

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

УТВЕРЖДАЮ

Директор

Герасименко Е.О.

Приказ № 02-02-03/2

от «25» сентября 2020 г



Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Python 3 для начинающих»

Срок реализации:

1 год

Возраст детей:

14-17 лет

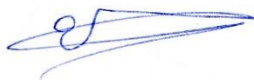
Составитель программы:

Комаров Е. А.

г. Норильск, 2020 г.

Программу составил(и):

педагог дополнительного образования



Егор Андреевич Комаров

Рецензент(ы):

Заведующий кафедрой «Информационные системы и технологии»,
ФГБОУ ВО «НГИИ»,
к.э.н., доцент



Михаил Вадимович Петухов

**ДООД одобрена на заседании методического совета
Филиал АНО «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей ИТ-Куб г. Норильск»**

Протокол от 30.09.2020 №1

Председатель методического совета
заведующий по учебной части



Елена Вячеславовна Белоусова

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	4
1.1. НОВИЗНА ДООП.....	4
1.2. АКТУАЛЬНОСТЬ ДООП.....	4
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.....	7
1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ.....	7
2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.....	9
3. СОДЕРЖАНИЕ ДООП	11
4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	15
5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	15

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 изучения программирования. 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 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

Предлагаемая ДООП нацелена на изучение языка программирования Python. Новизна данной программы заключается в практической направленности. Обучающиеся не просто слушают материал, но реализуют проекты начиная со второго занятия.

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

ДООП 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 создает благоприятные условия для интеллектуального развития ребенка, профессионального самоопределения, развития познавательной деятельности и творческой самореализации обучающихся.

3. Общеобразовательный. Данная ДООП 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.

Знания, полученные при прохождении данной программы, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по информатике, 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

В связи с простотой синтаксиса в сравнении с другими языками программирования (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.

1.4. 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.

Обучающие задачи:

1. формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
2. знакомство с принципами и методами функционального программирования, объективно – ориентированного программирования;
3. приобретение навыков работы в интегрированной среде разработки на языке Python;
4. изучение конструкций языка программирования Python;
5. знакомство с основными структурами данных и типовыми методами обработки этих структур;
6. познакомить с понятием проекта и алгоритмом его разработки;
7. ознакомить учащихся с понятиями «экспертные системы» и «нейронные сети», развить основные навыки работы с этими технологиями;
8. сформировать у учащихся представление о современных тенденциях развития технологий искусственного интеллекта в мире.

Воспитательные задачи:

1. развить у учащихся инициативность и самостоятельность;
2. мотивировать к созданию собственных проектов;
3. развить стремление к получению качественного законченного результата в проектной деятельности;
4. воспитать социально значимые качества личности человека: ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления.

Развивающие задачи:

1. развить логический, абстрактный и образный типы мышления;
2. развить творческие способности;
3. сформировать самостоятельность и творческий подход к решению задач с использованием средств вычислительной техники;

4. приобрести навыки поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач.

ДООП представляет 5 этапов. Обучение начинается со знакомства с языком 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 знакомятся с основами ООП и готовят проект по итогам учебного года.

Особенностью данной ДООП 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 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. В рабочей программе заложено знакомство с библиотеками random, tkinter, pygame, PyQT5 и др. последнем пятом этапе Обучающиеся ознакомятся с азами объектно-ориентированного программирования и будут участвовать в разработке собственного проекта.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Python 3 для начинающих» ориентирована на обучающихся от 14 до 17 лет. Занятия проводятся в группе из 12 человек.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на ДООП осуществляется в соответствии с Порядком приема и отчисления обучающихся Филиала 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 учебный год. Годовая нагрузка на обучающегося составляет 144 часа.

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, что определяется Санитарно-эпидемиологическими правилами и нормативами СанПиН 2.4.4.3172-14.

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

При проведении занятий используются три 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:

- проведение опросов по пройденным темам и проектные работы в конце каждого модуля, педагогом проводится 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.

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

В результате освоения образовательной программы обучающийся должен:

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.

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.

Обладать навыками:

- соблюдения требований техники безопасности;
- работы в изучаемых программных средах;

- составления алгоритмов;
- применения на практике основных команд и операторов изучаемых языков;
- разработки, тестирования и отладки несложных программ;
- 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 тестов¹, 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.

