#### Министерство образования и науки Самарской области

Государственное бюджетное образовательное учреждение дополнительного образования Самарской области «Самарский областной центр детско-юношеского технического творчества»

Принята на заседании Методического Совета Протокол № 3

от «<u>20</u>» шоще 2022 г.

Принята на заседании кафедры факультета среднего и профессионального образования  $\Phi\Gamma$ АОУ ВО СГЭУ Протокол № 4 от « 05 » сифем 2022 г.

УТВЕРЖДАЮ
И.о. директора ГБОУ ДО СО СОЦДЮТТ

/А.С. Сафронов/
2022 г.

СОЦДЮТТ

Дополнительная общеобразовательная общеразвивающая программа технической направленности

# Программирование на Python (начальный уровень)

Возраст детей: 12-18 лет Срок обучения: 1 год

#### Разработчик:

Яковлев Дмитрий Николаевич педагог дополнительного образования 1 квалификационной категории

# Оглавление

Оглавление	2
Пояснительная записка	3
Учебно-тематический план	6
МОДУЛЬ 1. «Алгоритмы и структуры данных на языке Python»	6
МОДУЛЬ 2. «Полезные библиотеки»	9
МОДУЛЬ 3. «ООП»	10
МОДУЛЬ 4.1. «Разработка приложения»	11
МОДУЛЬ 4.2. «Базы данных и оконные приложения»	12
Контрольно-измерительный блок	15
Ресурсное обеспечение программы	16
Список используемой литературы	18
Приложение 1Ошибка! Закладка не	определена.
Промежуточный мониторинг по программе «Python»	200
Приложение 2	277
Календарно-тематический план программы «Python»	277
Приложение 3	29
Бланк наблюдения за динамикой личностного развития обучающихся	29

#### Краткая аннотация

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Python» (далее - программа) включает в себя 4 тематических модуля. Программа содержит реализацию проектной деятельности и направлена на овладение базовыми знаниями языка программирования Python. Обучаясь по данной программе, учащиеся смогут проявить интерес к инженерным профессиям, связанных с языками программирования

#### Пояснительная записка

Направленность программы – техническая.

Актуальность программы заключается в актуальности и широких возможностях языка программирования Python. Python - очень популярный и востребованный на рынке язык программирования. Например, в известном рейтинге ТІОВЕ он занимает третье место, при этом его востребованность в последнее время растет. Python – это язык будущего. Также программа соответствует нормативно-правовым документам регионального и федерального уровней: Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ», «Стратегия развития воспитания в РФ на период до 2025 года» (утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р), Приказ Министерства просвещения России от 9.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», Приказ Министерства просвещения РФ от 3.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей», Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», Методические рекомендации по проектированию дополнительных общеразвивающих программ, направленных письмом министерства образования и науки Самарской области России от 18.11.2015 № 09-3242, Методические рекомендации по разработке дополнительных общеобразовательных программ (Приложение к письму министерства образования и науки Самарской области 03.09.2015 № МО-16-09-01/826-ТУ).

Технологии, изучаемые в данной программе, наилучшим образом позволяют быстро познакомить учащихся с основами программирования и работы с данными.

- Python это универсальный язык программирования. Он пригоден для создания самых разных программ, от игр до веб-сайтов;
  - Python современный, простой и удобный язык;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро.

**Новизна программы** заключается в том, что она разработана с учётом современных тенденций по принципу модульного освоения материала и максимально отвечает запросу общества и развитию кадрового потенциала региона. Также новизна программы в том, что она использует сетевую форму деятельности: часть модулей обучающиеся изучают в центре цифрового образования детей «IT-cube», один модуль обучающиеся проходят непосредственно в своем колледже.

**Педагогическая целесообразность** программы заключается в том, что обучающиеся приобретут навыки, которые позволят им создавать работающие приложения и станут основой для дальнейшего изучения программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. Гибкая структура программы позволит для каждого учащегося сформировать собственный, наиболее для него подходящий путь вхождения в профессию.

Отпичительная особенность программы состоит в том, что она в краткой форме знакомит обучающихся с основами языка программирования Python и его применения при работе с базами данных. Программа реализуется в сетевой форме: 72 часа (модули 1, 2, 3, 4.2) реализуются на базе центра цифрового образования детей «IT-cube» (структурное подразделение ГБОУ ДО СО СОЦДЮТТ), 26 часов (модуль 4.1) реализуются на базе ФГАОУ ВО «Самарский государственный экономический университет» (кафедра факультета среднего и профессионального образования). В каждом модуле программы присутствует как теоретическая, так и практическая часть, поэтому обучающиеся будут видеть конкретны результат освоения предмета.

**Цель программы** — изучение навыков программирования на языке Python и последующее использование полученных знаний и методов в учебной и профессиональной деятельности.

#### Задачи:

Обучающие:

- сформировать у обучающихся представление об основных элементах программирования;
- познакомить с синтаксисом языка программирования Python;
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.
- изучить работу с данными и язык SQL.

# Развивающие:

- совершенствовать аналитические навыки;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени;

# Воспитательные:

- воспитывать в учащихся усидчивость, аккуратность, умение доводить начатое дело до конпа;
- формировать коммуникативные навыки.

