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

«Python 3 для новичков. Продвинутый Python»

Срок реализации: 1 год
Возраст детей: 13-17 лет
Составители программы: Комаров Е. А.

Норильск, 2023

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека все больше и больше внедряются компьютеры. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причем зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, т. к. выделяется малое количество времени для его изучения. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является лишь 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 это 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

Новизна программы состоит в том, что реализуется возможность обучению навыкам работы в группе, создания коллективных проектов, чего практически невозможно достичь при изучении традиционных языков Бейсик и Паскаль. Возможность увидеть результаты своего труда в Интернет также стимулирует интерес детей получить представление об алгоритмах и исполнителях, основных алгоритмических конструкциях языков программирования. В рамках предлагаемой программы 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. Исключительно велика роль программирования для формирования мышления школьников, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся.

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

В связи с простой 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 получат 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.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения по отношению к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции — одной из ключевых компетенций современной жизни. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных школьников.

1.4 Цель ДООП

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

Обучающие задачи:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами создания интерфейса программ;
- создание, обработка и редактирования баз данных на различных платформах;
- 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;
- знакомство с понятием проекта и алгоритмом его разработки;
- знакомство с git и github.

Воспитательные задачи:

- развитие у учащихся инициативность и самостоятельность;
- мотивация к созданию собственных проектов;
- развитие стремление к получению качественного законченного результата в проектной деятельности;
- воспитание социально значимые качества личности человека: ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления.

Развивающие задачи:

- развитие логического, абстрактного и образного типов мышления;
- развитие творческих способностей;
- 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 представляет 4 этапа. На первом этапе обучающиеся повторяют и закрепляют знания полученные на первом году обучения. На втором этапе знакомятся с интерфейсом pyqt5, и создают собственный UI. Далее обучающиеся узнают дополнительные инструменты программиста, такие как: git, sql, docker т.д., изучают Django и создают собственный web-интерфейс.

1.5 Возраст обучающихся, участвующих в ДООП

Программа «Python 3. Графический интерфейс и базы данных» ориентирована на обучающихся от 13 до 17 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, знать синтаксис языка Python, уметь писать функции, понимать принципы объектно-ориентированного программирования. Обязательно предварительное освоение курса по программе «Python 3 для новичков. Основы Python».

1.7 Срок реализации ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 часа за год.

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 40 минут) с обязательным перерывом.

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

1.9 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;
- создавать web-приложения;
- работать с системой контроля версий;
- использовать репозиторий GitHub;
- работать с базами данных;
- 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:

- синтаксис языка программирования Python;
- Bash-команды для Git;
- принципы создания web – сервисов с помощью Django;

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.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	**Повторение**	**6**	**3**	**3**	
1.1	Повторение синтаксиса Python	2	1	1	
1.2	Повторение функционального программирования	2	1	1	
1.3	Повторение ООП	2	1	1	
2	**PyQt5**	**26**	**9**	**17**	
2.1	Знакомство с модулем pyQT5	2	1	1	
2.2	Знакомство с интерфейсом Qt Designer	2	1	1	
2.3	Создание дизайна окон в Qt Designer	2	1	1	
2.4	Способы преобразования дизайна в код	2	1	1	
2.5	Расширение функциональности Python GUI приложения	2	1	1	
2.6	Создание программы с простейшим интерфейсом.	6	2	4	
2.7	Групповая практическая работа	10	2	8	Защита групповой работы
3	**Git-система контроля версий**	**6**	**3**	**3**	
3.1	Введение в систему контроля версий Git.	2	1	1	
3.2	Внедрение системы контроля Git & GitHub	2	1	1	
3.3	История работы и ветки	2	1	1	
4	**Базы данных для Python для разработчиков**	**14**	**6**	**8**	
4.1	Введение в базы данных. Типы баз данных	2	1	1	

4.2	Работа с PostgreSQL. Создание БД	2	1	1	
4.3	Select-запросы, выборки из одной таблицы	2	1	1	
4.4	Продвинутая выборка данных	2	1	1	
4.5	Работа с PostgreSQL из Python	2	1	1	
4.6	Python и БД. ORM	4	1	3	
5	**Профессиональная работа с Python**	**16**	**7**	**9**	
5.1	Модули, пакеты, импорты в Python	2	1	1	
5.2	Регулярные выражения	4	1	3	
5.3	Веб-скрапинг	4	2	2	
5.4	Итераторы, генераторы	4	2	2	
5.5	Декораторы	2	1	1	
6	**Django: создание функциональных веб-приложений**	**22**	**10**	**12**	
6.1	Знакомство с Django. Подготовка и запуск проекта.	4	2	2	
6.2	Обработка запросов	2	1	1	
6.3	Динамическое формирование страниц на основе шаблонов	4	1	3	
6.4	Базы данных (2 части)	4	2	2	
6.5	Знакомство с API на примере Django REST framework	4	2	2	
6.6	Django REST framework: фильтрация, валидация, аутентификация	2	1	1	
6.7	Тестирование Django-приложений с использованием Pytest	2	1	1	
7	**Python в веб-разработке**	**20**	**7**	**13**	
7.1	Flask	4	1	3	
7.2	Docker	4	2	2	
7.3	CI/CD	2	1	1	
7.4	Asyncio	4	1	3	
7.5	Aiohttp	4	1	3	
7.6	Celery	2	1	1	
8	**Итоговый проект**	**34**	**4**	**30**	
8.1	Выполнение проекта	**32**	**2**	**30**	
8.2	Защита проекта	**2**	**2**	**0**	**Защита проекта.**

3. СОДЕРЖАНИЕ ПРОГРАММЫ

ТЕМА 1: Повторение

1.1 Повторение синтаксиса Python

Теоретическая работа: Знакомство с модулем pyQT5. Состав. Возможности устройство.
Практическая работа: Установка модуля pyQT5. Первый запуск. Создание простейшего окна.