¹Примеры тестовых заданий представлены в Приложении 1.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Раздел 1. Введение в программирование	26	11	15	
1.1.	Тема 1.1. Введение.	2	1	1	
1.2.	Тема 1.2. Устройство языка Python.	2	1	1	
1.3.	Тема 1.3. Данные	2	1	1	
1.4.	Тема 1.4. Обработка данных	2	1	1	
1.5.	Тема 1.5. Построение командной работы.	4	1	3	
1.6.	Тема 1.6. Условный оператор	2	1	1	
1.7.	Тема 1.7. Циклы	2	1	1	
1.8.	Тема 1.8. Команды циклов остановок и прерывания циклов.	2	1	1	
1.9.	Тема 1.9. Циклы с подсчетом «range».	2	1	1	
1.10.	Тема 1.10. Вложенные циклы.	2	1	1	
1.11.	Тема 1.11. Групповая работа. Решение задач.	4	1	3	Практическая работа
2	Раздел 2. Базовые конструкции Python	30	12	18	
2.1	Тема 2.1. Знакомство с коллекциями.	2	1	1	
2.2	Тема 2.2. Знакомство со строками.	2	1	1	
2.3.	Тема 2.3. Срезы в строках.	2	1	1	
2.4	Тема 2.4. Методы строк.	4	1	3	
2.5.	Тема 2.5. Знакомство со списками.	4	2	2	
2.6.	Тема 2.6. Вложения.	2	1	1	
2.7.	Тема 2.7. Кортежи.	2	1	1	
2.8.	Тема 2.8. Словари	2	1	1	
2.9.	Тема 2.9. Множества.	2	1	1	
2.10.	Тема 2.10. Файлы.	2	1	1	
2.11.	Тема 2.11. Групповая практическая работа.	6	1	5	Практическая работа
3	Раздел 3. Знакомство с функциями	16	5	11	
3.1.	Тема 3.1. Функции.	4	1	3	
3.2.	Тема 3.2. Области видимости.	2	1	1	
3.3.	Тема 3.3. Передача параметров в функции.	2	1	1	
3.4.	Тема 3.4. Расширенные возможности функций.	4	1	3	
3.5.	Тема 3.5. Групповая	4	1	3	Практическая

	практическая работа				ская работа
4	Раздел 4. Работа с Модулями	38	11	27	
4.1.	Тема 4.1. Модули.	2	1	1	
4.2.	Тема 4.2. Знакомство с модулями Python. Часть №1 (random)	4	1	3	
4.3.	Тема 4.3. Знакомство с модулями Python. Часть №2 графика. Tkinter. PyQt5	6	1	5	
4.4.	Тема 4.4. Групповая практическая работа. Создаем игру «Сапёр» на PyQt5	4	1	3	
4.5.	Тема 4.5. Знакомство с модулями Python. Часть №3 анимация. Знакомство с модулем pygame.	2	1	1	
4.6.	Тема 4.6. Геометрические примитивы в pygame. Знакомство с анимацией.	2	1	1	
4.7.	Тема 4.7. Групповая практическая работа. Летающий круг.	4	1	3	
4.8.	Тема 4.8. Знакомство с модулями Python Часть №4 (морфология). Морфологический анализатор русского языка модуль rymorphy2.	2	1	1	
4.9.	Тема 4.9. Морфологический анализ текста с помощью rymorphy2.	4	1	3	
4.10.	Тема 4.10. Знакомство с модулями Python. Часть №5 работа с документами. Модуль python - docx	4	1	3	
4.11.	Тема 4.11. Групповая практическая работа.	4	1	3	Практическая работа
5	Раздел 5. Введение в ООП	34	9	25	
5.1.	Тема 5.1. Введение в ООП.	2	1	1	
5.2.	Тема 5.2. Основы написания классов.	6	3	3	
5.3.	Тема 5.3. Проектирование и разработка классов.	6	2	4	
5.4.	Тема 5.4. Групповая практическая работа	4	1	3	
5.5	Тема 5.5. Разработка проекта.	16	2	14	Защита проекта
	Итого	144	48	96	

3. СОДЕРЖАНИЕ ДООП

Раздел 1. Введение в программирование

Тема 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.

Тема 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.

Тема 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.

Тема 1.4. Обработка данных.

Теоретическая работа: Знакомство с различными способами ввода данных.

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

Тема 1.5. Построение командной работы.

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

Тема 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.

Тема 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

Тема 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.

Тема 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.

Тема 1.10. Вложенные циклы.

Теоретическая работа: Знакомство с вложенными циклами.

Тема 1.11. Групповая работа. Решение задач.

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

Раздел 2. Базовые конструкции в Python

Тема 2.1 Знакомство с коллекциями.

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

Тема 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.

Тема 2.3. Срезы в строках.

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

Практическая работа: решение задач на извлечение среза из строки.

Тема 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.

Тема 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.

Тема 2.6. Вложения.