Программа предусматривает базовый уровень освоения содержания программы, предполагающий использование общедоступных универсальных форм организации материала, среднюю сложность задач, поставленных перед обучающимися.

**Адресат программы** – дети в возрасте 14-18 лет. Наполняемость групп: 8-12 человек.

Срок освоения программы – 1 год обучения.

**Режим занятий**: 98 академических часа в год, 2 академических часа в неделю. Программой предусмотрена возможность обучения по индивидуальному образовательному маршруту.

В каникулярное время занятия проводятся в соответствии с календарным учебным графиком, допускается изменение форм занятий, проведение воспитательных мероприятий.

# Ожидаемы результаты

#### Личностные результаты

- формирование навыка доводить дело до конца,
- формирование мировоззрения, необходимого в современном цифровом мире,
- формирования позитивного отношения к коллегам, их мнению,
- формирование ответственного отношения к обучению.

# Метапредметные результаты

- Умение создавать модели для решения практических задач.
- Умение разрешать конфликты на основе выработки общей позиции
- Умение критически оценивать сроки реализации задуманного проекта.
- Умение вносить изменение в проект, корректировать изначальный план
- Умение ставить для себя новые задачи

# Предметные результаты

Модульный принцип построения программы предполагает описание предметных результатов в каждом конкретном модуле.

# Методы организации учебного процесса

При организации занятий для достижения поставленных целей и решения задач используются различные формы проведения.

- индивидуальные занятия,
- коллективные занятия,
- мозговой штурм,
- проектная деятельность,
- подготовка к участию в конкурсах.

#### Оценка и критерии оценки освоения программы

# Методы контроля

- тестирование,
- контрольные работы,
- устный опрос,
- наблюдение за деятельностью ученика,
- проектная деятельность.

Данные методы используются при анализе деятельности обучающихся, при организации текущей, промежуточной и итоговой аттестации обучающихся.

#### Учебно-тематический план

№ п/п	Наименование	Количество часов		
V (2 11/ 11	модуля	Всего	Теория	Практика
1.	«Алгоритмы и структуры данных на языке Python»	30	10	20
2.	«Полезные библиотеки»	12	2	10
3.	«ООП»	12	4	8
4.1	«Разработка приложений»	18	3	15
4.2	«Базы данных и оконные приложения»	26	6	20
	ИТОГО	98	25	73

# МОДУЛЬ 1. «Алгоритмы и структуры данных на языке Python»

Реализация этого модуля направлена на обучение первоначальным навыкам программирования на языке Python.

**Цель модуля:** заложить базис для дальнейшего изучения программирования и создания приложений.

# Задачи модуля:

- изучить основы языка Python;
- познакомить с оболочками для разработки программ;
- изучить основные управляющие конструкции языка;
- рассказать про коллекции
- познакомить с понятием функции

# Ожидаемы результаты

# Предметные результаты

- умение писать грамотный, красивый код;
- умение анализировать как свой, так и чужой код;
- умение импортировать модули в программу.

# Учебно-тематический план модуля 1

№	Тема, содержание	Общее кол- во часов	Теория	Практика
1	Введение в Python. Установка и знакомство со средой. Первая программа. Понятие консольного приложения	2	1	1
2	Ввод и вывод данных	2		2
3	Переменные. Типы данных	2	1	1
4	Условия if, else. Операторы сравнения	2	1	1
5	Вычисления. Библиотека math.	2		2
6	Цикл while	2	1	1
7	Цикл for	2	1	1
8	Строки. Методы для работы со строками. Срезы	2	1	1
9	Списки и Кортежи	2		2
10	Множества	2	1	1
11	Словари	2		2
12	Двумерные массивы	2		2
13	Функции. Передача параметров. Рекурсия	2	1	1
14	Области видимости переменных.	2	1	1
15	Лямбда-функции и функции высшего порядка	2	1	1
Итог	Γ <b>0</b>	30	10	20

# Содержание программы модуля 1

Тема 1. Введение в Python. Установка и знакомство со средой. Первая программа. Понятие консольного приложения.

<u>Теория:</u> устройство языка Python. Среда разработки IDLE. Сохранение Python-программ Техника безопасности на занятии. Понятие «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

Практика: установка среды программирования.

# Тема 2. Ввод и вывод данных

<u>Практика:</u> вывод данных на языке Python. Ввод и отладка программ в среде. Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print(), практическая работа на ввод и вывод данных.

Тема 3. Переменные. Типы данных

<u>Теория:</u> понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input()

Практика: практическая работа «Разные переменные».

Тема 4. Условия if, else. Операторы сравнения

<u>Теория:</u> условный оператор. Переменные и арифметика. Погружение в условия. Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

Практика: решение задач.

# Тема 5. Вычисления. Библиотека math

<u>Практика:</u> словари и множества. Подключение и использование модулей стандартной библиотеки Python. Функции модуля Collections. Решение задач.

# Тема 6. Цикл while

<u>Теория:</u> понятие «цикл с предусловием». Конструкция while и её синтаксис. Зацикливание и выход из цикла с помощью команды break.

<u>Практика:</u> решение задач на циклические алгоритмы. Виды циклов и их конструкции. Решение задач на применение циклов while. Проект «Бомба взорвалась!». Написание программы по определенным условиям.

