Студент группы ИС-27 Поповян А.Д.

Практическое занятие № 6_1

Тема: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ со списками в IDE PyCharm Community.

Постановка задачи.

Дан целочисленный список размера N, не содержащий одинаковых чисел. Проверить, образуют ли его элементы арифметическую прогрессию. Если образуют, то вывести разность прогрессии, если нет — вывести 0.

Тип алгоритма: Линейный

Текст программы:

```
def my_def(N):
    # Проверяем длину списка
    if len(N) < 2:
        return 0

    # Сортируем список, чтобы корректно проверять арифметическую прогрессию
    N = sorted(N)

    # Находим разность между первым и вторым элементами
    my_sum = N[1] - N[0]

    # Проходим по списку начиная со второго элемента и сравниваем разности
    for i in range(2, len(N)):
        try:
            if N[i] - N[i - 1] != my_sum:
                return 0
        except ValueError:
            print("Ошибка: введено некорректное значение.")


    # Если все разности совпали, возвращаем найденную разницу
    return my_sum


# Пример использования
N = [1, 5, 10, 16, 25]
num = my_def(N)
print(num)  # Вывод: 0
```

Студент группы ИС-27 Поповян А.Д.

Протокол работы программы:

0

Process finished with exit code 0

Студент группы ИС-27 Поповян А.Д.

Практическое занятие № 6_2

Тема: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ со списками в IDE PyCharm Community.

Постановка задачи.

Дан список A размера N. Сформировать новый список B того же размера, элементы которого определяются следующим образом: BK = 2*AK, если AK < 5, AK/2 в противном случае.

Тип алгоритма: Циклический.

Текст программы.

```
#Функция создает новый список B на основе списка A.
def list(A):
    #Создаем пустой список B, в который будем добавлять преобразованные
элементы
    B = []
    #Проходим по каждому элементу списка A
    for element in A:
        try:
            #Если элемент из A меньше 5, то в B записывается удвоенный
элемент.
            if element < 5:
                B.append(2 * element)
            #Либо записывается половина элемента.
            else:
                B.append(element / 2)
        except ValueError:
            print("Ошибка: введено некорректное значение")
    #Возращение функции в B
    return B


#Даем список A
A = [1, 3, 5, 7, 9]
#Сформатировали новый список в B
B = list(A)
#Вывод B
print(B)
```

Студент группы ИС-27 Поповян А.Д.

Протокол работы программы:

[2, 6, 2.5, 3.5, 4.5]

Process finished with exit code 0

Студент группы ИС-27 Поповян А.Д.

Практическое занятие № 6_3

Тема: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ со списками в IDE PyCharm Community.

Постановка задачи.

Дано множество A из N точек (точки заданы своими координатами x, y). Среди всех точек этого множества, лежащих в первой или третьей четверти, найти точку, наиболее близкую к началу координат. Если таких точек нет, то вывести точку с нулевыми координатами. Расстояние R между точками с координатами (x1, y1) и (x2, y2) вычисляется по формуле: R = √(x2 – x1)2 + (y2 – y1)2 . Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

Тип алгоритма: Циклический

Текст программы.

```
#Импортируем модуль math для использования функции sqrt
import math

#Функция находит ближайшую точку к началу координат среди точек,
расположенных в первой или третьей четверти.
def my_def(x_coordinates, y_coordinates):
    #Проверяем, что списки одинаковой длины
    if len(x_coordinates) != len(y_coordinates):
        #Если списки разной длины, выбрасываем ошибку
        raise ValueError("Списки должны быть одинаковой длины")

    # Переменная для хранения ближайших координат
    A = None
    #Начальное значение минимального расстояния
    num2 = float('inf')

    #Проходим по каждому элементу в списках
    for i in range(len(x_coordinates)):
        #Текущая x
        x = x_coordinates[i]
        #Текущая y
        y = y coordinates[i]
```

```
# Если произведение координат больше нуля, точка в первой или третьей
четверти
        if x * y > 0:
            # Вычисляем расстояние до начала координат
            num1 = math.sqrt(x ** 2 + y ** 2)
            # Если расстояние меньше текущего минимума
            if num1 < num2:
                #Обновляем минимальное расстояние
                num2 = num1  # Обновляем минимальное расстояние
                #Сохраняем координаты ближайшей точки
                A = (x, y)

    #Возвращаем ближайшую точку или (0, 0), если подходящей точки нет
    return A or (0, 0)

#Обработа исключение
try:
    #Считываем количество точек
    N = int(input())
    #Считываем абсциссы точек
    x_coordinates = list(map(int, input().split()))
    #Считываем ординаты точек
    y_coordinates= list(map(int, input().split()))

    #Вызываем функцию для поиска ближайшей точки
    B = my_def(x_coordinates, y_coordinates)
    #Выводим результат
    print(B[0], B[1])

#Обработка исключение
except ValueError:
    print("Ошибка: введено некорректное значение.")
```

Протокол работы программы:

12

32

512

32 512

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ линейный и циклический структуры в IDE PyCharm Community. Были использованы языковые конструкции Try, Except. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.