


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

СОГЛАСОВАНО:

Председатель методического  
совета

 Н.В. Грицюк  
протокол № 12  
от «3» июня 2024 г.

УТВЕРЖДАЮ:

Директор филиала

 Е. А. Дыптан  
Приказ № 02-02-59  
от «3» июня 2024 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2024 – 2025 уч. год  
технической направленности  
«Python 3 для новичков. Продвинутый Python»**

Форма реализации программы – очная  
Срок реализации – 1 год  
Возраст обучающихся – 15-17 лет

Составитель программы:  
Теников Лев Евгеньевич,  
Комаров Егор Андреевич,  
педагог дополнительного образования

г. Норильск, 2024 г.

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Дополнительная общеобразовательная общеразвивающая программа «Python 3 для новичков. Продвинутый Python» имеет техническую направленность и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года утвержденной распоряжением Правительства Российской Федерации от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи от 28.09.2020 г. № 28.

Уровень программы – базовый. Программа предполагает углубленное изучение алгоритмов и закрепление умений использования языка программирования Python в объектно-ориентированной методологии. Программа является логическим продолжением (вторым годом обучения) курса «Python 3 для новичков. Основы Python».

Отличительной особенностью Python являются простота освоения и высокая скорость разработки программ. Также достоинством языка Python является большое количество готовых к использованию библиотек в различных областях: анализ данных и машинное обучение, научные вычисления, визуализация, сетевое программирование. В процессе программирования дети получают дополнительное образование в области электроники и информатики, а также знания в области технического английского языка.

Программирование развивает математическое и алгоритмическое мышление, способность к абстракции и умение логически связывать объекты разной природы в сочетании, представить всё, что угодно, из нуля и единиц. Данная ДООП направлена на предпрофессиональную ориентацию для тех, кто хочет связать свою трудовую деятельность с информационными технологиями, с развитием «цифровой экономики» и «цифрового общества».

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, для участия в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

### **1.1 НОВИЗНА ДООП**

Школьники, прошедшие обучение по данной программе, получают знания, умения и навыки, необходимые для сознательного выбора в будущем профессии, связанной с программированием. Программа позволяет познакомиться и понять тонкости профессии программист и профессий

связанных с разработкой IT-приложений, оценить себя в этой сфере, выработать профессионально важные качества. Существенной особенностью программы является её направленность на формирование у воспитанников потребности в инженерном образовании и научном исследовании.

Инновационной является авторская методика, позволяющая сочетать изучение достаточно сложного языка программирования с исследовательской деятельностью и разработкой проектов самых разнообразных по содержанию, позволяющая ребенку в конце курса обучения почувствовать себя настоящим профессионалом.

## **1.2 АКТУАЛЬНОСТЬ ДООП**

Актуальность предлагаемой программы заключается в том, что она ориентирована на приоритетные направления социально-экономического и территориального развития Красноярского края, определенных в Стратегии социально-экономического развития Красноярского края на период до 2030 года (утверждена постановлением Правительства Красноярского края от 24.12.2015 № 9-4112), в которой поставлена задача качественного изменения структуры направленностей дополнительного образования и увеличения кружков и секций технического профиля.

Актуальность программы обусловлена потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области программирования, а также высоким интересом подростков к IT-сфере.

Содержание программы отвечает современным потребностям детей и родителей в получении необходимых для личностного развития и успешной адаптации к требованиям жизни в современном информационном обществе теоретических знаний и практических навыков в сфере программирования. Предлагаемая программа «Python 3 для новичков. Продвинутый Python» способствует формированию математического склада ума, развитию аналитического мышления, что является важным аспектом для подрастающего поколения.

## **1.3 ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ**

Педагогическая целесообразность программы заключается в том, что содержание программы, используемые технологии, формы и методы обучения создают и обеспечивают необходимые условия для личностного развития и творческого труда обучающихся и позволяют удовлетворить индивидуальные потребности обучающихся в интеллектуальном развитии. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации учащихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Данная программа способствует развитию 4к-компетенций детей

(коммуникация, креативность, командная работа, критическое мышление), тем самым отвечая потребностям общества и государственным образовательным стандартам. В программе реализуются системный, комплексный, личностно-ориентированный и теоретический подходы к развитию детей. Адаптация материала соответствует возрастным и психофизиологическим особенностям обучающихся.

#### **1.4 ЦЕЛЬ ДООП**

**Цель** программы — углубленное развитие алгоритмического и логического мышления учащихся 15-17 лет в процессе программирования на языке Python.

##### **Основные задачи:**

- закрепить у обучающихся знания о возможностях, синтаксисе, технологиях языка Python, познакомить с принципами и методами структурного, функционального и объектно-ориентированного программирования, с основными структурами данных и типовыми методами обработки этих структур;
- способствовать развитию базовой части математического аппарата, применяемого в современном программировании;
- углубить знания и умение применять основные базовые алгоритмические конструкции, методы программирования для решения прикладных математических и информационных задач;
- формировать умения работы на компьютере с использованием интегрированной среды IDLEPython;
- отработать навыки использования различных библиотечных модулей Python;
- воспитать целеустремленность и упорство в достижении результата в процессе решения учебных задач;
- развить творческое мышление и воображение, умение излагать мысли в четкой логической последовательности;
- сформировать умение отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений и информационного поиска.
- приобщить к проектно-творческой деятельности, развить умения планирования проекта, умение работать в группе.