# Тема 7. Цикл for

<u>Теория:</u> понятие «цикл», «цикл со счётчиком». Конструкция for и её синтаксис.

<u>Практика:</u> проект «Таблица умножения»: создание приложения по определенным условиям.

# Тема 8. Строки. Методы для работы со строками. Срезы

<u>Теория:</u> понятие «строка». Создание строк. Переменные внутри строк. Операции со строками. Понятие «список». Создание списков. Добавление/удаление элементов в/из списка. Операции со списками.

Практика: решение практических задач.

# Тема 9. Списки и Кортежи

<u>Практика:</u> практическая работа – решение задач. Создание кортежа. Операции с кортежем. Создание словаря.

# Тема 10. Множества

<u>Теория</u>: работа со списками, строками, множествами и кортежами в Python. Понятие итератора, подпрограммы, процедуры, функции. Функции в Python.

Практика: решение практических задач.

# Тема 11. Словари

Практика: решение практических задач с использованием словарей и множеств.

#### Тема 12. Двумерные массивы

Практика: решение практических задач с применением массивов.

# Тема 13. Функции. Передача параметров. Рекурсия

<u>Теория:</u> понятие «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело.

<u>Практика:</u> Создание и вызов функции. Решение задач на отработку понятия «функция», её строение и синтаксис.

# Тема 14. Области видимости переменных

Теория: переменные и область видимости.

<u>Практика:</u> решение практических задач с применением разных областей видимости переменных.

# Тема 15. Лямбда-функции и функции высшего порядка

<u>Теория:</u> функции, функциональная парадигма программирования. Понятие лямбда функции и области ее применения. Функции модуля functools.

Практика: решение практических задач.

# МОДУЛЬ 2. «Полезные библиотеки»

Реализация этого модуля направлена на изучение возможностей экосистемы языка Python.

**Цель модуля:** научить решать практические задачи на языке программирования Python.

# Задачи модуля:

- изучить модули, входящие в стандартную библиотеку Python;
- познакомить с подключением внешних библиотек и утилитой рір;
- научить работать с документацией и искать нужную информацию в Интернете;
- решение прикладных задач.

#### Ожидаемы результаты

# Предметные результаты

- умение разбивать решение задачи на подзадачи, составлять алгоритм
- умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умение искать и обрабатывать ошибки в коде;
- умение использовать библиотеки Python по их назначению

#### Учебно-тематический план модуля 2

Nº	Тема, содержание	Общее кол- во часов	Теория	Практика
1	Работа с файловой системой и файлами	2		2
2	Работа с датой и временем	2		2
3	Утилита рір: установка дополнительных библиотек	4	1	3
4	Обработка исключений	4	1	3
Ито	ΓΟ	12	2	10

# Содержание программы модуля 2

Тема 1. Работа с файловой системой и файлами

<u>Практика:</u> понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python. Запись в файл.

Тема 2. Работа с датой и временем

<u>Практика:</u> практические задание на использование даты и времени в прикладных задачах.

Тема 3. Утилита рір: установка дополнительных библиотек

Теория: понятие «утилита pip», применение, назначение, установка.

Практика: практическая работа на применение утилиты рір.

# Тема 4. Обработка исключений

<u>Теория:</u> понятие исключения, синтаксис их обработки. Применение исключений при разработке и отладке программ.

Практика: решение практических задач.

# МОДУЛЬ 3. «ООП»

Реализация этого модуля направлена на знакомство с современными способами проектирования приложений.

**Цель модуля:** введение в объектно-ориентированное программирование (ООП). **Задачи модуля:** 

- изучить основы ООП;
- на практике освоить основные принципы ООП: инкапсуляцию, наследование, полиморфизм;
- проектирование своего приложения в парадигме ООП.

# Ожидаемы результаты

# Предметные результаты

- умение применять ООП в решении прикладных задач,
- умение использовать инкапсуляцию, наследование, полиморфизм,
- умение разрабатывать с нуля приложение с использованием ООП.

#### Учебно-тематический план модуля 3

№	Тема, содержание	Общее кол- во часов	Теория	Практика
1	Введение в ООП. Определение класса и создание экземпляра класса	2	1	1
2	Инкапсуляция	2	1	1
3	Наследование	4	1	3
4	Полиморфизм	4	1	3
Ито	Γ0	12	4	8

# Содержание программы модуля 3

Тема 1. Введение в ООП. Определение класса и создание экземпляра класса

<u>Теория:</u> Понятие «объектно-ориентированного программирования». Восходящий и нисходящий способы разработки. Методика объектно-ориентированного дизайна приложения. Шаблоны проектирования.

Практика: проектирование и реализация компьютерной игры

# Тема 2. Инкапсуляция

<u>Теория:</u> причины появления, принципы и основные сущности объектноориентированного подхода к разработке ПО. Инкапсуляция. Практика: решение практических задач.

# Тема 3. Наследование

<u>Теория:</u> причины появления, принципы и основные сущности объектноориентированного подхода к разработке ПО. Наследование. Характеристики наследников.

Практика: решение практических задач.

