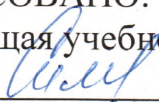


Министерство образования Края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

СОГЛАСОВАНО:

Заведующая учебной частью

 Д.С. Симашова

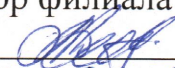
протокол № 13

от « 02 » 09 2026 г.



УТВЕРЖДАЮ:

Директор филиала

 Е. С. Тадер

Приказ № 02-02-94

от « 02 » 09 2026 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2026– 2027 уч. год**
технической направленности

«Python-3 для начинающих»

Форма реализации программы – очная

Срок реализации – 1 год

Возраст обучающихся – 12-17 лет

Составитель программы:

Захаров Илья Сергеевич

педагог дополнительного образования

г. Норильск, 2026 г.

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
1.1. НОВИЗНА ПРОГРАММЫ ДООП.....	4
1.2. АКТУАЛЬНОСТЬ ПРОГРАММЫ ДООП.....	4
1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ:.....	4
1.4. ЦЕЛЬ ДООП.....	5
1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП.....	5
1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП.....	5
1.7. СРОК РЕАЛИЗАЦИИ ПРОГРАММЫ:	6
1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ.....	6
1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ.....	7
2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.....	9
3. СОДЕРЖАНИЕ ДООП	11
4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО- МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	15
5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....	16
6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП.....	16
7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	16

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Python 3 для начинающих» 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 – стартовый. Освоение программного материала данного уровня предполагает получение учащимися первоначальных знаний в области программирования на языке Python.

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 техникой, 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 достойно и в необходимом объеме. Следствием этого является лишь формальное восприятие учащимися основ программирования и неумение применять полученные знания на практике.

Изучение программирования невозможно без регулярного 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. 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.

1.1. НОВИЗНА ДООП

Новизна данной программы заключается в практической направленности и широком использовании в обучении современных интернет-технологий. Обучающиеся не просто слушают материал, но и реализуют проекты начиная со второго занятия.

В современности организация учебного процесса при помощи онлайн технологий приобретает особое значение. Программа предполагает систему дополнительного взаимодействия педагога и учеников через getcourse-среду, которая позволяет встраивать решения кода онлайн в рабочие каналы сервера. Реализация программы с использованием платформы getcourse позволяет сделать обучение интерактивным, дать постоянный доступ к учебным материалам вне учебного времени. Такая система организации дистанционной работы очень востребована в условиях большого количества активированных по погодным условиям дней.

1.2. АКТУАЛЬНОСТЬ ДООП

Актуальность программы обусловлена необходимостью воспитания инженерно-технических кадров в сфере IT технологий. ДООП соотносится с тенденциями развития дополнительного образования и, согласно Концепции развития дополнительного образования, способствует формированию и развитию у обучающихся имеющихся навыков операционного и логического стиля мышления, исследовательских и творческих качеств. Программа отвечает современным потребностям детей и родителей в получении необходимых теоретических знаний и навыков в сфере программирования для личностного развития и успешной адаптации к требованиям жизни в современном информационном обществе.

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.

Программа предполагает участие обучающихся в интеллектуальных соревнованиях по программированию различных уровней, что даст возможность детям полностью реализовать свой интеллектуальный и творческий потенциал.

1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Педагогическая целесообразность заключается в том, что данная программа позволит выявить заинтересованных учащихся, проявивших интерес к знаниям, оказать им помощь в формировании устойчивого интереса к программированию на языке Python. В процессе изучения Python учащиеся научатся программировать на языке будущего, это повысит уровень логического мышления, сформирует аналитический склад ума.

Программа «Python 3 для начинающих» является целостной и непрерывной в течение всего процесса обучения, и позволяет учащемуся шаг за шагом раскрывать в себе творческие возможности и самореализоваться в современном мире.

Педагогическая целесообразность программы обусловлена ее профориентационной направленностью, т.к. полученные знания, умения и навыки помогут каждому обучающемуся в их дальнейшей жизни, а также сформируют навыки самостоятельного изучения других программных продуктов.

1.4. ЦЕЛЬ ДООП

Цель ДООП — сформировать у учащихся комплекс знаний в области алгоритмизации и программирования для реализации творческих проектов на языке Python.

Задачи:

- Формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- Знакомство с принципами, методами функционального и объектно-ориентированного программирования;
- Приобретение навыков работы в интегрированной среде разработки на языке Python;
- Изучение алгоритмических конструкций языка программирования Python;
- Знакомство с основными структурами данных и типовыми методами обработки этих структур;
- Знакомство с понятием проекта и алгоритмом его разработки;
- 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;
- Формирование правильного методологического подхода к познавательной и практической деятельности;
- Формирование мотивации учащихся к изобретательству, созданию собственных программных реализаций.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Python 3 для начинающих» ориентирована на обучающихся от 10 до 13 лет. Занятия проводятся в группе из 12 человек.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на Программу осуществляется в соответствии с Положением о наборе Филиала АНО ДТ «Красноярский «Кванториум» в г. Норильске «Центр цифрового образования детей IT-Куб г. Норильск».