#### **1.5 ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП**

Программа «Python 3 для новичков. Продвинутый Python» ориентирована на обучающихся от 15 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 знать синтаксис языка Python, уметь писать функции, понимать принципы объектно-ориентированного программирования. Обязательно предварительное освоение курса по программе «Python 3 для новичков. Основы Python».

При приеме в группы продвинутого уровня существует отбор в форме проведения входного тестирования, которое включает в себя задачи по использованию простых алгоритмических конструкций. Предусмотрена возможность начать обучение без знания языка программирования, но с хорошей математической подготовкой (умение решать нестандартные задачи по математике в объеме основного общего образования).

### **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 в очной форме. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час 40 минут) с обязательным перерывом.

Программное содержание, методы, формы, средства обучения отбирались с учетом основных направлений развития дополнительного образования, отраженных в Концепции развития дополнительного образования детей (распоряжение Правительства Российской Федерации от 31 марта 2022 г. № 678-р).

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

Для решения поставленных задач на всех этапах обучения каждая тема раздела 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 практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения. Основные задания являются обязательными для выполнения всеми обучающимися группы. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый обучающийся получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Для такой 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 разделам для определения уровня знаний обучающихся. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса. По типу организации взаимодействия педагогов с обучающимися при реализации программы используются лично ориентированные технологии, технологии сотрудничества.

### **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;
- 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;
- базовые конструкции в Python.

#### **Предметные результаты:**

- закреплены знания о возможностях, синтаксисе, технологиях языка Python, изучены принципы и методы структурного, функционального и объектно-ориентированного программирования, основные структуры данных и типовые методы обработки этих структур;
- освоены основные базовые алгоритмические конструкции и методы программирования для решения прикладных математических и информационных задач;
- сформированы умения работы на компьютере с использованием интегрированной среды IDLEPython;
- отработаны навыки использования различных библиотечных модулей Python.

#### **Личностные результаты:**

- воспитана целеустремленность и упорство в достижении результата в процессе решения учебных задач;
- сформирована самостоятельность и творческий подход к решению

задач с использованием средств вычислительной техники.

**Метапредметные результаты:**

- развито творческое мышление и воображение, умение излагать мысли в четкой логической последовательности;
- сформировано умение отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений и информационного поиска.
- сформированы умения поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач.

**2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1	Введение.	2	2		
<b>Раздел 1. Базовое программирование (26 часов)</b>					
2	Запуск среды и основные горячие клавиши	2	1	1	
3	Условный оператор	4	2	2	
4	Простые встроенные функции	2	1	1	
5	Задачи с циклом while	4	2	2	
6	Отладчик	2	1	1	
7	Задачи циклом for	4	2	2	
8	True и False, break и continue	2	1	1	
9	Вложенные циклы	4	2	2	
10	Текущий контроль по разделу «Базовое программирование»	2	0	2	Контрольная работа
<b>Раздел 2. Основные конструкции в Python (40 часов)</b>					
11	Множества	2	1	1	
12	Строки	4	2	2	
13	Решение задач	2	0	2	
14	Списки	4	2	2	
15	Кортежи	2	1	1	

16	Преобразование коллекций	2	1	1	
17	Методы split и join. Списочные выражения	2	1	1	
18	Методы списков и строк	4	2	2	
19	Решение задач	4	0	4	
20	Вложенные списки	2	1	1	
21	Словари	2	1	1	
22	Практика по работе с tkinter	2	1	1	
23	Решение задач	4	1	3	
24	Текущий контроль по разделу «Базовые конструкции в Python»	2	0	2	Контрольная работа
25	Подведение итогов	2	2	0	
<b>Раздел 3. Решение прикладных задач в Python (76 часов)</b>					
26	Функции	2	1	1	
27	Функции. Возвращение значений из функции	2	1	1	
28	Области видимости переменных	2	1	1	
29	Решение задач	4	1	3	
30	Функции. Передача параметров	2	1	1	
31	Функции с переменным числом аргументов	2	1	1	
32	Рекурсия	2	1	1	
33	Функции как объект. Лямбда-функции	2	1	1	
34	Переопределение функций и декораторы	4	2	2	
35	Обработка коллекций. Поточковый ввод sys.stdin	2	1	1	
36	Итераторы и коллекции	2	1	1	
37	Решение задач	4	1	3	



