	34	10	24	
2.1	Знакомство с коллекциями.	4	1	3	
2.2	Знакомство со строками.	4	2	2	Практическая работа
2.3.	Срезы в строках.	6	2	4	Практическая работа
2.4	Методы строк.	8	2	6	Практическая работа
2.5.	Знакомство со списками.	6	2	4	Практическая работа
2.6.	Групповая практическая работа.	6	1	5	Практическая работа
3	Знакомство с функциями	16	6	10	
3.1.	Функции.	4	1	3	Практическая работа
3.2.	Области видимости.	2	1	1	Практическая работа
3.3.	Передача параметров в функции.	2	1	1	Практическая работа
3.4.	Расширенные возможности функций.	8	3	5	Практическая работа
4	Работа с Модулями	34	9	25	
4.1.	Модули.	2	1	1	
4.2.	Знакомство с модулями	8	3	5	Практическая

	Python. Часть №1 (random)				работа
4.3.	Знакомство с модулями Python. Часть № 2 графика. Tkinter. PyQt5	6	1	5	Практическая работа
4.4.	Групповая практическая работа. Создаем игру «Сапёр» на PyQt5	4	1	3	Практическая работа
4.8.	Групповая практическая работа.	6	1	5	Практическая работа
4.9.	Разработка проекта.	8	2	6	Защита проекта
	Итого	144	47	97	

СОДЕРЖАНИЕ ПРОГРАММЫ

ТЕМА 1. Введение в программирование

1.1. Введение.

Теоретическая работа: Вводная лекция, знакомство с учениками. Проведение инструктажа по технике безопасности. Знакомство с историей языков программирования.

Практическая работа: Знакомство со средой разработки (IDLE) Pycharm. Установка и первый запуск.

1.2. Устройство языка Python.

Теоретическая работа: Иерархическое устройство языка Python. Знакомство с соглашением, о том, как писать код для языка Python pep8.

Практическая работа: Использование команд input и print.

1.3. Данные

Теоретическая работа: Знакомство с основными типами данных. Правила объявления переменных. Знакомство с Арифметическими действиями с переменными.

Практическая работа: Использование умножения, деления, сложения, вычитания с переменными одного типа, переменными разных типов.

1.4. Обработка данных

Теоретическая работа: Знакомство с различными способами ввода данных.

Практическая работа: Решение простейших задач на работу с арифметическими действиями.

1.5. Построение командной работы.

Теоретическая работа: знакомство с групповой работой. Разделение задач на подзадачи. Умение ставить цели и задачи для группы.

1.6. Условный оператор

Теоретическая работа: Знакомство с логическими функциями и/или/не, использование логических элементов в ветвлении. Правила оформления блоков ветвления с помощью отступов.

Практическая работа: Создание ветвлений, использование множественных ветвлений при решении задач.

1.7. Циклы

Теоретическая работа: Знакомство с понятием цикла, общий вид цикла.

Практическая работа: Практическое использование циклов While и for. Практическое использование отладчика в Pycharm

1.8. Команды циклов остановок и прерывания циклов.

Теоретическая работа: Знакомство с командами остановки и прерывания цикла break, continue. Преимущества и недостатки использования этих команд.

Практическая работа: Опробация полученных знаний на практике при решении задач.

1.9. Циклы с подсчетом «range».

Теоретическая работа: Знакомство с функцией range в циклах for. Генерация ряда чисел в заданном диапазоне.

Практическая работа: Использование аргументов при вызове функции range ().

Создание возрастающих и убывающих последовательностей с различным шагом.

1.10. Вложенные циклы.

Теоретическая работа: Знакомство с вложенными циклами.

1.11. Групповая работа. Решение задач.

Практическая работа: Разделение класса на группы. Разделение обязанностей в группе. Решение задач по пройденным темам.

Тема 2. Базовые конструкции в Python

2.1 Знакомство с коллекциями.

Теоретическая работа: Коллекция – «контейнер», содержащий различные элементы.

2.2 Знакомство со строками.

Теоретическая работа: Знакомство со строками. Функции и методы строк. Индексация. Операции над строками. Конкатенация. Повторение

Практическая работа: Выполнение базовых операций над строками: конкатенация, дублирование, определение длины строки (функция len), доступ к символам по индексу.