Зачисление на обучение происходит исходя из результатов очного тестирования и собеседования.

1.7. СРОК РЕАЛИЗАЦИИ ПРОГРАММЫ

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

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.

Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств. Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить.

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 течении части занятия.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы. Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

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

Учащиеся будут уметь:

- Решать простые, сложные и нестандартные задачи;
- Распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;
- Организовывать данные для эффективной алгоритмической обработки;
- Использовать логические значения, операции и выражения с ними;
- Выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов, простых и табличных величин;
- Реализовывать полученные навыки для решения задач в олимпиадах по программированию различного уровня;
- Воспринимать математические, естественно-научные и профессиональные знания, умение самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;
- Реализовать полученные знания с помощью проектной работы, направленной на решение прикладных задач.

Учащиеся будут знать:

- Основные подходы к созданию программ на высокоуровневом языке программирования Python;
- Синтаксис, основные алгоритмические конструкции и парадигмы программирования на Python;
- Основные элементы программирования: ввод/вывод информации, данные, операции с данными, условное выполнение, циклы, подпрограммы.
- Основные библиотеки (random, tkinter и др.).

Предметные результаты:

- Сформированы и развиты навыки алгоритмического и логического мышления, грамотной разработки программ;
- Изучены принципы, методы функционального и объективно-ориентированного программирования;
- Приобретены навыки работы в интегрированной среде разработки на языке Python;
- Изучены конструкции языка программирования Python;
- Рассмотрены основные структуры данных и типовые методы обработки этих структур.

Метапредметные результаты:

- Рассмотрено понятие проекта и алгоритм его разработки;
- Сформирована самостоятельность и творческий подход к решению задач с использованием средств вычислительной техники;
- Приобретены навыки поиска информации в сети Интернет, и использования информации при решении задач.

Личностные результаты:

- Сформирована коммуникативная компетентность в общении и сотрудничестве со сверстниками и взрослыми в процессе образовательной, учебно-исследовательской, творческой и других видах деятельности.

Механизм оценки результативности

Контроль качества образования осуществляется в форме тестов, выполнения практических и проектных работ.

В конце каждого раздела обучающиеся выполняют задания по написанию программ на различные темы, связанные с пройденным материалом. При полном прохождении курса оценка знаний будет проводиться по итогам выполнения обучающимися индивидуальных или групповых проектов по темам, которые выбирают сами обучающиеся исходя из своих предпочтений.

Критериями оценки теоретических знаний являются: степень усвоения теоретического материала, глубина, широта и системность теоретических знаний, грамотное использование компьютерных терминов.

Критериями уровня овладения практическими умениями и навыками являются: разнообразие умений и навыков, грамотность (соответствие существующим нормативам и правилам, технологиям) практических действий, свобода владения специальным компьютерным оборудованием и программным обеспечением, качество творческих проектов, учащихся — грамотность исполнения, использование творческих элементов.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
Раздел 1. Введение в программирование (60 часов)					
1.1.	Введение. Устройство языка Python.	2	1	1	Тест
1.2.	Данные. Обработка данных	2	1	1	Практическая работа
1.3.	Построение командной работы.	4	2	2	Практическая работа
1.4.	Условный оператор	12	4	8	Практическая работа
1.5.	Циклы	8	3	5	Практическая работа
1.6.	Команды циклов останова и прерывания циклов.	8	3	5	Практическая работа
1.7.	Циклы с подсчетом «range».	12	4	8	Практическая работа
1.8.	Вложенные циклы.	8	3	5	Практическая работа
1.9.	Групповая работа. Решение задач.	4	0	4	Практическая работа
Раздел 2. Базовые конструкции Python (34 часа)					
2.1.	Знакомство с коллекциями.	4	1	3	
2.2.	Знакомство со строками.	4	2	2	Практическая работа
2.3.	Срезы в строках.	6	2	4	Практическая работа
2.4.	Методы строк.	8	2	6	Практическая работа
2.5.	Знакомство со списками.	6	2	4	Практическая работа
2.6.	Групповая практическая работа.	6	0	4	Практическая работа
Раздел 3. Знакомство с функциями (16 часов)					
3.1.	Функции.	4	1	3	Практическая работа
3.2.	Области видимости.	2	1	1	Практическая работа