38	Библиотеки Python. Часть №1 (Модули стандартной библиотеки)	4	2	2	
39	Библиотеки Python. Часть №2 (графика)	4	2	2	
40	Библиотеки Python. Часть №3 (звук)	2	1	1	
41	Библиотеки Python. Часть №4 (морфология)	2	1	1	
42	Библиотеки Python. Часть №5 (документы)	2	1	1	
43	Библиотеки Python. Часть №6 (Numpy)	2	1	1	
44	Решение задач	4	1	3	
45	Введение в ООП	2	1	1	
46	Введение в ООП. Полиморфизм	2	1	1	
47	ООП. Определение операторов	2	1	1	
48	ООП. Наследование	4	2	2	
49	Проектирование и разработка классов	8	4	4	
50	Решение задач	4	1	3	
51	Текущий контроль по разделу «Решение прикладных задач в Python»	2	0	2	Контрольная работа
52	Подведение итогов	2	2	0	
Итого часов:		144	64	80	

### 3. СОДЕРЖАНИЕ ДООП

#### **Введение:**

Теория (2 ч): Вводная лекция, знакомство с учениками. Проведение инструктажа по технике безопасности. Знакомство с историей языков программирования.

#### **Раздел 1. Базовое программирование (26 часов).**

### **Тема 1.1. Запуск среды и основные горячие клавиши (2 часа).**

Теория (1 ч): Иерархическое устройство языка Python. Соглашение, о том, как писать код для языка Python пер8. Горячие клавиши.

Практика (1 ч): Использование функций input и print. Использование умножения, деления, сложения, вычитания с переменными одного типа, переменными разных типов.

### **Тема 1.2. Условный оператор (4 часа).**

Теория (2 ч): Логические функции и/или/не, использование логических элементов в ветвлении. Правила оформления блоков ветвления с помощью отступов.

Практика (2 ч): Создание ветвлений, использование множественных ветвлений при решении задач.

### **Тема 1.3. Простые встроенные функции (2 часа).**

Теория (1 ч): Основные типы данных. Правила объявления переменных. Арифметические действия с переменными.

Практика (1 ч): Использование умножения, деления, сложения, вычитания с переменными одного типа, переменными разных типов.

### **Тема 1.4. Задачи с циклом while (4 часа).**

Теория (2 ч): Понятие цикла. Общий вид цикла.

Практика (2 ч): Практическое использование цикла While.

### **Тема 1.5. Отладчик (2 часа).**

Теория (1 ч): Отладчик среды разработки. Работа с точками останова. Анализ выполнения программ с помощью debugger.

Практика (1 ч): Разбор примеров работы программ с циклами и условными операторами. Примеры классических ошибок, способы их решения.

### **Тема 1.6. Задачи с циклом for (4 часа).**

Теория (2 ч): Цикл for. Понятие итерируемого объекта. Перебор итерируемого объекта с помощью цикла for.

Практика (2 ч): Решение задач на перебор объектов. Использование аргументов при вызове функции range. Создание возрастающих и убывающих последовательностей с различным шагом.

### **Тема 1.7. True и False, break и continue (2 часа).**

Теория (1 ч): Знакомство с командами остановки и прерывания цикла break, continue. Преимущества и недостатки использования этих команд.

Практика (1 ч): Апробация полученных знаний на практике при решении задач.

### **Тема 1.8. Вложенные циклы (4 часа).**

Теория (2 ч): Вложенные циклы. Способы работы с данными разной степени вложенности. Знакомство с матрицами. Примеры применения матриц при обработке данных.

Практика (2 ч): Решение задач на работу с данными разной степени вложенности. Построение матриц.

### **Тема 1.9. Текущий контроль по разделу «Базовое программирование»**

Практика (2 ч): Решение задач по вариантам. Контроль знаний по пройденным темам.

## **Раздел 2. Основные конструкции в Python (40 часов).**

### **Тема 2.1. Множества (2 часа).**

Теория (1 ч): Знакомство с коллекциями. Что такое множество, методы множества. Особенности множества. Взаимодействие множества с множеством.

Практика (1 ч): Решение задач используя множества. Хранение данных в множестве.

### **Тема 2.2. Строки (4 часа).**

Теория (2 ч): Строки. Индексация. Срезы строк. Операции над строками. Конкатенация. Повторение.

Практика (2 ч): Решение задач на работу со строками. Использование функции len для определения количества объектов в контейнере.

### **Тема 2.3. Решение задач (2 часа).**

Практика (2 ч): Создание форм для анкет. Создание программы для регистрации и авторизации пользователя.

### **Тема 2.4. Списки (4 часа).**

Теория (2 ч): Списки. Создание списков. Индексация списков. Срезы списков. Данные внутри списков.

Практика (2 ч): Решение задач по сортировке и обработке информации внутри списков. Использование списков для решения прикладных задач.

### **Тема 2.5. Кортежи (2 часа).**

Теория (1 ч): Знакомство с кортежами. Способы создания кортежа. Отличие кортежа от списка. Области применения кортежа.

Практика (1 ч): Использование массового присваивания, использование кортежа и оператора "\*" для запаковки и распаковки итерируемых объектов.

### **Тема 2.6. Преобразование коллекций (2 часа).**

Теория (1 ч): Способы преобразования коллекций. Классические примеры. Области применения.