# Тема 4. Полиморфизм

<u>Теория:</u> причины появления, принципы и основные сущности объектноориентированного подхода к разработке ПО. Полиморфизм.

Практика: решение практических задач.

# МОДУЛЬ 4.1. «Разработка приложения»

Реализация этого модуля направлена на знакомство с обработкой данных в Я $\Pi$  Python.

**Цель модуля:** на примере создания оконного приложения познакомить с возможностями Python, связанными с обработкой данных.

# Задачи модуля:

- изучить основы SQL;
- изучить СУБД sqlite и работу с ней из Python;
- изучить основы PyQT;
- создать работающее приложение с базой данных.

# Ожидаемы результаты

# Предметные результаты

- знание основ SQL;
- знание взаимодействия СУБД sqlite и Python;
- применение PyQT для разработки приложений для решения прикладных задач.

# Учебно-тематический план модуля 4.1

№	Тема, содержание	Общее кол- во часов	Теория	Практика
1	Интегрированная среда	1	1	2
	разработчика	4	1	3
2	Этапы разработки приложения	4	1	3
	Визуальное событийно-	2	1	1
3	управляемое программирование	2	1	1
4	Разработка оконного приложения	2		2
	Разработка пользовательских	2		2
5	приложений	2		2
	Основы программирования баз	2		2
6	данных	2		2

7	Разработка клиент-серверных приложений. Аттестация	2		2
Итого		18	3	15

# Содержание программы модуля 4.1

Тема 1. Интегрированная среда разработчика

<u>Теория:</u> понятие и реализация итераторов и генераторов. Предназначение, особенности устройства и работы, типовые сферы применения.

Практика: решение практических задач.

Тема 2. Этапы разработки приложения

Теория: изучение этапов разработки приложения. Тестирование и отладка.

Практика: решение практических задач с использованием отладки и тестировки.

Тема 3. Визуальное событийно-управляемое программирование

<u>Теория:</u> знакомство с понятием «событие», свойство и характеристики. Назначение и применение на практике.

Практика: решение практических задач.

Тема 4. Разработка оконного приложения

Практика: решение практических задач с использованием оконного приложения.

Тема 5. Разработка пользовательских приложений

<u>Практика:</u> знакомство с библиотекой tkinter, изучение основных команд, виджетов, упаковщиков и приемов работы. Создание приложения с графическим интерфейсом.

Тема 6. Основы программирования баз данных

<u>Практика:</u> изучение преимуществ использования баз данных, практические задания на использование баз данных в пользовательских приложениях.

Тема 7. Разработка клиент-серверных приложений. Аттестация

Практика: разработка клиент-серверного приложения.

#### МОДУЛЬ 4.2. «Базы данных и оконные приложения»

Реализация этого модуля направлена на практическую работу, используя обработку данных в языке программирования Python.

**Цель модуля:** на примере создания оконного приложения познакомить с возможностями Python, связанными с обработкой данных.

#### Задачи модуля:

- изучить основы SQL;
- изучить СУБД sqlite и работу с ней из Python;
- изучить основы PyQT;
- создать работающее приложение с базой данных.

# Ожидаемы результаты

# Предметные результаты

- знание основ SQL;
- знание взаимодействия СУБД sqlite и Python;

– применение PyQT для разработки приложений для решения прикладных задач.

# Учебно-тематический план модуля 4.2

№	Тема, содержание	Общее кол- во часов	Теория	Практика
1	Основы SQLite. Создание Базы данных.	2	1	1
2	Язык SQL. Запросы. SELECT	2	1	1
3	Запросы INSERT, DELETE, UPDATE	2	1	1
4	Вложенные запросы	2	1	1
5	Работа с несколькими таблицами. Виды связей.	2	1	1
6	Библиотеки в Python для работы с SQLite	2		2
7	Понятие курсора.	2		2
8	Библиотека РуQТ	2	1	1
9	QtDesigner	2		2
10	Возможности РуQТ при работе с таблицами	2		2
11	Диалоги	2		2
12	Сбора независимого приложения.	2		2
13	Итоговое занятие. Аттестация	2	<u>-</u>	2
Итог	Γ0	26	6	20

# Содержание программы модуля 4.2

Тема 1. Основы SQLite. Создание Базы данных

<u>Теория:</u> Основные понятия баз данных. Основы SQLite. Назначение и свойства баз данных. СУБД.

Практика: практические задачи на обращения к базам данных.

Тема 2. Язык SQL. Запросы. SELECT

Теория: SQL, синтаксис, запросы. Запрос SELECT.

Практика: практические задачи на работу выборки базы данных.

Тема 3. Запросы INSERT, DELETE, UPDATE

Теория: запросы INSERT, DELETE, UPDATE. Назначение и использование.

Практика: практические задачи на отработку запросов.

Тема 4. Вложенные запросы

<u>Теория:</u> понятие «вложенного запроса», применение на практике. Достоинства и недостатки.

Практика: решение практических задач.

Тема 5. Работа с несколькими таблицами. Виды связей

<u>Теория:</u> Основные понятия баз данных. Основы SQLite. Назначение и свойства баз данных. СУБД.

Практика: практические задачи на обращения к базам данных.

