

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 № 22-22-35
от «5» сентября 2023 г.



Дополнительная общеобразовательная общеразвивающая
программа технической направленности

«Основы программирования на языке Python
(Лицей Академии Яндекс)»

Срок реализации: 1 год

Возраст детей: 13-17 лет

Составители программы: Герасименко Е. О.

Норильск, 2023

СОДЕРЖАНИЕ

1.	3	
1.1.	НОВИЗНА ДООП	4
1.2.	АКТУАЛЬНОСТЬ ДООП	5
1.3.	ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ	5
1.4.	ЦЕЛЬ ДООП	5
1.5.	ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП	6
1.6.	УСЛОВИЯ ВХОЖДЕНИЯ В ДООП	6
1.7.	СРОК РЕАЛИЗАЦИИ ДООП	6
1.8.	РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ	6
1.9.	ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ	7
2.	УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	9
3.	СОДЕРЖАНИЕ ДООП	12
4.	МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	13
5.	РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	13

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека все больше и больше внедряются компьютеры и, как следствие, возрастает потребность в ИТ-специалистах.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне т. к. выделяется малое количество времени для изучения программирования. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ программирования и неумение применять полученные знания на практике.

Изучение программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе для изучения выбран язык программирования Python. Этот выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это снижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических частях программирования, а не на заучивании синтаксиса языка. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

1.1. НОВИЗНА ДООП

Программа «Основы программирования на языке Python» - технической направленности повышенного уровня сложности. Данная программа была подготовлена опираясь на уникальный опыт преподавания программирования в Школе анализа данных Яндекса (АНО ДПО «ШАД») и на факультете компьютерных наук НИУ ВШЭ. В ней большое внимание уделяется практической работе на компьютере, самостоятельному написанию кода.

Программа ориентирована на создание необходимых условий для личностного развития обучающихся, позитивной социализации и профессионального самоопределения.

Программа имеет техническую направленность, в связи с этим рассматриваются следующие разделы для изучения:

1. Технологический. Данная программа рассматривается как средство, позволяющее развивать наиболее передовые на сегодняшний день технологии – информационные, которые включают в себя, как теоретические знания, так и практические навыки.

2. Общеразвивающий. Обучение по данной программе создаёт благоприятные условия для интеллектуального развития ребенка, профессионального самоопределения, развития познавательной деятельности и творческой самореализации обучающихся.

3. Общеобразовательный. Данная программа позволяет развить основные познавательные процессы, умение анализировать, выявлять взаимосвязи и зависимости, делать логические выводы, и опирается на такие дисциплины, как теория управления, программирование, теория информации.

Научившись программировать на языке, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность конструкций Python позволит учащимся потом с легкостью выучить любой другой язык программирования.

Знания, полученные при прохождении данной программы, могут быть использованы обучающимися при сдаче ЕГЭ, для участия в олимпиадах по информатике, при написании научной работы по физике, химии, биологии и другим наукам, а также являются фундаментом для дальнейшего совершенствования мастерства программирования.

1.2. АКТУАЛЬНОСТЬ ДООП

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, не только на пользовательском уровне, но и на уровне начинающего программиста. В школьном курсе информатики программирование нередко представлено лишь на элементарном уровне. Следствием этого является формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Заключается в том, что дети приобретут практические навыки, которые станут основой для дальнейшего изучения программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Осваивая данную программу, обучающиеся будут овладевать навыками, которые будут востребованы в ближайшие десятилетия в специальностях, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, получаемые в процессе обучения по программе.

Важным аспектом программы является самостоятельная работа над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых:

- для решения задачи необходимо найти какую-то информацию в сети Интернет;
- может потребоваться устранение ошибки, которую не так просто быстро обнаружить;
- условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю).

Курс «Основы программирования на языке Python» рассчитан на 106 учебных часов (по 2 занятия в неделю, каждое продолжительностью 2 академических часа) и 38 часов самостоятельной работы и предназначен для учеников 8-го и 9-го классов школ разного уровня подготовки и с разной степенью мотивации.

1.4. ЦЕЛЬ ДООП

Цель программы: развитие алгоритмического и логического мышления посредством языка программирования «Python».

Обучающие задачи:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-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.

Воспитательные задачи:

- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- воспитание упорства в достижении результата;
- расширение кругозора обучающихся в области программирования.

Развивающие задачи:

- развитие у обучающихся интереса к программированию;
- приобретение навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач.