Практика (1 ч): Использование преобразования коллекций для доработки программы по созданию анкеты.

### **Тема 2.7. Методы split и join. Списочные выражения (2 часа).**

Теория (1 ч): Повторение способов перевода между типа контейнеров. Разделение строки по словам. Указание разделителя. Склеивание списка строк в строку. Указание элемента для объединения строковых элементов.

Практика (1 ч): Получение данных от пользователя. Сортировка полученных данных по одному из элементов.

### **Тема 2.8. Методы списков и строк (4 часа).**

Теория (2 ч): Методы классов данных. Синтаксис методов.

Практика (2 ч): Применение методов списков и методов строк при решении задач. Поиск элемента в строке. Преобразование строки и списка.

### **Тема 2.9. Решение задач (4 часа).**

Практика (4 ч): Решение задач на работу со строками и списками.

### **Тема 2.10. Вложенные списки (2 часа).**

Теория (1 ч): Повторение способов работы со списками. Двойная индексация. Глубина вложений.

Практика (1 ч): Работа с вложенными списками на примере игр «морской бой» и «шахматы».

### **Тема 2.11. Словари (2 часа).**

Теория (1 ч): Способы создания словарей. Генераторы словарей. Ключи. Способы создания. Операции над словарями. Генераторы словарей. Методы: Clear, сору и др.

Практика (1 ч): Выполнение заданий по созданию словарей. Поиску по ключу, изменение и вывод словарей.

### **Тема 2.12. Практика по работе с tkinter (2 часа).**

Теория (1 ч): Библиотека tkinter. Создание бесконечного цикла внутри приложения. Создание кона. Знакомство с функциями.

Практика (1 ч): Написание программы с интерфейсом.

### **Тема 2.13. Решение задач (4 часа).**

Теория (1 ч): Разделение группы на команды. Знакомство с командной работой. Распределение обязанностей в команде. Выдача группового задания.

Практика (3 ч): Решение поставленной задачи в группе.

**Тема 2.14.** Текущий контроль по разделу «Основные конструкции в Python» (2 часа).

Практика (2 ч): Решение задач по вариантам. Контроль знаний по пройденным темам.

**Тема 2.15.** «Подведение итогов» (2 часа).

Теория (2 ч): Разбор задач. Примеры решений задач. Разбор ошибок.

### **Раздел 3. Решение прикладных задач в Python (76 часов).**

**Тема 3.1.** Функции (2 часа).

Теория (1 ч): Знания о качестве кода. Правил оформления кода. Основные паттерны программирования. Разделение кода на подпрограммы. Понятие функции. Особенности работы функций.

Практика (1 ч): Решение задач групповой работы с помощью функций.

**Тема 3.2.** Функции. Возвращение значений из функции (2 часа).

Теория (1 ч): Как возвращать значение из функции. Возврат значений из функции. Инструкции return, def.

Практика (1 ч): Применение полученных знаний для решения практических задач.

**Тема 3.3.** Области видимости переменных (2 часа).

Теория (1 ч): Области локальной и глобальной видимости, области нелокальной видимости. Операторы global, nonlocal,

Практика (1 ч): Использование инструкций global и nonlocal, при решении задач с функциями.

**Тема 3.4.** Решение задач (4 часа).

Теория (1 ч): Распределение группы по командам. Выдача заданий.

Практика (3 ч): Решение поставленных задач в группе. Создание консольного интерфейса для создания списка учеников.

**Тема 3.5.** Функции. Передача параметров (2 часа).

Теория (1 ч): Отличие между переменной и значением. Функции, изменяющие значение аргумента. Id Объектов. Изменяемость и неизменяемость объектов.

Практика (1 ч): Написание программ с получением функцией различных значений, и преобразование значений согласно коду программы.

**Тема 3.6.** Функции с переменным числом аргументов (2 часа).

Теория (1 ч): Распаковка и упаковка значений. Аргументы по умолчанию. Именованные аргументы. Инструкция pass. Согласованность аргументов.

Практика (1 ч): Работа с функциями. Получение неизвестного количества именованных и неименованных аргументов.

### **Тема 3.7. Рекурсия (2 часа).**

Теория (1 ч): Факториал и число сочетаний. Расчет факториала с помощью рекурсии. Глубина рекурсии. Декларативное программирование. Ошибки при использовании рекурсии. Деревья рекурсии.

Практика (1 ч): Построение последовательности Фибоначи с помощью рекурсии. Подсчет длины списка с помощью рекурсии.

### **Тема 3.8. Функции как объект. Лямбда-функции (2 часа).**

Теория (1 ч): Функция как объект. Функция `filter`, Лямбда-функции, Функция `map`. Списочные выражения. Поточковый ввод.

Практика (1 ч): Применение функций высшего порядка для решения задач.

### **Тема 3.9. Переопределение функций и декораторы (4 часа).**

Теория (2 ч): Переопределение функций. Согласованность аргументов. Функции внутри функций. Декораторы. Нелокальные переменные.