Тема 6. Библиотеки в Python для работы с SQLite

Практика: практические задачи на знание библиотек.

Тема 7. Понятие курсора

Практика: решение практических задач.

Тема 8. Библиотека PyQT

<u>Теория:</u> знакомство с библиотекой РуQТ.

Практика: практические задачи на знание библиотек.

Тема 9. QtDesigner

Практика: решение практических задач.

Тема 10. Возможности РуQТ при работе с таблицами

Практика: решение практических задач.

Тема 11. Диалоги

Практика: решение практических задач.

Тема 12. Сборка независимого приложения

Практика: решение практических задач.

Тема 13. Итоговое занятие. Аттестация

Практика: практическая работа, итоговая аттестация.

# Контрольно-измерительный блок

# Формы подведения итогов реализации программы

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: входную, промежуточную и итоговую аттестацию результатов обучения детей. Входная аттестация проходит в начале учебного года (сентябрь) с целью анализа знаний обучающихся на старте освоения программы. Промежуточная аттестация проводится в конце первого полугодия с целью сопоставления целей и результатов освоение программы на середину учебного года и возможностью варьировать наполняемость программы и инструментарий для достижения цели к концу учебного года. Итоговый контроль проводится в конце года с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

Форма	Описание	Критерии оценки
•	Входной мониторинг	
Тестирование и	Фронтальный опрос в	Приложение 1
фронтальный опрос	начале входного	_
	мониторинга позволяет	
	определить шкалу	
	критериев. Если ребята	
	знают основы	
	алгоритмизации и изучали	
	языки программирования	
	ранее, то оценка теста идет	
	по семибалльной системе,	
	если нет – то оценка идет	
	по пятибалльной системе.	
	Задания во входном	
	тестировании направлены	
	на знание логики и	
	математики. Основы языка	
	Python и его синтаксис в	
	них не рассматриваются.	
	Промежуточный мониторин	Γ
Тестирование	Тестирование на	Приложение 2
	определение результата	
	освоения программы.	
	Итоговый мониторинг	
Защита проекта	Оконное приложение на	• Структурная сложность
	Python с использованием	• Качество кода
	библиотеки PyQT.	• Использование БД
		• Оригинальность идеи
		• Работоспособность
		• Дизайн

#### Ресурсное обеспечение программы

#### Методическое обеспечение

Образовательный процесс осуществляется в очной форме. Но данная программа предполагает реализацию и в дистанционном формате (с использованием платформ ZOOM, Discord).

- В образовательном процессе используются следующие методы:
- 1) словесные (беседа, опрос, дискуссия и т.д.);
- 2) игровые;
- 3) метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
  - 4) метод проектов;
  - 5) наглядные:
    - демонстрация плакатов, схем, таблиц, диаграмм;
    - использование технических средств;
    - просмотр кино- и телепрограмм;
  - 6) практические:
    - практические задания;
    - анализ и решение проблемных ситуаций и т.д.

Методические пособия, разработанные педагогом с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности. Электронное приложение к учебникам К.Ю. Полякова Информатика и ИКТ. Набор цифровых образовательных ресурсов — дидактические материалы, интерактивные тесты, анимационные плакаты.

# Материально-техническое обеспечение программы

Требования к помещению:

- компьютерный класс, отвечающий требованиям СанПиН для учреждений дополнительного образования;
- кабинет с 12 рабочими местами для обучающихся, 1 рабочим местом для педагога;
- качественное освещение;
- столы, стулья (по росту и количеству детей);
- демонстрационная магнитная доска;
- технические средства обучения (ТСО) (мультимедийное устройство);
- ноутбуки (12 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО;
- проектор.

# Оборудование:

- 1. Компьютеры:
- не менее 2 ГБ оперативной памяти;
- процессор с тактовой частотой не менее 1,2 ГГц;
- диагональ мониторов не менее 12 дюймов;

- наличие колонок или наушников;
- свободные 50 ГБ на накопителях;
- Интернет не медленнее 1 Мбит / с.
- 2. Проекционное оборудование;
- 3. Маркерная доска;
- 4. Программное обеспечение:
- операционная система любая, желательно Windows;
- поддерживаемые браузеры для работы LMS:
- Yandex Browser 17+
- Chrome 60+
- Python 3.6.0 или выше;
- среда Wing IDE 101 версии 6 или выше для Python;
- среда PyCharm Community Edition;
- пакет PyQt4 (на Qt5);
- пакет библиотек с SciPy: numpy, scipy, matplotlib, ipython + ipythonnotebook, sympy, pandas;
- ΠΟ Anaconda.

#### Список используемой литературы

- 1. М. Лутц. Изучаем Python. Вильямс. 2019.
- 2. М. Лутц. Python, карманный справочник. Вильямс. 2019.
- 3. А. Бьюли. Изучаем SQL. Символ-плюс. 2017.
- 4. Н. Прохоренок, В. Дронов. Python3 и PyQT5, Разработка приложений. БХВ-Петербург. 2019.