По окончании курса ученик приобретает следующие компетенции:

- знание основ современных языков программирования;
- умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умение искать и обрабатывать ошибки в коде;
- умение разбивать решение задачи на подзадачи;
- способность писать грамотный, красивый код;
- способность анализировать как свой, так и чужой код;
- способность работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- способность грамотно строить коммуникацию, исходя из целей и ситуации.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Python» ориентирована на обучающихся от 13 до 18 лет. Занятия проводятся в группе из 12 человек.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

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).

1.7. СРОК РЕАЛИЗАЦИИ ДООП

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 106 часов очных и 38 часов самостоятельных занятий. Дети, освоившие стартовый уровень, могут продолжить обучение на базовом уровне программы «Программирование на языке Python» (базовый уровень) без входного контроля.

1.8. 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 45 минут) с обязательным

перерывом.

Основной тип занятий - комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится педагогом самостоятельно, но с учетом рекомендованного календарно-тематического плана. Из-за регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в группе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом учащиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки. Пример заданий, выполняемых на занятиях, представлен в приложении 1.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведет к закреплению знаний, а также служит индикатором успешности образовательного процесса.

1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ

Личностные результаты:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершенным творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретенной благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному

уровню развития информационных технологий;

- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Предметные результаты:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- умение соблюдать нормы информационной этики и права.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название разделов, тем	Количество часов			Форма контроля
		Всего	Теория	Практика	
1	Модуль 1. Введение в программирование.	18	8	10	
1.1.	Знакомство с системой Яндекс. Простейшие программы с выводом на экран.	2	1	1	Практическая работа
1.2.	Условный оператор. Переменные и арифметика.	2	1	1	Практическая работа
1.3.	Простые встроенные функции.	2	1	1	Практическая работа
1.4.	Знакомство с циклом while	2	1	1	Практическая работа
1.5.	Отладчик	2	1	1	Практическая работа
1.6.	Знакомство с циклом for.	2	1	1	Практическая работа
1.7.	True и False, break и continue.	2	1	1	Практическая работа
1.8.	Вложенные циклы.	2	1	1	Практическая работа
1.9.	Обобщающий урок по теме: "Ветвления и циклы".	2	0	2	Практическая работа
2	Модуль 2. Базовые конструкции в Python.	30	9	21	
2.1.	Множества.	2	1	1	Практическая работа
2.2.	Строки и срезы. Индексация.	4	2	2	Практическая работа
2.3.	Знакомство со списками.	2	1	1	Практическая работа
2.4.	Обобщающий урок по теме: "Строки".	2	0	2	Практическая работа
2.5.	Кортежи. Преобразование коллекций.	2	1	1	Практическая работа
2.6.	Методы split и join. Списочные выражения.	2	1	1	Практическая работа
2.7.	Методы списков и строк.	2	1	1	Практическая работа
2.8.	Обобщающий урок по теме: "Простые коллекции".	2	0	2	Практическая работа

2.8.	Самостоятельная работа №1 Простые коллекции.	2	0	2	Самостоятельная работа
2.7.	Вложенные списки.	2	1	1	Практическая работа
2.8.	Знакомство со словарями.	2	1	1	Практическая работа
2.9.	Подготовка к контрольной работе.	2	0	2	Практическая работа
2.10.	Контрольная работа №1.	2	0	2	Контрольная работа
2.11.	Анализ результатов контрольной работы	2	0	2	Практическая работа
3	Модуль 3. Решение прикладных задач в Python	36	12	24	
3.1.	Функции. Возвращение значений из функций. Области видимости переменных. Функции: передача параметров. Функции с переменным числом аргументов. Функции как объект. Лямбда-функции.	12	6	6	Практическая работа
3.2.	Обработка коллекций. Поточковый ввод sys.stdin.	2	1	1	Практическая работа
3.3.	Решение задач на тему «Обработка коллекций. Поточковый ввод sys.stdin».	2	0	2	Практическая работа
3.4.	Обобщающий урок по теме: «Функции».	2	0	2	Практическая работа
3.5.	Самостоятельная работа №2 по теме «Функции».	2	0	2	Самостоятельная работа
3.6.	Библиотеки Python.	10	5	5	Практическая работа
3.7.	Решения задач на тему «Библиотеки Python».	2	0	2	Практическая работа
3.8.	Обобщающий урок по теме: "Библиотеки Python".	2	0	2	Практическая работа
3.9.	Самостоятельная работа №3 Библиотеки Python.	2	0	2	Самостоятельная работа
4	Модуль 4. Объектно-ориентированное программирование в Python	22	8	14	
4.1.	Введение в ООП. Полиморфизм.	4	2	2	Практическая работа