Практика (2 ч): Создание логгера для функции.

### **Тема 3.10. Обработка коллекций. Поточковый ввод `sys.stdin` (2 часа).**

Теория (1 ч): Итерируемые объекты. Функции `max`, `min`, `sorted` и использование ключей сортировки. Функции `any` и `all`. Поточковый ввод `sys.stdin`.

Практика (1 ч): Решение задач на поиск информации в потоке вводимых данных.

### **Тема 3.11. Итераторы и коллекции (2 часа).**

Теория (1 ч): Итерируемый объект. Итератор. Способы работы с итератором. Методы `__iter__` и `__next__`.

Практика (1 ч): Написание собственного итератора и генератора.

### **Тема 3.12. Решение задач (4 часа).**

Теория (1 ч): Распределение учебной группы по командам. Постановка задачи группе. Разделение задачи на подзадачи. Распределение подзадачи внутри группы.

Практика (3 ч): Выполнение поставленной задачи в группе.

### **Тема 3.13. Библиотеки Python. Часть №1 (Модули стандартной библиотеки) (4 часа).**

Теория (2 ч): Способы установки модулей. Импорт функций из модулей. Импорт всех функций из модуля, перезагрузка модуля.

Знакомство со встроенным модулем Random, основные функции генератора случайностей. Работа с библиотекой datetime.

Практика (2 ч): Разбор функций генератора чисел, написание программ с использованием генератора чисел. Использование библиотеки datetime для работы с датой и временем.

### **Тема 3.14. Библиотеки Python. Часть №2 (графика) (4 часа).**

Теория (2 ч): Растровые изображения. Библиотека PIL. Создание изображения и рисование. Типы фильтров изображений.

Практика (2 ч): Создание функции для работы с изображениями.

### **Тема 3.15. Библиотеки Python. Часть №3 (звук) (2 часа).**

Теория (1 ч): Теория звука. Работа с аудиофайлами. Сырые аудиоданные.

Практика (1 ч): Ускорение записи. Убрать тишину в записи.

### **Тема 3.16. Библиотеки Python. Часть №4 (морфология) (2 часа).**

Теория (1 ч): Морфологический анализатор rymorphu2. Функции. Разбор слов. Работа с тегами. Постановка слов в начальную форму. Согласование с числительными.

Практика (1 ч): Практическая задача на морфологический разбор текста, поиск слова в тексте, поиск ошибок в тексте.

### **Тема 3.17. Библиотеки Python. Часть №5 (документы) (2 часа).**

Теория (1 ч): Знакомство с функциями модуля python-docx. Работа с документами. Создание текстового документа. Работа с документом как с шаблоном. Создание таблиц excel.

Практика (1 ч): Написание программы для автоматического заполнения документов.

### **Тема 3.18. «Библиотеки Python. Часть №6 (Numpy)» (2 часа).**

Теория (1 ч): Измерение скорости работы приложений. Массивы в NumPy. Размерность массива. Индексация в массивах. Массовые операции. Матрицы. Сортировка.

Практика (1 ч): Создание игры «змейка».

### **Тема 3.19. Решение задач (4 часа).**

Теория (1 ч): Распределение группы по командам. Выдача заданий.

Практика (3 ч): Решение поставленных задач в группе. Создание консольного интерфейса для создания списка учеников.

### **Тема 3.20. Введение в ООП (2 часа).**

Теория (1 ч): Основные понятия в ООП. Создание классов. Методы классов. Инкапсуляция.

Практика (1 ч): Создание класса для построения карты игры «Морской бой».

**Тема 3.21. Введение в ООП. Полиморфизм (2 часа).**

Теория (1 ч): Полиморфизм - свойства кода работать с разными типами данных. Утиная типизация. Проверка типа объекта.

Практика (1 ч): Создание классов Table для обработки двумерных таблиц.

**Тема 3.22. ООП. Определение операторов (2 часа).**

Теория (1 ч): Специальные методы. Переопределение поведения объекта в функции print. Метод `_repr_`. Знакомство с другими специальными методами.

Практика (1 ч): Создание классов с различными способами обработки данных.

**Тема 3.23. ООП. Наследование (4 часа).**

Теория (2 ч): Основные паттерны ООП. Иерархия классов. Механизм наследования. Перегрузка параметров класса. Функция `super`. Множественное наследование.

Практика (2 ч): Создание классов для социальной сети. Создание сложных классов для компьютерной игры.

**Тема 3.24. Проектирование и разработка классов (8 часа).**

Теория (4 ч): Создание зависимостей между классами. Выбор способа взаимодействия между классами.

Практика (4 ч): Создание игры «Шахматы».

**Тема 3.25. Решение задач (4 часа).**

Теория (1 ч): Распределение по группам, выбор темы групповой работы.

Практика (3 ч): Выполнение работы по выбранной теме. Защита работы.

**Тема 3.26. Текущий контроль по разделу «Решение прикладных задач в Python» (2 часа).**

Практика (2 ч): Решение задач по вариантам. Контроль знаний по пройденным темам.