3.3.	Передача параметров в функции.	2	1	1	Практическая работа
3.4.	Расширенные возможности функций.	8	3	5	Практическая работа
Раздел 4. Работа с Модулями (34 часа)					
4.1.	Модули.	2	1	1	
4.2.	Знакомство с модулями Python. Часть №1 (random)	8	3	5	Практическая работа
4.3.	Знакомство с модулями Python. Часть № 2 графика. Tkinter. PyQt5	6	1	5	Практическая работа
4.4.	Групповая практическая работа. Создаем игру «Сапёр» на PyQt5	4	1	3	Практическая работа
4.5.	Групповая практическая работа.	6	0	6	Практическая работа
4.6.	Разработка проекта.	8	0	8	Защита проекта
Итого		144	42	102	

3. СОДЕРЖАНИЕ ДООП

Раздел 1. Введение в программирование (60 часов)

Тема 1.1. Введение. Устройство языка Python (2 часа).

Теория: Вводная лекция, знакомство с учениками. Проведение инструктажа по технике безопасности. Знакомство с историей языков программирования. Иерархическое устройство языка Python. Знакомство с соглашением, о том, как писать код для языка Python per8.

Практическая работа: Знакомство со средой разработки (IDLE) Pycharm. Установка и первый запуск. Использование команд input и print.

Тема 1.2. Данные. Обработка данных (2 часа).

Теория: Знакомство с основными типами данных. Правила объявления переменных. Знакомство с Арифметическими действиями с переменными. Знакомство с различными способами ввода данных.

Практическая работа: Использование умножения, деления, сложения, вычитания с переменными одного типа, переменными разных типов.

Тема 1.3. Построение командной работы (4 часа).

Теория: Знакомство с групповой работой. Разделение задач на подзадачи. Умение ставить цели и задачи для группы.

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

Тема 1.4. Условный оператор (12 часов).

Теория: 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 1.5. Циклы (8 часов).

Теория: Знакомство с понятием цикла, общий вид цикла.

Практическая работа: Практическое использование циклов While и for. Практическое использование отладчика в Pycharm

Тема 1.6. Команды циклов остановок и прерывания циклов (8 часов).

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

Практическая работа: Апробация полученных знаний на практике при решении задач.

Тема 1.7. Циклы с подсчетом «range». (12 часов).

Теория: 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 1.8. Вложенные циклы. (8 часов).

Теория: Знакомство с вложенными циклами.

Практическая работа: Апробация полученных знаний на практике при решении задач.

Тема 1.9. Групповая работа. Решение задач (4 часа).

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

Раздел 2. Базовые конструкции в Python (34 часа).

Тема 2.1 Знакомство с коллекциями (4 часа).

Теория: Коллекция – «контейнер», содержащий различные элементы.

Тема 2.2 Знакомство со строками (4 часа).

Теория: 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 (6 часов).

Теория: Знакомство со срезами, аргументы среза, отрицательные параметры.

Практическая работа: решение задач на извлечение среза из строки.

Тема 2.4 Методы строк (8 часов).

Теория: 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 (6 часов).

Теория: 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. Групповая практическая работа (6 часов).

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

Раздел 3. Знакомство с функциями (16 часов).

Тема 3.1. Функции (4 часа).

Теория: 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 (2 часа).

Теория: 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 (2 часа).

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

Практическая работа: Работа рекурсивных функций, использование lambda функции.

Тема 3.4. Расширенные возможности функций (8 часов).

Теория: Знакомство с дополнительными возможностями функции.

Практическая работа: Решение задач по теме функции.

Раздел 4. Работа с модулями (34 часа).

Тема 4.1. Модули (2 часа).

Теория: Способы установки модулей. Импорт функций из модулей. Импорт всех функций из модуля, перезагрузка модуля.

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

Тема 4.2. Знакомство с модулями Python. Часть №1 (random) (8 часов).

Теория: 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 (6 часов).

Теория: 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 «Сапёр».

Тема 4.5. Групповая практическая работа (6 часов).

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

Тема 4.6. Разработка проекта (8 часов).

Практическая работа: Работа над проектом. Защита проекта.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

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

Программное обеспечение:

1. Операционная система Windows 10
2. Пакет программ MS OFFICE
3. PyCharm
4. Wing
5. Gilly

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. К. Ю. Поляков, Е. А. Еремин. Информатика. Углубленный уровень. Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2014
2. М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2011
3. Задачи по программированию. Под ред. С. М. Окулова. М.: БИНОМ, Лаборатория знаний, 2006
4. С. М. Окулов. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2012
5. Бриггс, Джейсон Б87 Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М.: Манн, Иванов и Фербер, 2017

Дополнительная литература:

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

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

Захаров Илья Сергеевич, педагог дополнительного образования.

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

В конце каждого раздела обучающиеся выполняют задания по написанию программ на различные темы, связанные с пройденным материалом. 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.

