

Практическое занятие №6

Тема: составление программ со списками в IDE PyCharm Community.

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

Задача 1.

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

Текст программы:

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

import random
def is_arithmetic_progression(lst):

    if len(lst) < 2:
        return 0

    lst.sort()

    difference = lst[1] - lst[0]

    for i in range(2, len(lst)):
        if lst[i] - lst[i - 1] != difference:
            return 0

    return difference
n = input('Введи число: ')
while type(n) != int:
    try:
        n = int(n)
    except ValueError:
        print('Введите ЧИСЛО!')
        n = input('Введи число: ')

random_list = random.sample(range(1, 100), n)

print("Список:", random_list)
difference = is_arithmetic_progression(random_list)
```

```
print("Разность прогрессии:", difference)
```

Протокол работы программы:

2

Process finished with exit code 0

Задача 2.

Постановка задачи.

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

Текст программы:

```
# Дан список A размера N. Сформировать новый список B того же размера, элементы
# которого определяются следующим образом: B_k = 2 * A_k, если A_k < 5, A_k/2 в противном
# случае.
N = input("Введите размер списка N: ")
while type(N) != int:
    try:
        N = int(N)
    except ValueError:
        print("Введите ЧИСЛО!")
        N = input("Введи число: ")

# Создаем список A из N чисел, введенных пользователем
A = []
print("Введите", N, "элементов списка:")
for i in range(N):
    num = input(f"Элемент {i + 1}: ")
    while type(num) != int:
        try:
            num = int(num)
        except ValueError:
            print("Введите ЧИСЛО!")
            num = input(f"Элемент {i + 1}: ")
    A.append(num)

# Создаем новый список B, преобразуя элементы из списка A
B = []
for x in A:
    if x < 5:
        B.append(2 * x) # Если элемент меньше 5, умножаем на 2
    else:
        B.append(x / 2) # Иначе делим на 2
```

```
# Выводим результаты
print("Исходный список A:", A)
print("Новый список B:", B)
```

Протокол работы программы:

Исходный список A: [1, 4, 6, 8, 3]

Новый список B: [2, 8, 3.0, 4.0, 6]

Process finished with exit code 0

Задача 3.

Постановка задачи. 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: $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$. Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

Текст программы:

```
# 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:  $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$ . Для хранения данных о каждом
#наборе точек следует использовать по два списка: первый список для хранения абсцисс,
#второй — для хранения ординат.

import math

def find_closest_point(points):

    x_coords = []
    y_coords = []

    for point in points:
        x, y = point
        x_coords.append(x)
        y_coords.append(y)

    closest_point = None
```

```
min_distance = float('inf')

for i in range(len(points)):
    x = x_coords[i]
    y = y_coords[i]

    if (x > 0 and y > 0) or (x < 0 and y < 0):
        distance = math.sqrt(x**2 + y**2)
        if distance < min_distance:
            min_distance = distance
            closest_point = (x, y)

if closest_point is None:
    return (0, 0)
else:
    return closest_point

points = [(1, 2), (-1, -2), (3, 4), (-3, -4), (0, 1)]
closest = find_closest_point(points)
print("Ближайшая точка:", closest)
```

Протокол работы программы:

Ближайшая точка: (1, 2)

Process finished with exit code 0

Вывод: В ходе практической работы я закрепил навыки усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community