# Ресурсы в сети Интернет

- 1. https://pythonworld.ru/
- 2. <a href="https://metanit.com/python/">https://metanit.com/python/</a>
- 3. https://metanit.com/sql/
- 4. <a href="https://stepik.org/course/67/">https://stepik.org/course/67/</a>

# Нормативно-правовые источники

- 1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ»,
- 2. «Стратегия развития воспитания в РФ на период до 2025 года» (утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р),
- 3. Приказ Министерства просвещения России от 9.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»,
- 4. Приказ Министерства просвещения РФ от 3.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»,
- 5. Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»,
- 6. Методические рекомендации по проектированию дополнительных общеразвивающих программ, направленных письмом министерства образования и науки Самарской области России от 18.11.2015 № 09-3242,
- 7. Методические рекомендации по разработке дополнительных общеобразовательных программ (Приложение к письму министерства образования и науки Самарской области 03.09.2015 № МО-16-09-01/826-ТУ).

# Список литературы для педагога

- 1. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
- 2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
- 3. Лутц М. Изучаем Python. СПб.: Символ-Плюс, 2019
- 4. Окулов С. М. Основы программирования. М.: Бином. Лаборатория знаний, 2012.
- 5. Поляков К. Ю., Еремин Е. А. Информатика. Углублённый уровень.
- 6. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.

7. Эльконин, Д. Б. Детская психология: учеб. пособие для студ. высш. учеб. заведений / Д. Б. Эльконин; ред.сост. Б. Д. Эльконин. – 4-е изд., стер. – М.: Издательский центр «Академия», 2007. – 384 с

# Список литературы для обучающегося

- 1. Рутhon для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. М.: Манн, Иванов и Фербер, 2017 320 с. К. Вордерман и др.
- 2. Программирование для детей: Иллюстрированное руководство по языкам Scratch и Python, 224 стр. Издательство: Манн, Иванов и Фербер, 2017 г.
- 3. Программирование для детей на языке Python, 96 стр. Издательство: АСТ, 2017 г. Б. Пэйн. Python для детей и родителей, 352 стр. Издательство: Эксмо, 2017 г.
- 4. П. Томашевский. Привет, Python! Моя первая книга по программированию, 256 стр. Издательство: Наука и Техника, 2018 г.

# Промежуточный мониторинг по программе «Python»

#### Тест 1

# Задача 1

На пароме A грузовых машин и B легковых. Каждая грузовая весит C килограмм, легковая D килограмм. Какой вес перевозит паром?

# Ввод

Целые числа A, B, C, D на отдельных строках.

# Вывод

Общий вес в килограммах.

# Пример 1

Ввод	Вывод
1	10
2	
4	
3	

# Пример 2

Ввод	Вывод
2	250
5	
100	
10	

# Задача 1\*

Написать программу для перевода градусов Фаренгейта в градусы Цельсия.

Формула 
$$C = (F - 32) \times 5/9$$

# Ввод

Одно число (необязательно целое) – температура в градусах Фаренгейта.

# Вывод

Температура в градусах Цельсия с точность до сотых или больше.

# Пример 1

Ввод	Вывод
50	10

# Пример 2

Ввод	Вывод
40	4.444444444444

#### Залача 2

Программа должна вводить строки с клавиатуры до тех пор, пока пользователь не введет две одинаковые строки подряд. После этого вывести количество введенных строк и завершить работу. Дублированную строку считать за одну.

#### Ввол

Две или более строки.

# Вывод

Число полученных строк

# Пример 1

Ввод	Вывод
hello	1
hello	

# Пример 2

Ввод	Вывод
Стол	6
Стул	
Парта	
Дорога	
Стол	
Небо	
Небо	

#### Залача 3

Два датчика замеряют некую меняющуюся величину. Величина считается надежно измеренной только если оба выдают одно значение, иначе измерения игнорируются. .

# Ввод

Две строки из целых чисел через пробел.

# Вывод

Строка надежно измеренных значений через пробел.

# Пример 1

	Ввод	Вывод
Ī	1 3 55 731	1 55 731
	1 2 55 731	

Пример 2

Ввод	Вывод
15 5 20 5	5
5 5 4 4	

# Задача 4

Дана строка из круглых скобок. Проверить, правильно ли они расставленны.

# Ввод

Строка из круглых скобок

# Вывод

Yes либо No

# Пример 1

B	вод	Вывод
0		Yes

Пример 2

Ввод	Вывод
)(	No

Пример 3

Ввод	Вывод
00	Yes

Пример 4

Ввод	Вывод
(())	Yes

Пример 5

Ввод	Вывод
())(	No

#### Задача № 1.

Требуется определить победителей среди участников спортивного соревнования, занявших первые три призовых места по наибольшему количеству очков. Количество участников соревнования не превышает 100 человек. Соревнование построено таким образом, что никакие два или несколько участников не могут набрать одинаковое количество очков. Входная информация представлена в виде массива положительных целочисленных очков участников. В выходной массив требуется занести индексы очков победителей соревнования из исходного массива в порядке возрастания номера их места (первое, второе, третье). Считать, что начальный элемент массива имеет индекс 0.

# Формат входного файла

Во входном файле Input.txt находится массив очков участников соревнования  $X(0 \le X_i \le 100)$ . Элементы массива заданы в одну строку и разделены 1 пробелом. Количество элементов массива от 10 до 100.

# Формат выходного файла

В выходной файл Output.txt вывести индексы призеров.