**Тема 3.27. Подведение итогов (2 часа)**

Теория (2 ч): Разбор задач. Примеры решений задач. Разбор ошибок.

#### **4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

Для проведения занятий в кабинете необходим следующий состав аппаратного и программного обеспечения:

1. Учебный кабинет, удовлетворяющий санитарно-гигиеническим требованиям, для занятий группы 12 человек (компьютеры или ноутбуки, парты, стулья).



2. Рабочее место преподавателя, укомплектованное компьютером, компьютерным столом, стулом.
3. Периферийное оборудование: МФУ, колонки, проектор.
4. Канцелярские принадлежности: ручки, карандаши, маркеры, файлы, папки.
5. Дополнительное оборудование помещения: шкаф, тумба, маркерная доска.

## **5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

### **Основная литература:**

1. Николаевич, В. А. Программирование на Python в примерах и задачах/ В.А. Федоров. – Москва: Бомбара, 2023, - 616 с.
2. Лутц, Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. – Спб.: ООО «Диалектика», 2020. – 832 с.
3. Лутц, Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. – Спб.: ООО «Диалектика», 2020. – 720 с.

### **Электронные ресурсы:**

1. Официальный сайт проекта. – <https://www.python.org/>
2. Официальная документация. – <https://docs.python.org/>
3. Список правил и оформления кода pep. – <https://peps.python.org/>
4. Пример уроков по python. - <https://devpractice.ru/python-lessons/>
5. Среда разработки для python. – <https://www.jetbrains.com/pycharm/>

## 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».

**Теников Л. Е.**, педагог дополнительного образования, студент Федерального государственного бюджетного образовательного учреждения высшего образования «Заполярный государственный университет им. Н.М. Федоровского».

## 7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### Текущий контроль по разделу «Введение в программирование»

1. Во время отлива на берегу оказалось некоторое количество водорослей. Первая артель собрала в 3 раза больше, чем вторая, а вторая в 2 раза больше, чем третья. Оставшиеся водоросли собрали местные жители для себя. Определите и выведите в строку через пробел, сколько водорослей осталось местным жителям, сколько 1, 2 и 3 артелям.

#### Формат ввода

Целое число – количество водорослей.

#### Формат вывода

Четыре числа: сколько водорослей останется местным жителям, сколько собрала 1 артель, 2 артель и 3 артель. Числа записаны в одну строку через пробел.

2. Напишите программу, которая, зная скорости передвижения подозреваемых, определяет, кто из них мог совершить преступление.

#### Формат ввода

Скорость перемещения (миль в час) человека (действительное число)

Скорость 1 подозреваемого (действительное число)

Скорость 2 подозреваемого (действительное число)

Расстояние в милях (целое число)

Время в минутах (целое число)

#### Формат вывода

В одной строке через пробел вывести всех подозреваемых. Если никто не подозревается, то есть находятся на расстоянии меньшем, чем могли бы, будь они преступниками, вывести: **Не они**.

3. Существует туристический язык. Пусть правила будут такими:

1. длина слова не больше 13 символов;
2. в словаре слово находится между **беззар** и **хлябь** (включая их);

3. если в нем есть сочетание **стан**, то количество букв обязательно должно быть нечетным;
4. ну и не должно быть сочетания **мир**.

Если все условия выполнены, нужно вывести **МОЖЕТ**, иначе **НЕ МОЖЕТ**.

#### **Формат ввода**

Слово в нижнем регистре.

#### **Формат вывода**

**МОЖЕТ** или **НЕ МОЖЕТ**.

### **Текущий контроль по разделу «Базовые конструкции в Python»**

1. Напишите программу, которая ищет "те" камни, то есть слова, в которых есть заданная буква.

#### **Формат ввода**

Вводится буква, которую нужно найти в словах. Затем вводятся слова, пока не будет введено слово *END*.

#### **Формат вывода**

Выведите самое длинное слово из введенных, в котором есть заданная буква. Если есть несколько одинаково длинных слов, то вывести первое. Хотя бы одно такое слово точно есть.

2. Напишите программу, выбирающую предпочтительное направление поворота в заданном мире.

#### **Формат ввода**

Вводятся два числа  $a$  и  $b$  – границы мира, затем вводится количество строк с числами, записанными через один или несколько пробелов, затем сами строки.

#### **Формат вывода**

Из каждой строки выбрать числа, попадающие в диапазон  $[a; b)$ , и вывести удвоенные значения этих чисел через пробел в исходном порядке.

3. Как отличить мегалиты от обычных булыжников? Довольно просто, если знать секрет. (Мегалиты кратны сразу нескольким числам.) Напишите программу, выбирающую мегалиты.

#### **Формат ввода**

Вводится строка чисел для проверки кратности через  $\#$ . Затем вводится строка чисел через пробел – это глыбы, среди которых нужно выбрать мегалиты.

#### **Формат вывода**

Из второй строки выберите числа, кратные всем числам из первой строки, и выведите их без повторений в возрастающем порядке по одному в строке.