4.2.	Определение операторов.	2	1	1	Практическая работа
4.3.	ООП. Наследование.	4	2	2	Практическая работа
4.4.	Проектирование и разработка классов.	4	2	2	Практическая работа
4.5.	Решение задач на тему «Проектирование и разработка классов».	2	0	2	Практическая работа
4.6.	Обобщающий урок по теме: "ООП".	2	0	2	Практическая работа
4.7.	Контрольная работа №2.	2	0	2	Контрольная работа
4.8.	Анализ контрольной работы №2.	2	1	1	Практическая работа
	Итого:	106	37	69	

3. СОДЕРЖАНИЕ ДООП

Модуль 1. Введение в программирование

Знакомство с виртуальной средой взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой, автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов for, while и операторов ввода-вывода. Технология разработки программы.

На этом этапе обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В течение модуля ученики решают большое количество задач: от самых простых до сложных.

Модуль 2. Базовые конструкции языка Python

Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python.

Работа со списками, строками, множествами и кортежами в Python. Понятие итератора. Методы списков и строк. Знакомство со словарями.

Решение задач.

Модуль 3. Решение прикладных задач в Python

Понятие подпрограммы, процедуры, функции. Функции в Python.

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip.

Решение задач.

Модуль 4. Объектно-ориентированное программирование в Python

Введение в ООП.

Создание классов. Методы классов. Инициализация экземпляров класса. Инкапсуляция. Наследование. Полиморфизм. Специальные методы. Проверка типа объекта. Проектирование и разработка классов.

Решение задач.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

1. Стол преподавателя
2. Стул преподавателя
3. Стол обучающегося
4. Стул обучающегося
5. Рабочая станция преподавателя
6. Ноутбук обучающегося
7. Интерактивная доска
8. Проектор
9. МФУ
10. Точки подключения к электрической сети

Программное обеспечение:

1. Операционная система Windows 10
2. Интерпретатор языка программирование Python
3. Пакет библиотек и расширений Anaconda
4. PyCharm
5. Wing

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Бриггс, Джейсон, Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017.с.
2. Доусон М. Програмируем на Python. СПб: Питер, 2018.
3. Лутц М. Изучаем Python, том 1, 5-е изд: Пер. с англ. – СПб: ООО «Диалектика», 2019.

Дополнительная литература:

1. Поляков К. Ю., Еремин Е. А. Информатика. Углубленный уровень.
2. Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2014.
3. Окулова С. М. Задачи по программированию. М.: БИНОМ, Лаборатория знаний, 2006.
4. Окулов С. М. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2012.

Электронные ресурсы:

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

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Е.О. Герасименко педагог дополнительного образования, преподаватель по направлениям «Основы программирования на языке Python». Образование высшее. Негосударственное образовательное частное учреждение высшего образования «Московский финансово-промышленный университет «Синергия», г. Москва. Бакалавр по направлению «Информационные системы и технологии».

Задание 1.

1. Создать произвольный список.
2. Добавить новый элемент типа `str` в конец списка.
3. Добавить новый элемент типа `int` на место с индексом.
4. Добавить новый элемент типа `list` в конец списка.
5. Добавить новый элемент типа `tuple` на место с индексом.
6. Получить элемент по индексу.
7. Удалить элемент.
8. Найти число повторений элемента списка.

Задание 2. Получите первый и последний элемент списка.

Задание 3. Поменяйте значения переменных `a` и `b` местами.

Задание 4. Проверить, есть ли в последовательности дубликаты.

Задание 5.

1. Создать произвольный словарь.
2. Добавить новый элемент с ключом типа `str` и значением типа `int`.
3. Добавить новый элемент с ключом типа кортеж(`tuple`) и значением типа список(`list`).
4. Получить элемент по ключу.
5. Удалить элемент по ключу.
6. Получить список ключей словаря.

Задание 6.

1. Создать множество (`set`).
2. Создать неизменяемое множество (`frozenset`).
3. Выполнить операцию объединения созданных множеств.
4. Выполнить операцию пересечения созданных множеств.