

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 № 0236

от «5» сентября 2023 г.



Дополнительная общеобразовательная общеразвивающая
программа технической направленности

«Python 3. Второй год обучения»

Срок реализации: 1 год

Возраст детей: 10-14 лет

Составители программы: Дзюбенко Е. В.

Норильск, 2023

Содержание

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
Новизна программы ДООП	3
Актуальность программы ДООП	4
Педагогическая целесообразность:.....	4
Цель ДООП.....	4
Возраст обучающихся, участвующих в программе	5
Условия вхождения в программу:.....	5
Срок реализации программы:	6
Режим занятий, формы и методы обучения.....	6
Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов.	6
УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	8
СОДЕРЖАНИЕ ПРОГРАММЫ	10
МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО- МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	12
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	13

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных учащихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

Новизна программы ДООП

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.

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

В рамках предлагаемого курса «Программирования на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и

метапредметных результатов учащихся особенно важно, т.к. именно они активизирует процесс индивидуально-личностного становления учащихся.

Актуальность программы ДООП

Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выразить свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

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

Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи:

Образовательные:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Воспитательные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

Развивающие:

- развить познавательные процессы (внимание, восприятие, логическое мышление, память),
- развить креативность,
- развить способности к самореализации.

Программа представляет 5 этапов. Обучение начинается со знакомством языка Python и его синтаксисом. Затем изучаются множества и способы работы с ними. На третьем этапе знакомятся с функциями. Затем изучаются некоторые библиотеки. И на последнем этапе 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.

На первом этапе обучающиеся ознакомятся выводом и вводом данных, с основами синтаксиса языка Python, узнают о типах данных, переменных. Узнают о типах циклов и вложенных циклах. Научатся строить конструкции ветвлений if/else/elif. На этом этапе закладываются основы работы с Python. На втором этапе обучающиеся знакомятся с основными конструкциями языка, для обработки и хранения больших объемов данных. Узнают какие виды данных неизменяемые, как делать выборку по объектам. Научатся делать многомерные матрицы. На третьем этапе обучающиеся знакомятся и работают с функциями. Узнают, как передавать и изымать данные из функции, научатся сами писать функции. Четвертый этап поможет познакомиться с известными библиотеками.

Возраст обучающихся, участвующих в программе

Программа «Python 3. Второй год обучения» ориентирована на обучающихся от 10 до 14 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.

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе – обучение учащихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п. Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

Ожидаемые образовательные результаты и эффекты, способы предъявления 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 написания 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	Базовые конструкции Python	56	16	40	
1.1.	Знакомство с коллекциями.	4	1	3	Практическая работа
1.2.	Знакомство со строками.	4	2	2	Практическая работа
1.3.	Срезы в строках.	6	2	4	Практическая работа
1.4.	Методы строк.	8	2	6	Практическая работа
1.5.	Знакомство со списками.	6	2	4	Практическая работа
1.6.	Методы списков	10	3	7	Практическая работа
1.7.	Выводы элементов списков	4	1	3	Практическая работа
1.8.	Списочные выражения	4	1	3	Практическая работа
1.9.	Сортировка списков	4	1	3	Практическая работа
1.10.	Групповая практическая работа.	6	1	5	Практическая работа
2	Знакомство с функциями	28	10	18	
2.1	Функции.	4	1	3	Практическая работа
2.2	Области видимости.	2	1	1	Практическая работа
2.3.	Передача параметров в функции.	2	1	1	Практическая работа
2.4	Локальные и глобальные переменные	2	1	1	Практическая работа
2.5.	Функции с возвратом значений	4	1	3	Практическая работа
2.6.	Расширенные возможности функций.	8	3	5	Практическая работа
2.7.	Рекурсивные функции	6	2	4	Практическая работа
3	Работа с Модулями	60	15	45	Практическая работа
3.1.	Модули.	2	1	1	Практическая работа
3.2.	Знакомство с модулями Python. Часть №1 (random)	8	3	5	Практическая работа
3.3.	Знакомство с модулями Python. Часть № 2 графика. Tkinter. PyQT5	6	1	5	Практическая работа