Теоретическая работа: Строки, вложенные в списки, и списки, вложенные в списки, особенности индексации. Как обработать матрицу с помощью вложений.

Тема 2.7. Кортежи.

Теоретическая работа: что такое кортежи, как и для чего их используют. В чем отличие от списков.

Практическая работа: Решение задач на использование кортежей.

Тема 2.8. Словари.

Теоретическая работа: Способы создания словарей. Генераторы словарей. Ключи. Способы создания. Операции над словарями. Генераторы словарей. Методы: Clear, сору и др.

Практическая работа: Выполнение заданий по созданию словарей. Поиску по ключу, изменение и вывод словарей.

Тема 2.9. Множества.

Теоретическая работа: что такое множество, методы множества.

Практическая работа: Решение задач на создание, преобразование, удаление коллекций. Использование циклов для получения данных из коллекций.

Тема 2.10. Файлы.

Теоретическая работа: Способы создания файла. Экспорт данных из файла. Чтение файла. Сохранение в файл.

Практическая работа: Получение exe файла из проекта в Pycharm с помощью pyinstaller.

Тема 2.11. Групповая практическая работа.

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

Раздел 3. Знакомство с функциями

Тема 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.

Тема 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.

Тема 3.3. Передача параметров в функции.

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

Тема 3.4. Расширенные возможности функций.

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

Тема 3.5. Групповая практическая работа.

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

Раздел 4. Работа с модулями

Тема 4.1. 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.

Тема 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.

Тема 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.

Тема 4.4. Групповая практическая работа. Создаем игру «Сапёр» на PyQt5.
Практическая работа: Создание игры «Сапер».

Тема 4.5. Знакомство с модулями Python. Часть №3 анимация. Знакомство с модулем pygame.

Теоретическая работа: Знакомство с модулем Pygame, функции модуля,

Тема 4.6. Геометрические примитивы в pygame. Знакомство с анимацией.

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

Тема 4.7. Групповая практическая работа. Летающий круг.

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

Тема 4.8. Знакомство с модулями Python Часть №4 (морфология).

Морфологический анализатор русского языка модуль rymorphy2.

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

Тема 4.9. Морфологический анализ текста с помощью rymorphy2.

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

Тема 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.

Тема 4.11. Групповая практическая работа.

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

Раздел 5. Введение в ООП

Тема 5.1. Введение в ООП.

Теоретическая работа: знакомство с особенностями ООП в общем и его реализацией в языке Python. Классы. Методы. Вызов методов.

Тема 5.2. Основы написания классов.

Теоретическая работа: Знакомство с наследованием параметров класса. Оператор class. Наследование. Дерево атрибутов.

Тема 5.3. Проектирование и разработка классов.

Теоретическая работа: Знакомство с дополнительными функциями и методами при создании классов.

Практическая работа: Выполнение заданий на написание классов.

Тема 5.4. Групповая практическая работа.

Практическая работа: Решение задач

Тема 5.5 Разработка проекта.

Теоретическая работа: Распределение по группам, выбор темы проекта.

Практическая работа: Выполнение проекта по выбранной теме. Подготовка к защите проекта. Защита проекта.

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:

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.А. Комаров 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».

1. Что будет напечатано?

```
kvps = {"user", "bill", "password", "hillary"}  
print(kvps['password'])
```

- a. user
- b. bill
- c. password
- d. ничего. TypeError

2. Что делает следующий код?

```
def a(b, c, d): pass
```

- a. Определяет список и инициализирует его.
- b. Определяет функцию, которая ничего не делает
- c. Определяет функцию, которая передает параметры.
- d. Определяет пустой класс.

3. Что будет напечатано при исполнении следующего кода?

```
print(type(1/2))
```

- a. type 'int'
- b. type 'number'
- c. type 'double'
- d. type 'float'

4. Что будет напечатано?

```
name = "snow storm"  
print("%s" % name[6:8])
```

- a. st
- b. sto
- c. to
- d. Syntax Error

5. Что будет напечатано?

```
x = True  
y = False  
z = False  
if not x or y:  
    print(1)  
elif not x or not y and z:  
    print(2)  
elif not x or y or not y and x:  
    print(3)  
else:  
    print(4)
```

- a. 1
- b. 2
- c. 3
- d. 4

6. Что выведет следующий фрагмент кода

```
x = 4.5  
y = 2
```

```
print(x // y)
```

- a.
- b. 2.25
- c. 9
- d. 20.25
- e. 21

7. Что выведет следующая программа?

```
a = [1,2,3,None(),[],]
```

```
print(len(a))
```

- a. Syntax Error
- b. 4
- c. 5
- d. 6