1.2 Повторение функционального программирования

Теоретическая работа: Знакомство с приложением Qt Designer. Возможности приложения. Знакомство с меню. Интерфейс приложения.
Практическая работа: Создание простейших окон. Макет приложения.

1.3 Повторение ООП

Теоретическая работа: Эргономика управления. Интуитивный интерфейс.
Практическая работа: Создание окна приложения. Настройка виджетов. Командная разработка интуитивного интерфейса.

ТЕМА 2: PyQT5

2.1 Знакомство с модулем pyQT5

Теоретическая работа: Знакомство с модулем pyQT5. Состав. Возможности устройство.
Практическая работа: Установка модуля pyQT5. Первый запуск. Создание простейшего окна.

2.2 Знакомство с интерфейсом Qt Designer.

Теоретическая работа: Знакомство с приложением Qt Designer. Возможности приложения. Знакомство с меню. Интерфейс приложения.
Практическая работа: Создание простейших окон. Макет приложения.

2.3 Создание дизайна окон в Qt Designer.

Теоретическая работа: Эргономика управления. Интуитивный интерфейс.
Практическая работа: Создание окна приложения. Настройка виджетов. Командная разработка интуитивного интерфейса.

2.4 Способы преобразования дизайна в код.

Теоретическая работа: Использование .ui файла. Знакомство с командами для преобразования .ui в .py файл
Практическая работа: Преобразование файл интерфейса с помощью pyui5. Знакомство с файлом интерфейса.

2.5 Расширение функциональности Python GUI приложения.

Теоретическая работа: Классы, методы, функции для создания интерфейса.
Практическая работа: Создание функционала виджетов. Функциональное и объектно-

ориентированное программирование при создании интерфейса. События. Реакция на события.

2.6 Создание программы с простейшим интерфейсом.

Теоретическая работа: Распределение по группам. Выбор программы для создания интерфейса. Проектирование интерфейса. Пример текстовый редактор.
Практическая работа: Создание программы по выбранному плану. Тестирование программы. Выпуск программы в продакшн.

2.7 Повторение изученного материала.

Практическая работа: Проверка знаний по пройденному материалу. Пример заданий и критерии оценивания. Приложение 1

ТЕМА 3: Git-система контроля версий

3.1 Введение в систему контроля версий Git.

Теоретическая работа: Теория о системе контроля версий. Отличие Git от других VCS.
Практическая работа: Установка Git на Linux, macOS, Windows.

3.2 Внедрение системы контроля Git & GitHub.

Теоретическая работа: Работа в локальном репозитории, фиксация изменений, игнорирование файлов.
Практическая работа: Регистрация на Github, создание репозитория, создание файла readme.md.

3.3 История работы и ветки Git & GitHub.

Теоретическая работа: Ветки. Теги. Работа с историей. Откат изменений.
Практическая работа: Работа по слиянию веток репозитория, откат изменений и создание новых веток.

ТЕМА 4: Базы данных для python разработчиков

4.1 Введение в базы данных. Типы баз данных.

Теоретическая работа: БД и СУБД. Типы БД. Таблицы, атрибуты, кортежи. Первичный ключ и связи между таблицами. Нормальные формы
Практическая работа: Проектирование схемы для музыкального сайта.

4.2 Работа с PostgreSQL. Создание БД

Теоретическая работа: Что такое SQL. Типы запросов. Типы полей. Связи между отношениями.
Практическая работа: Создание схемы с различными типами связей между отношениями. Создание SQL-Запросов для создания БД.

4.3 Select-запросы, выборки из одной таблицы

Теоретическая работа: запросы: INSERT, UPDATE, DELETE. SELECT-запросы и

арифметические операторы.
Практическая работа: Заполнение базы данных с помощью INSERT запросов. Создание SELECT – запросов

4.4 Продвинутая выборка данных

Теоретическая работа: Агрегирующие функции и операторы. Объединение таблиц с помощью JOIN. Индексы.
Практическая работа: Написание продвинутых SELECT запросов.

4.5 Работа с PostgreSQL из Python

Теоретическая работа: Работа с библиотекой psycorpg2. Подключение к БД. Создание SELECT-запросов.
Практическая работа: Создание программу для управления клиентами на Python.

4.6 Python и БД. ORM

Теоретическая работа: Концепт ORM. Абстракция ORM.
Практическая работа: С помощью SQLAlchemy реализовать БД, по заданной схеме. Составить запросы выборки данных.