#### Пример

<u>Input.txt</u>	Output.txt
20 50 40 80 76 30 75 52 45 70	3 4 6

#### Залача № 2

База данных видеокамер, замеряющих скорости автомобилей содержит N записей о номерах проехавших автомобилей и их скорости в момент съемки. Известна таблица штрафов за превышение допустимой скорости:

Скорость	Штраф
от 0 до 70 км/ч	нет штрафа (0 рублей)
от 71 до 80 км/ч	100 рублей
от 81 до 100 км/ч	300 рублей
от 101 до 120 км/ч	1000 рублей
от 121 км/ч	2000 рублей

Необходимо для каждого зафиксированного камерой номера автомобиля определить суммарный штраф и вывести такой список в порядке убывания штрафов.

# Формат входного файла

В первой строке входного файла Input.txt находится целое число N ( $1 \le N \le 1000000$ ). Следующие N строк входного файла содержат описания проехавших автомобилей — сначала номер автомобиля — строка, состоящая из 8 маленьких латинских букв или цифр, потом через пробел одно целое число  $V_i$  ( $0 \le V_i \le 250$ ) — зафиксированная скорость в км/ч.

# Формат выходного файла

В выходной файл Output.txt для каждого зафиксированного на камеру автомобиля выведите сначала номер автомобиля, потом через пробел суммарный штраф. Этот список необходимо вывести в порядке убывания суммарного штрафа. В случае нескольких автомобилей с одинаковым суммарным штрафом, их можно вывести в любом порядке.

# Пример входных и выходных данных

Input.txt	Output.txt
5	a111aa78 3000
a111aa78 150	00000063 300
o000oo63 70	a234bv56 0
a234bv56 65	
a111aa78 110	
o000oo63 83	

#### Задача №3

В банкомате есть купюры различных номиналов. Программа определяет сколько и каких купюр необходимо выдать человеку, чтобы выданная сумма была не больше запрошенной, но при этом максимально близка к ней. Происходит это следующим образом. В первую очередь выдаются купюры самого большого номинала. Если эти купюры кончаются в банкомате, или если добавление еще одной купюры делает сумму больше запрашиваемой, то начинают выдаваться купюры номинала, второго по величине, и т. д. Так происходит до тех пор, пока выдаваемая сумма не будет в точности равна запрошенной, либо же пока не выяснится, что суммы не равны, но меньших купюр нет. Необходимо сообщить, какие суммы будут выданы по каждому запросу. При этом, естественно, нужно учитывать, что после каждого запроса число купюр в банкомате уменьшается.

# Формат входного файла

В первой строке входного файла input.txt дано число N ( $1 \le N \le 10$ ) — количество различных типов купюр. В следующих N строках входного файла описаны все возможные типы купюр. Каждый тип характеризуются двумя числами  $C_i$  и  $D_i$  ( $1 \le C_i \le 5000$ ,  $1 \le D_i \le 100$ ) — номинал купюры и количество купюр такого номинала. В следующей строке дано число M ( $1 \le M \le 100$ ) — количество поступивших запросов. В следующих M строках описаны запросы в порядке их поступления. Каждый запрос характеризуется одним числом  $X_i$  ( $1 \le X_i \le 100000$ ) — запрашиваемой суммой.

#### Формат выходного файла

В выходной файл output.txt требуется вывести результаты обработки всех запросов в порядке поступления, по одному на строке. Про каждый запрос выведите сумму, выданную пользователю после обработки запроса или 0, если деньги кончились.

#### Пример входных и выходных данных

Input.txt	Output.txt
3	220
50 3	15
10 20	20
5 1	
3	
223	
15	
27	

#### Залача № 4

Адрес сети Интернет, запрашиваемый пользователем, может состоять из четырех частей, из которых первые три являются доменными именами различного уровня и разделены точками, а последняя отделена от них двоеточием и является портом, к которому обращается пользователь. Например, при обращении пользователя к www.yandex.ru:8080 слово ги является доменным именем первого уровня, слово yandex — доменным именем второго уровня, слово www — доменным именем третьего уровня, а 8080 является портом. Вам необходимо по списку запросов пользователей перечислить все различные порты, к которым обращались пользователи.

#### Формат входного файла

В первой строке входного файла input.txt находится одно натуральное число N ( $1 \le N \le 100$ ) — количество адресов в данном вам списке. В следующих N строках содержатся сами адреса в разных допустимых форматах доменной нотации, описанной выше. Все доменные имена всех уровней состоят только из букв латинского алфавита, а порты — только из цифр. Длина любого данного вам адреса не превышает 250 символов.

#### Формат выходного файла

В первой строке выходного файла output.txt выведите одно целое число K — количество различных портов, к которым обращались пользователи. В следующих K строках выведите сами порты. Каждый порт должен быть выведен ровно один раз в отдельной строке. Порты могут быть выведены в любом порядке.

#### Пример входных и выходных данных

Input.txt	Output.txt
6	3
www.yandex.ru:3128	3128
yandex.ru:80	80
YA.ru:80	666
WWW.YA.RU	
www.yandex.ru	
www.ya.ru:666	

# Задача №5

Одной из задач, возникающей при разработке программного обеспечения для кнопочных мобильных телефонов, является поиск контактов в телефонной книжке. Дан список контактов, записанных строчными английскими буквами, и запрос в виде набора цифр. Каждая цифра запроса соответствует одной из букв в приведенной ниже таблице.