Предметные результаты оцениваются следующим образом:
5 баллов (высокий уровень) – 91-100% выполнения заданий
4 балла (повышенный уровень) - 71-90% выполнения заданий
3 балла (базовый уровень) – 50-70% выполнения заданий

ВАРИАНТ 1.1

- 1) Укажите значение переменной s после выполнения следующего кода:

```
s = 0
k = 30
d = k - 5
k = 2 * d
s = k - 100
```

Введите численный ответ _____

- 2) Укажите значение переменной x после выполнения следующего кода:

```
x = 3
y = 4
z = x + y
z = z + 1
x = y
y = 6
x = z + y + 3
```

- 3) Введите численный ответ _____

Что будет выведено на экран в результате выполнения следующей программы?

```
num1 = 22
num2 = 75
if num1 // 9 == 0 or num2 % 9 == 0:
    print('число', num1, 'выиграло')
else:
    print('число', num2, 'выиграло')
```

Введите численный ответ _____

- 4) Как правильно ввести текст 123**456**789

- 5) Какую функцию расчета квадратного корня _____

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

- 1) Напишите программу, которая считывает три целых числа и выводит на экран их произведение. Каждое число записано в отдельной строке.

- 2) Напишите программу, вычисляющую объём куба и площадь его полной поверхности, по введённому значению длины ребра.
- 3) Напишите программу для нахождения периметра треугольника.
- 4) Напишите программу для расчета среднего арифметического двух чисел.
- 5) Напишите программу, которая считывает целое число и определяет является ли оно двузначным нечетным числом.

ВАРИАНТ 1.2

- 1) Укажите значение переменной y после выполнения следующего кода:

```
y = 0
k = 20
d = k - 5
k = 2 * d
y = k - 80
```

Введите численный ответ _____

- 2) Какое значение будет выведено на экран после выполнения следующей программы, если с клавиатуры введено число 7?

```
a = int(input())
if a >= 2 and a <= 17:
    b = 3
    p = a * a + b * b
else:
    b = 5
p = (a + b) * (a + b)
print(p)
```

Введите численный ответ _____

- 3) Укажите значение переменной x после выполнения следующего кода:

```
x = 3
y = 4
z = x + y
z = z + 1
x = y
y = 5
x = z + y + 7
```

Введите численный ответ _____

- 4) Как правильно ввести текст 147***258***369

- 5) Как правильно вызвать функцию числа π _____

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

- 1) Напишите программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе.
- 2) Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом.
- 3) Напишите программу для нахождения площади треугольника.
- 4) Напишите программу для расчета среднего арифметического трех чисел.
- 5) Напишите программу, которая считывает целое число и определяет является ли оно трехзначным четным числом.

ВАРИАНТ 2.1

- 1) На вход программе подается строка. Напишите программу, которая меняет регистр символов, другими словами замените все строчные символы заглавными и наоборот.
- 2) На вход программе подается строка текста и строка-разделитель. Напишите программу, которая вставляет указанный разделитель между каждым символом введенной строки текста.
- 3) Напишите функцию `print_fio(name, surname, patronymic)`, которая принимает три параметра:
 - `name` – имя человека;
 - `surname` – фамилия человека;
 - `patronymic` – отчество человека;

а затем выводит на печать ФИО человека.

- 4) Напишите функцию `get_days(month)`, которая принимает в качестве аргумента номер месяца и возвращает количество дней в данном месяце.
- 5) Тимур загадал число от 11 до  n. За какое наименьшее количество вопросов (на которые Тимур отвечает "больше" или "меньше") Руслан может гарантированно угадать число Тимура?

ВАРИАНТ 2.2

- 1) Дополните приведенный код, используя форматирование строк с помощью метода `format`, так чтобы он вывел текст: «In 2010, someone paid 10k Bitcoin for two pizzas.» (без кавычек).

- 2) На вход программе подается строка текста, содержащая целые числа. Напишите программу, которая по заданным числам строит столбчатую диаграмму.
- 3) Напишите функцию `print_digit_sum()`, которая принимает одно целое число `num` и выводит на печать сумму его цифр.
- 4) Напишите функцию `number_of_factors(num)`, принимающую в качестве аргумента число и возвращающую количество делителей данного числа.
- 5) На вход программе подается строка текста на английском языке, в которой нужно зашифровать все слова. Каждое слово строки следует зашифровать с помощью шифра Цезаря (циклического сдвига на длину этого слова). Строчные буквы при этом остаются строчными, а прописные – прописными. Гарантируется, что между различными словами присутствует один пробел.

ИТОГОВАЯ АТТЕСТАЦИЯ ЗА ГОД

Курс заканчивается выполнением зачетного проекта с использованием графического модуля, как итоговый проект.

С использованием модуля Tkinter необходимо создать игру.

Итоговый проект – 25 баллов

От 20 баллов зачет