ТЕМА 5: Профессиональная работа с Python

5.1 Модули, пакеты, импорты в Python.

Теоретическая работа: Модули. Пакеты. Импорты. Декомпозиция. PIP
Практическая работа: Разработка структуры программы по шаблону. Знакомство с модулем datetime.

5.2 Регулярные выражения.

Теоретическая работа: Что такое регулярные выражения. Синтаксис регулярных выражений. Модуль re в Python. Примеры использования.
Практическая работа: Применение регулярных выражений для форматирования адресной книги.

5.3 Веб-скрапинг

Теоретическая работа: Что такое web-scrapping. Проблемы при скрапинге. Инструменты. Извлечение информации.
Практическая работа: Поиск статьи на сайте. Сохранение лога данных.

5.4 Итераторы, генераторы

Теоретическая работа: List и память. Как работает цикл for. Итераторы. Генератор и Yeld.
Практическая работа: Написание итератора и генератора для получения данных из списка.

5.5 Декораторы

Теоретическая работа: Паттерны проектирования. Как изменить функцию. Декораторы. Область видимости. Параметризированный декоратор.

Практическая работа: Написание декоратора-логгера.

ТЕМА 6: Django: создание функциональных веб-приложений

6.1 Знакомство с Django. Подготовка и запуск проекта.

Теоретическая работа: Что такое Django. Создание проекта. Функции. URL.
Практическая работа: Реализация view функций и настройка URL.

6.2 Обработка запросов.

Теоретическая работа: Лендинг. Пагинация. Работа с файловым менеджером.
Практическая работа: Тестирование дизайна лендингов. Релизация пагинации данных.

6.3 Динамическое формирование страниц на основе шаблонов.

Теоретическая работа: Создание шаблонов. Хед-контейнер. Фут-контейнер.
Практическая работа: Создание статистических таблиц по данным с сайта.

6.4 Базы данных.

Теоретическая работа: Работа с ORM в Django.
Практическая работа: Создание онлайн-библиотеки.

6.5 Знакомство с API на примере Django REST framework.

Теоретическая работа: Создание, чтение, обновление, удаление сущностей по API.
Практическая работа: Создание программы для получения данных с API обработка и сохранение в БД.

6.6 Django REST framework: фильтрация, валидация, аутентификация.

Теоретическая работа: Работа с модулем валидации Django REST framework.
Практическая работа: Регистрация и аутентификация пользователей на сайте с помощью Django REST framework.

6.7 Тестирование Django-приложений с использованием Pytest.

Теоретическая работа: Библиотека Pytest. Создание декораторов для запуска. Сравнения результата.
Практическая работа: По готовому приложению создать тесты.

ТЕМА 7: Python в веб-разработке

7.1 Flask.

Теоретическая работа: Знакомство с Flask, POST, GET, DELETE запросы.
Практическая работа: Создание системы авторизации сайта с помощью Flask.

7.2 Docker.

Теоретическая работа: Знакомство с Docker. Работа с контейнерами.
Практическая работа: Создание собственного контейнера Docker для Flask приложения.

7.3 CI/CD.

Теоретическая работа: Непрерывная интеграция. Непрерывное тестирование.
Практическая работа: Запуск собственного приложения на Flask на Heroku. Запустить процессы CI/CD.

7.4 Asyncio.

Теоретическая работа: Знакомство с модулем Asyncio. Корутины. Футуры. Блокирующие функции.
Практическая работа: Создание асинхронных функций для выгрузки данных по API.

7.5 Aiohttp.

Теоретическая работа: Создание проекта. Структура проекта. View. Route. Создание шаблона. Запуск приложения.
Практическая работа: Переписать Flask приложение на Aiohttp.

7.6 Celery.

Теоретическая работа: Долго-выполняющиеся фоновые задачи. Работа с Celery.
Практическая работа: Создание асинхронного метода API для рассылки сообщений.

ТЕМА 8: Итоговый проект

8.1 Выполнение проект.

Теоретическая работа: Постановка задачи. Распределение на команды.
Практическая работа: Работа по выполнению проекта, по поставленной задаче.

8.2 Подведение итогов

Теоретическая работа: Защита работ учеников.

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:

11. Операционная система Windows 10
12. Пакет программ MS OFFICE
13. PyCharm
14. Wing

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

- К. Ю. Поляков, Е. А. Еремин. Информатика. Углубленный уровень. Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2019
- М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2019
- Задачи по программированию. Под ред. С. М. Окулова. М.: БИНОМ, Лаборатория знаний, 2017
- С. М. Окулов. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2017

Дополнительная литература:

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

Электронные ресурсы:

- Сайт pythonworld.ru – «Python 3 для начинающих»
- Сайт pythontutor.ru – «Питонтьютор»

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Комаров Е.А. педагог дополнительного образования, преподаватель по направлениям «Python для новичков» и «VR/AR». Образование высшее. ФГБОУ ВО «Казанский государственный энергетический университет» диплом по направлению «Электроэнергетика и электротехника» квалификация «Бакалавр». АНО ДПО «ФИПКиП» квалификация «учитель информатики»