2	3	4	5	6	7	8	9
a b c	d e f	g h i	j k l	m n o	pqrs	t u v	wxyz

То есть, например, запросу "23" удовлетворяют следующие 9 строк: "ad", "ae", "af", "bd", "be", "bf", "cd", "ce" и "cf". Ваша задача: по списку контактов и запросу выдать все контакты, содержащие как подстроку хотя бы одну строку, удовлетворяющую запросу. Например, по запросу "72" могут быть выбраны контакты"рара" (содержит подстроку "pa") и "sberbank" (содержит подстроку "rb").

### Формат входного файла

В первой строке входного файла input.txt дано число N ( $1 \le N \le 200$ ) — количество контактов. В следующих N строках записаны контакты, каждый из которых является строкой, длина которой не превышает 100 символов. Каждый контакт содержит только строчные английские буквы. В последней строке входного файла дан запрос в виде строки, содержащей цифры от 2 до 9. Длина запроса не превышает 100 символов.

#### Формат выходного файла

В выходной файл output.txt требуется вывести список контактов, удовлетворяющих запросу, в порядке, данном во входном файле. Гарантируется, что хотя бы один такой контакт существует.

# Пример входных и выходных данных

Input.txt	Output.txt
3	papa
mama	sberbank
papa	
sberbank	
72	

Количество правильных ответов	21-27 баллов	16-20	14	Менее 13
Процент освоение программы	80-100%	60-79%	50-59%	Менее 50%

# Календарно-тематический план программы «Python»

Дата		Nº			
По плану	По факту	34 <u>ч</u> Занятия	Тема занятия	во часов	
	17.09.2021	1	Введение в Python. Установка и знакомство со средой. Первая программа. Понятие консольного приложения	2	
Сентябрь	24.09.2021	2	Ввод и вывод данных		
Октябрь	1.10.2021	3	Переменные. Типы данных	2	
Октябрь	8.10.2021	4	Условия if, else. Операторы сравнения	2	
Октябрь	15.10.2021	5	Вычисления. Библиотека math.	2	
Октябрь	22.10.2021	6	Цикл while	2	
Октябрь	29.10.2021	7	Цикл for	2	
Ноябрь	5.11.2021	8	Строки. Методы для работы со строками. Срезы		
Ноябрь	12.12.2021	9	Списки и Кортежи		
Ноябрь	19.12.2021	10	Множества		
Ноябрь	26.12.2021	11	Словари	2	
Декабрь	3.12.2021	12	Двумерные массивы	2	
Декабрь	10.12.2021	13	Функции. Передача параметров. Рекурсия	2	
Декабрь	17.12.2021	14	Области видимости переменных.	2	
Декабрь	24.12.2021	15	Лямбда-функции и функции высшего порядка	2	
Январь	14.01.2022	16	Работа с файловой системой и файлами	2	
Январь	21.01.2022	17	Работа с датой и временем		
Январь	28.01.2022	18	Утилита рір: установка дополнительных библиотек		
Февраль	4.02.2022	19	Утилита рір: установка дополнительных библиотек		
Февраль	11.02.2022	20	Обработка исключений		

Февраль	18.02.2022	21	Обработка исключений	2
Февраль	25.02.2022	22	Введение в ООП. Определение класса и создание экземпляра класса	
Март	4.03.2022	23	Инкапсуляция	2
Март	11.03.2022	24	Наследование	2
Март	18.03.2022	25	Наследование	2
Март	25.03.2022	26	Полиморфизм	2
Апрель	1.04.2022	27	Полиморфизм	2
Апрель	8.04.2022	28	Интегрированная среда разработчика	2
Апрель	15.04.2022	29	Интегрированная среда разработчика	2
Апрель	22.04.2022	30	Этапы разработки приложения	
Апрель	29.04.2022	31	Этапы разработки приложения	2
 Май	6.05.2022	32	Визуальное событийно-управляемое программирование	2
Май	13.05.2022	33	Разработка оконного приложения	2
Май	20.05.2022	34	Разработка пользовательских приложений	2
Май	27.05.2022	35	Основы программирования баз данных	2
Июнь	3.06.2022	36	Разработка клиент-серверных приложений. Аттестация	2
	1	0	бщее количество часов	72

# Бланк наблюдения за динамикой личностного развития обучающихся

Направление / Группа\_\_\_\_\_\_

	T					ПОКАЗА	трпи	
					ПОКАЗАТЕЛИ			
<b>№</b> π/π	ФИО	Во время занятий проявляет устойчивый интерес и инициативу при освоении программы	Использует в общении базовую систему понятий	Активно сотрудничает со сверстниками, уважительно относится к мнению окружающих	Проявляет интерес к проектной деятельности, активно включается в групповую работу	Аккуратно относится к материально-техническим ценностям	Соблюдает правила безопасного поведения при работе с компьютерной техникой	Итого
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

3 балла – качество проявляется систематически2 балла – качество проявляется ситуативно

1 балл – качество не проявляется