### **Текущий контроль по разделу «Решение прикладных задач в Python»**

1. Напишите программу, собирающую словарь из того, что могут сделать для людей экзотические существа.

## Формат ввода

Вводятся строки вида:

*Существо - что может сделать.*

Существа не повторяются.

## Формат вывода

Выведите словарь, ключи которого существа, а значения – то, что они могут сделать.

2. Напишите функцию для выбора самого дальнего объекта, который виден в таком тумане.

Функция `swamp_fog()` принимает произвольное количество аргументов-кортежей (*объект, расстояние*) и именованный аргумент *farest* со значением по умолчанию 13. Функция возвращает количество объектов, расстояние до которых не превышает значения *farest*, и расстояние до самого далекого из них.

3. Напишите два класса:

**Settler** (Поселенец) и его наследник **Swamper** (Болотник).

Поселенец (**Settler**) инициализируется с двумя аргументами – именем и местом жительства. Умеет печатать информацию о себе: `__str__` – возвращает строку: *<Имя> (<имя класса>) from <место жительства>*.

Болотник (**Swamper**) инициализируется с одним аргументом – именем, и это понятно, ведь он живёт на болоте (место жительства *Swamp*). При инициализации список работ экземпляра класса пуст.

- Класс обеспечивает выполнение методов (*sw* – экземпляр класса):
- `sw.append(value)` – добавляет работу, которую научился делать;
- `sw[i]` – возвращает работу по индексу;
- `sw[i] = value` – меняет значение работы в списке по индексу;
- `sw.pop()` – удаляет работу по индексу, если индекс не передан, удаляется последняя работа в списке; возвращает удаляемое значение;
- `len(sw)` – возвращает длину списка работ; экземпляр класса можно вызвать без аргументов – возвращается строка имени, повторенная количество раз, равное длине списка работ.

**Задача 1.** Дана строка, состоящая из слов, разделенных пробелами.

Напишите программу, которая подсчитывает количество слов в этой строке.

### Формат входных данных

На вход программе подается строка.

### Формат выходных данных

Программа должна вывести количество слов в строке.

**Примечание.** Решите задачу в одну строчку кода.

### Sample Input 1:

Hello world!

### Sample Output 1:

2

**Sample Input 2:**

Timur forever young

**Sample Output 2:**

3

**Sample Input 3:**

The future belongs to those, who believe in beauty of their dreams

**Sample Output 3:**

12

Напишите программу. Тестируется через stdin → stdout

**Time Limit:** 15 секунд

**Memory Limit:** 256 MB

**Задача 2.** Дано пятизначное или шестизначное натуральное число. Напишите программу, которая запишет его последние пять цифр в обратном порядке.

**Формат входных данных**

На вход программе подается одно натуральное пятизначное или шестизначное число.

**Формат выходных данных**

Программа должна вывести число, которое получится в результате разворота, указанного в условии задачи. Число нужно выводить без незначащих нулей.

**Sample Input 1:**

12345

**Sample Output 1:**

54321

**Sample Input 2:**

987654

**Sample Output 2:**

945678

**Sample Input 3:**

25000

**Sample Output 3:**

52

**Sample Input 4:**

560000

**Sample Output 4:**

500006

Напишите программу. Тестируется через stdin → stdout

**Time Limit:** 15 секунд

**Memory Limit:** 256 MB

**Задача 3.** На вход программе подаётся натуральное число. Напишите

программу, которая вставляет в заданное число запятые в соответствии со стандартным американским соглашением о запятых в больших числах.

#### **Формат входных данных**

На вход программе

подаётся натуральное число  $n$ , ( $0 < n < 10100$ ).

#### **Формат выходных данных**

Программа должна вывести число с запятыми в соответствии с условием задачи.

##### **Sample Input 1:**

1000000

##### **Sample Output 1:**

1,000,000

##### **Sample Input 2:**

100

##### **Sample Output 2:**

100

##### **Sample Input 3:**

12345

##### **Sample Output 3:**

12,345

Напишите программу. Тестируется через stdin → stdout

**Time Limit:** 15 секунд

**Memory Limit:** 256 MB

#### **Задача 4. Роскомнадзор запретил букву а**

Необходимо написать программу, реализующую алгоритм написания [этой песни](#). Алгоритм выводит в конце предложения следующую в алфавитном порядке букву, если она встречается в строке текста, а очередную строку отображает уже без этой буквы.

#### **Формат входных данных**

На вход программе подается одно слово, записанное строчными русскими буквами без буквы "ё".

#### **Формат выходных данных**

Программа должна вывести в соответствии с указанным алгоритмом строки, количество которых равно количеству разных букв в строке, которая получается путем конкатенации введенного слова и строки "запретил букву".