3.4.	Групповая практическая работа. Создаем Графический редактор на PyQt5	6	1	5	Практическая работа
3.5.	Знакомство с модулями Python. Часть №3 анимация. Знакомство с модулем pygame.	4	1	3	Практическая работа
3.6.	Геометрические примитивы в pygame. Знакомство с анимацией.	4	1	3	Практическая работа
3.7.	Групповая практическая работа. Летающий круг.	4	1	3	Практическая работа
3.8.	Знакомство с модулями Python Часть №4 (морфология). Морфологический анализатор русского языка модуль rumorphy2.	4	1	3	Практическая работа
3.9.	Морфологический анализ текста с помощью rumorphy2.	4	1	3	Практическая работа
3.10.	Знакомство с модулями Python. Часть №5 работа с документами. Модуль python - docx	4	1	3	Практическая работа
3.11.	Групповая практическая работа.	6	1	5	Практическая работа
3.12.	Разработка проекта.	8	2	6	Защита проекта
	Итого	144	41	103	

СОДЕРЖАНИЕ ПРОГРАММЫ

ТЕМА 1. Базовые конструкции в Python

1.1 Знакомство с коллекциями.

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

1.2 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.

1.3. Срезы в строках.

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

Практическая работа: решение задач на извлечение среза из строки.

1.4 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.

1.5. 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.

1.6 Методы списков.

Теоретическая работа: Знакомство с методами специфичными для списков.

Практическая работа: Подсчет элементов списка, сортировка, добавление и удаление элементов списка.

1.7 Выводы элементов списков.

Теоретическая работа: Вывод списка на экран.

Практическая работа: Работа со списками, вывод списком на экран, срезы списков.

1.8 Списочные выражения.

Теоретическая работа: Знакомство со списочными выражениями.

Практическая работа: Преобразование списков с помощью различных методов и функций.

1.9 Сортировка списков.

Теоретическая работа: Правила сортировки списков.

Практическая работа: сортировка списка в алфавитном и обратном алфавитном порядке.

1.10. Групповая практическая работа.

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

Тема 2. Знакомство с функциями

2.1. 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.

2.2. 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.

2.3. Передача параметров в функции.

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

2.4. Локальные и глобальные переменные.

Теоретическая работа: Знакомство с понятиями. Определение, когда правильно использовать глобальные и локальные переменные, в каких случаях их необходимо описывать

Практическая работа: Написание функций с использованием локальных и глобальных переменных.

2.5. Функции с возвратом значений.

Теоретическая работа: Знакомство с ключевой особенностью функций — возможность возвращать значение.

Практическая работа: Решение задач по теме функции с возвратом значений.

2.6. Расширенные возможности функций.

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

2.7. Рекурсивные функции.

Теоретическая работа: Знакомство с рекурсией, способами ее записи.

Практическая работа: Решение задач по теме функции. Рекурсии.

Тема 4. Работа с модулями.

3.1. Модули.

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

Практическая работа: Знакомство с Pip3, работа с командной строкой.

3.2. 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.

3.3. 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.

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

Практическая работа: Создание игры «Сапер».

3.5. Знакомство с модулями Python. Часть №3 анимация. Знакомство с модулем pygame.

Теоретическая работа: Знакомство с модулем Pygame, функции модуля,

3.6. Геометрические примитивы в pygame. Знакомство с анимацией.

Практическая работа: Создание примитивов. Знакомство с анимацией, созданной с помощью модуля Pygame.

3.7. Групповая практическая работа. Летящий круг.

Практическая работа: Создание программы, в которой круг будет отскакивать от препятствий.

3.8. Знакомство с модулями Python Часть №4 (морфология).

Морфологический анализатор русского языка модуль rymorphy2.

Теоретическая работа: Морфологический анализатор rymorphy2. Функции. Разбор слов.

3.9. Морфологический анализ текста с помощью rymorphy2.

Практическая работа: Практическая задача на морфологический разбор текста, поиск слова в тексте, поиск ошибок в тексте.

4.10. Знакомство с модулями Python. Часть №5 работа с документами.

Модуль python – docx

Теоретическая работа: Знакомство с функциями модуля python-docx, создание шаблонов для заполнения текстового документа.

4.11. Групповая практическая работа.

Теоретическая работа: Создание программы, которая, используя различные модули записывает результат вычислений в виде шаблона в файл формата docx.

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»