

#### Описание технологии проведения экзамена:

- формат проведения экзамена: устный экзамен в **формате беседы** по вопросам экзаменационного билета и решения практической задачи,
- порядок формирования экзаменационного билета: 1-й вопрос – с 1 по 17 вопрос из перечня вопросов к экзамену, 2-й вопрос – с 34 по 16 вопрос, задача выбирается из перечня случайным образом.
- требования к ответу: изложение должно быть логичным, придерживаться четкого плана, для пояснения привести примеры.

#### Примерный перечень вопросов к экзамену:

1. Понятие программы и программирования. Виды программного обеспечения.
2. Жизненный цикл программного обеспечения. Стадии разработки программ.
3. Программная документация и стандарты на разработку прикладных программных средств.
4. Понятие алгоритма. Основные требования, предъявляемые к алгоритмам.
5. Способы описания алгоритмов.
6. Основные типы управляющих конструкций. Операторы выбора.
7. Основные типы управляющих конструкций. Циклы.
8. Рекурсивные и итерационные алгоритмы.
9. Методы сортировки данных.
10. Простые типы данных.
11. Структурные типы данных. Структура данных «очередь».
12. Структура данных «стек».
13. Алгоритмическая декомпозиция.
14. Императивный (процедурный) подход к проектированию программ. Понятия функции.
15. Принципы структурного проектирования.
16. Межмодульное взаимодействие.
17. Сущность функционального программирования.
18. Структура программы на языке высокого уровня.
19. Типы данных языка высокого уровня.
20. Реализация линейных вычислений. Операторы языка высокого уровня.
21. Операторы управления в языке высокого уровня.
22. Операторы цикла в языке высокого уровня.
23. Организация программы с помощью функций в языке высокого уровня.
24. Описание и вызов функции в языке высокого уровня.
25. Области действия переменных в языке высокого уровня.
26. Списки и их применение.
27. Кортежи и их применение.
28. Реализация передачи параметров в функцию в языке высокого уровня.
29. Применение документирования функций в языке высокого уровня.
30. Словари и их применение.
31. Реализация функционального программирования в языке высокого уровня.
32. Основы ООП. Реализация класса в языке высокого уровня.
33. Назначение наследования и пример реализации этого механизма в языке высокого уровня.
34. Реализация инкапсуляции в языке высокого уровня.

## Примерный перечень практических заданий

### Задание 1

1. Напишите программу, которая ожидает ввод последовательно произвольного количества вещественных чисел и добавляет их в список. Затем умножает каждое числом меньшее 10 на 1.13, большее 10 – умножает на 0.18, равное точно 10 – оставляет без изменения. После этого сортирует полученный список, и наконец, печатает его, округлив до двух знаков после запятой. Реализуйте также возможность сохранения указанных данных в файл. Тип файла выберите на свое усмотрение, в комментариях обоснуйте свой выбор.

### Задание 2

2. Реализуйте указанную функциональность в задании 1 с помощью функций. Количество функций и их наполнение на ваше усмотрение. В комментариях обоснуйте ваше решение.

### Задание 3

Создайте список (количество чисел не важно) самым простым способом, значения списка любые, например:

`a = [1, 20, 3, 10, 15]`

Реализуйте изменение значений списка (рекомендуется создать новый результирующий список) по следующему правилу – каждое число:

- меньшее 10 должно быть умножено на 1.13,
- большее 10 – умножить на 0.18,
- равное точно 10 – оставить без изменения.

Значения обоих списков выведите на экран.

### Задание 4

Создайте список по имени `cost`, содержащий десять значений самым простым способом, значения списка любые, представляющие стоимость производства десяти продуктов.

Требуется автоматически сгенерировать второй список с именем `price`, который имеет 20% наценку на каждую соответствующую стоимость.

### Задание 5

Дан список температурных изменений в течение дня (целые числа). Известно, что измеряющее устройство иногда сбоит и записывает отсутствие температуры (значение `None`). Выведите среднюю температуру за наблюдаемый промежуток времени, предварительно очистив список от неопределенных значений. Гарантируется, что хотя бы одно определенное значение в списке есть.

### Задание 6

Напишите функцию, которая принимает неограниченное количество числовых аргументов и возвращает кортеж из двух списков: отрицательных значений (отсортирован по убыванию); неотрицательных значений (отсортирован по возрастанию).

### **Задание 7**

Составьте две функции для возведения числа в степень: один из вариантов реализуйте в рекурсивном стиле.

### **Задание 8**

Создайте сценарий, который использует список имен файлов CSV в качестве источника для копирования файлов в плоский файл. Текущая дата и время должны быть добавлены к имени файла в качестве префикса перед копированием.

### **Задание 9**

Напишите функцию, которая воспринимает целое значение и возвращает число с обратным порядком цифр. Например, воспринимает число 7631, а возвращает число 1367.

### **Задание 10**

Напишите функцию, которая возвращает наибольший общий делитель двух целых чисел. Наибольший общий делитель двух целых чисел — это наибольшее целое, на которое без остатка делится каждое из двух чисел.

### **Задание 11**

Напишите функцию, которая вводит среднюю оценку студентов и возвращает 4, если средняя оценка 90 - 100, 3, если средняя оценка 80 - 89, 2, если средняя оценка 70 - 79, 1, если средняя оценка 60 - 69, и 0, если средняя оценка меньше 60.

### **Задание 12**

Напишите функцию, определяющую, является ли число простым (целое число является простым числом, если оно делится только на 1 и на само себя). Используйте эту функцию в программе, которая определяет и печатает все простые числа, лежащие в диапазоне от 1 до 10000.

### **Задание 13**

Напишите программу по следующему описанию: двум переменным присваиваются числовые значения; если значение первой переменной больше второй, то найти разницу значений переменных (вычесть из первой вторую), результат связать с третьей переменной; если первая переменная имеет меньшее значение, чем вторая, то третью переменную связать с результатом суммы значений двух первых переменных; во всех остальных случаях, присвоить третьей переменной значение первой переменной; вывести значение третьей переменной на экран.

### **Задание 14**

Свяжите переменную с любой строкой, состоящей не менее чем из 8 символов. Извлеките из строки первый символ, затем последний, третий с начала и третий с конца. Измерьте длину вашей строки.

### **Задание 15**

Присвойте произвольную строку длиной 10-15 символов переменной и извлеките из нее следующие срезы: первые восемь символов; четыре символа из центра строки; символы с индексами кратными трем.

### **Задание 16**

Создайте два любых списка и свяжите их с переменными. Извлеките из первого списка второй элемент. Измените во втором списке последний объект. Выведите список на экран. Соедините оба списка в один, присвоив результат новой переменной. Выведите получившийся список на экран. "Снимите" срез из соединенного списка так, чтобы туда попали некоторые части обоих первых списков. Срез свяжите с очередной новой переменной. Выведите значение этой переменной. Добавьте в список-срез два новых элемента и снова выведите его.

### **Задание 17**

Создайте словарь, связав его с переменной school, и наполните его данными, которые бы отражали количество учащихся в десяти разных классах (например, 1а, 1б, 2б, 6а, 7в и т.д.). Узнайте сколько человек в каком-нибудь классе. Представьте, что в школе произошли изменения, внесите их в словарь: в трех классах изменилось количество учащихся; в школе появилось два новых класса; в школе расформировали один из классов. Выведите содержимое словаря на экран.