**Примечание 1.** Текст исходной песни в первом тесте.

**Примечание 2.** Поем и решаем, друзья, поем и решаем.

##### **Sample Input 1:**

роскомнадзор

##### **Sample Output 1:**

роскомнадзор запретил букву а

роскомндзор зпретил букву б

роскомндзор зпретил укву в

роскомндзор зпретил уку д  
роскомнзор зпретил уку е  
роскомнзор зпртил уку з  
роскомнор пртил уку и  
роскомнор пртл уку к  
росомнор пртл уу л  
росомнор прт уу м  
росонор прт уу н  
росоор прт уу о  
рспр прт уу п  
рспр рт уу р  
с т уу с  
т уу т  
уу у

**Sample Input 2:**

тимур

**Sample Output 2:**

тимур запретил букву а  
тимур зпретил букву б  
тимур зпретил укву в  
тимур зпретил уку е  
тимур зпртил уку з  
тимур пртил уку и  
тмур пртл уку к  
тмур пртл уу л  
тмур прт уу м  
тур прт уу п  
тур рт уу р  
ту т уу т  
у уу у

Напишите программу. Тестируется через stdin → stdout

**Time Limit:** 15 секунд

**Memory Limit:** 256 MB

**Задача 5. Кремниевая долина**

Искусственный интеллект Антон созданный Гилфойлом взломал сеть умных холодильников. Теперь он использует их в качестве серверов "Пегого дудочника". Помогите владельцу фирмы отыскать все зараженные холодильники.

Для каждого холодильника существует строка с данными, состоящая из строчных букв и цифр, и если в ней присутствует слово "anton" (необязательно рядом стоящие буквы, главное наличие последовательности букв), то холодильник заражен и нужно вывести номер холодильника, нумерация начинается с единицы





возвращает целое число, цифры в котором переставлены в **обратном порядке**. Например, если введено число **561**, программа должна вернуть **165**, а если **-578**, то **-875**. Решите задачу двумя способами – с использованием методов строк и без. Какое решение более эффективно?

#### Задание 2

Вычислите частное от деления **x** на **y**, где **x** и **y** – целые положительные числа. **Допустимые** операции – сложение, вычитание и побитовый сдвиг.

#### Задание 3

Имеются текст **text** и подстрока **st**. Напишите программу, которая находит индекс **первого** вхождения **st** в **text**.

#### Задание 4

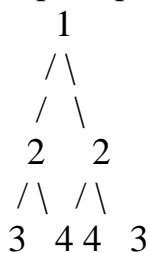
Напишите функцию для проверки симметричности двоичного дерева.

#### Задание 5

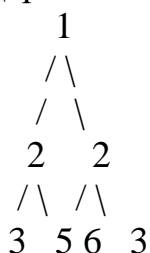
Напишите программу для подсчета количества правок, которые нужно выполнить, чтобы преобразовать строку S1 в строку S2. Например, для преобразования слова «лимузин» в «лимонад» нужно сделать 4 правки, а для приведения слова «кошка» к слову «кофта» достаточно 2 изменений.

**Задача:** Необходимо реализовать функцию, принимающую в аргументах строку, состоящую из букв и вернуть новую строку, в которой повторяющиеся буквы заменены количеством повторений. # AAAABVCSAA -> A4B2C1A2

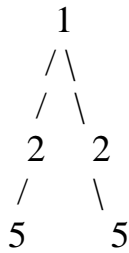
**Задача:** Напишите функцию для проверки симметричности двоичного дерева. Примеры деревьев: Дерево 1:



Дерево 2:



Дерево 3:



1. Пары чисел с минимальной разностью

Напишите функцию, которая будет принимать список чисел и возвращать два числа, абсолютная разность которых минимальна. Пару чисел нужно вернуть в виде списка, отсортированного по возрастанию.

2. Напишите функцию, которая будет принимать начальное слово и итоговую строку и возвращать уровень вложенности.

3. В предложение были добавлены лишние пробелы. Напишите функцию, которая будет принимать такое предложение и возвращать его же в исправленном виде.

4. Напишите функцию, которая будет принимать три числа: ширину и высоту прямоугольника и радиус круга и возвращать True, если прямоугольник может поместиться в круг.

5. Напишите функцию, которая будет принимать строку и две буквы. Функция должна проверять, всегда ли после первой переданной буквы идет вторая.

6. Создайте регулярное выражение для проверки того, является ли строка валидным числом с плавающей точкой. Обратите внимание: написать нужно паттерн, а не функцию.

7. Сиракузская последовательность генерируется следующим образом. Берем любое натуральное число  $n$ . Если оно четное, то делим его на 2, а если нечетное, то умножаем на 3 и прибавляем 1 (получаем  $3n + 1$ ).

8. Напишите функцию, которая будет принимать список чисел и проверять, является ли каждое число больше суммы всех предыдущих чисел.

9. Напишите функцию, которая будет принимать словарь и строку — кличку собаки (в нижнем регистре). Ключами словаря будут названия пропавших вещей (и кличка собаки, если она пропала), а значениями — их стоимость.

10. Вы занимаетесь сборкой игрушечных машинок. Каждая машинка должна иметь четыре колеса, один корпус и две фигурки человечков внутри.