2.3. Срезы в строках.

Теоретическая работа: Знакомство со срезами, аргументы среза, отрицательные параметры.

Практическая работа: решение задач на извлечение среза из строки.

2.4 Методы строк.

Теоретическая работа: Знакомство с методами специфичными для строк.

Практическая работа: Преобразование строк с помощью метода split. Поиск символов по строке с помощью метода find, замена символов в строке с помощью символа replace.

2.5. Знакомство со списками.

Теоретическая работа: что такое список, его отличие от строки, различные операции над списками, методы специфичные для списка.

Практическая работа: Решение задач на арифметические действия со списками, поиск минимального или максимального числа в списке, добавить элемент или удалить элемент из списка.

2.6. Групповая практическая работа.

Практическая работа: Применение навыков тимбилдинга для распределения по группам. Получение проблемной ситуации. Групповая работа по решению проблемы.

Тема 3. Знакомство с функциями

3.1. Функции.

Теоретическая работа: Знакомство с функциями. Как возвращать значение из функции. Возвращения значений из функции. Инструкции return, def.

Практическая работа: с помощью инструкции def объявляем функцию, с помощью инструкции return возвращаем значение функции.

3.2. Области видимости.

Теоретическая работа: Области локальной и глобальной видимости, области нелокальной видимости. Операторы global, nonlocal,

Практическая работа: Использование инструкций `global` и `nonlocal`, при решении задач с функциями.

3.3. Передача параметров в функции.

Теоретическая работа: Аргументы функций. Функции с позиционированием, с произвольным числом элементов, именованными аргументами.

3.4. Расширенные возможности функций.

Теоретическая работа: Знакомство с дополнительными возможностями функции, работа рекурсивных функций, использование `lambda` функции.

3.5. Групповая практическая работа.

Практическая работа: Решение задач по теме функции. Командная работа в группах

Тема 4. Работа с модулями.

4.1. Модули.

Теоретическая работа: Способы установки модулей. Импорт функций из модулей. Импорт функций всех функций из модуля, перезагрузка модуля.

Практическая работа: Знакомство с `pip3`, работа с командной строкой.

4.2. Знакомство с модулями Python. Часть №1 (random)

Теоретическая работа: знакомство со встроенным модулем `Random`, основные функции генератора.

Практическая работа: Разбор функций генератора, написание программ с использованием генератора чисел.

4.3. Знакомство с модулями Python. Часть № 2 графика. Tkinter. PyQt5

Теоретическая работа: Знакомство с графическими модулями. Создание пользовательского интерфейса для программ. Расположение элементов пользовательского интерфейса в окне.

4.4. Групповая практическая работа. Создаем игру «Сапёр» на PyQt5

Практическая работа: Создание игры «Сапер».

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

- ✓ Стол преподавателя
- ✓ Стул преподавателя
- ✓ Стол обучающегося
- ✓ Стул обучающегося
- ✓ Рабочая станция преподавателя
- ✓ Ноутбук обучающегося
- ✓ Интерактивная доска
- ✓ Проектор
- ✓ МФУ

Точки подключения к электрической сети

- ✓ Программное обеспечение:
- ✓ Операционная система Windows 10
- ✓ Пакет программ MS OFFICE
- ✓ PyCharm
- ✓ Wing
- ✓ Gilly

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

- ✓ К. Ю. Поляков, Е. А. Еремин. Информатика. Углубленный уровень. Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2014
- ✓ М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2011
- ✓ Задачи по программированию. Под ред. С. М. Окулова. М.: БИНОМ, Лаборатория знаний, 2006
- ✓ С. М. Окулов. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2012
- ✓ Бриггс, Джейсон Б87 Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М.: Манн, Иванов и Фербер, 2017

Дополнительная литература:

- ✓ Информатика и ИКТ. Задачник – практикум в двух частях. Под ред. И.Г. Семакина Е.К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014

Электронные ресурсы:

- ✓ Материалы и презентации к урокам в LMS Яндекс. Лицея.
- ✓ Сайт pythonworld.ru – «Python 3 для начинающих»
- ✓ Сайт pythontutor.ru – «Питонтьютор»