Студент группы ИС-27 Ковалев.Р.Д.

Практическое занятие № 6

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 1:

Дан список A размера N. Вывести вначале его элементы с четными номерами (в порядке возрастания номеров), а затем — элементы с нечетными номерами (также в порядке возрастания номеров): A2, A4, A6, . . ., A1, A3, A5, … . Условный оператор не использовать.

Тип алгоритма: Циклический

Текст программы:

```python
#Дано число R и список размера N. Найти два соседних элемента списка, сумма
#которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания
#их индексов (определение наиболее близких чисел - то есть такой элемент АК, для
#которого величина |АК - R| является минимальной).
def couple(R, A):  1 usage  Thegame2022 *
    if len(A) < 2:
        raise ValueError("Список должен содержать хотя бы два элемента")
    closest_pair = (A[0], A[1])  #Инициализируем начальную пару
    min_difference = abs((A[0] + A[1]) - R)
    for i in range(len(A) - 1):
        current_sum = A[i] + A[i + 1]
        difference = abs(current_sum - R)
        if difference < min_difference:
            min_difference = difference
            closest_pair = (A[i], A[i + 1])
    return sorted(closest_pair)

if __name__ == "__main__":
    try:
        #Запрашиваем у пользователя ввод числа R
        R = float(input("Введите число R: "))
        #Запрашиваем ввод элементов для списка A
        B = input("Введите элементы списка A: ")
        #Если цифры введены без пробелов,то обрабатываются как отдельные цифры
        if ' ' not in B:
            A = []
            for char in B:
                A.append(int(char))
        else:
            A = list(map(int, B.split()))
        #Выводим исходный список
        print(f"Исходный список: {A}")
        #Вызываем функцию поиска ближайшей пары
        closest_pair = couple(R, A)
        #Выводим ближайшую пару
        print(f"Ближайшая пара к {R}: {closest_pair}")
    except ValueError as e:
        print(f"Произошла ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")
```

```python
#Дан список A размера N. Вывести вначале его элементы с четными номерами (в
#порядке возрастания номеров), а затем – элементы с нечетными номерами (также в
#порядке возрастания номеров): A2, A4, A6, . . ., A1, A3, A5, ... . Условный оператор не
#использовать.
def print_even_and_odd_elements(A):  1 usage  Thegame2022 *
    try:
        #Проверка на пустой список
        if len(A) == 0:
            raise ValueError("Список пуст.")
        #Получаем элементы с четными индексами
        even_elements = A[1::2]
        #Получаем элементы с нечетными индексами
        odd_elements = A[::2]
        #Объединяем срезы и выводим результат
        result = even_elements + odd_elements
        print("Результирующий список:", result)

    except ValueError as e:
        print(f"Произошла ошибка: {e}")


if __name__ == "__main__":
    try:
        #Запрашиваем ввод элементов для списка A
        B = input("Введите элементы списка A: ")
        #Если цифры введены без пробелов,то обрабатываются как отедльные цифры
        if ' ' not in B:
            A = []
            for num in B:
                A.append(int(num))
        else:
            A = B.split()
            A = list(map(int, A))

        print("Исходный список:", A)

        print_even_and_odd_elements(A)

    except ValueError as e:
        print(f"Произошла ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")
```

Студент группы ИС-27 Ковалев.Р.Д.

Протокол работы программы:

Введите элементы списка A через пробел: 1 2 3 4 5 6 7 8 9

Исходный список: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Результирующий список: [2, 4, 6, 8, 1, 3, 5, 7, 9]

Process finished with exit code 0

Постановка задачи 2:

Дано число R и список размера N. Найти два соседних элемента списка, сумма которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания их индексов (определение наиболее близких чисел - то есть такой элемент AK, для которого величина |AK - R| является минимальной).

Тип алгоритма: Циклический.

Текст программы:

```
#Дан список размера N и целое число K (1 < K < N). Осуществить сдвиг элементов
#списка влево на K позиций (при этом AN перейдет в AN-K, AN-1 – в AN-K-1, ..AK+1 – в
#A1, а исходное значение K первых элементов будет потеряно). Последние K
#элементов полученного списка положить равными 0.
def shift_list_left(A, K):  1 usage  Thegame2022 *
    try:
        #Проверка на допустимость значения K
        if not (1 <= K < len(A)):
            raise ValueError("Значение K должно быть больше 0 и меньше длины списка.")
        #Выполнение сдвига элементов влево на K позиций
        shifted_A = A[K:] + A[:K]
        #Вывод исходного и результирующего списков
        print(f"Исходный список: {A}")
        print(f"Сдвинутый список: {shifted_A}")
    except ValueError as e:
        print(f"Произошла ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")

if __name__ == "__main__":
    try:
        #Запрашиваем ввод элементов для списка A
        B = input("Введите элементы списка A: ")
        #Если цифры введены без пробелов,то обрабатываются как отдельные цифры
        if ' ' not in B:
            A = []
            for char in B:
                A.append(int(char))
        else:
            A = list(map(int,  B.split()))
        #Запрашиваем у пользователя ввод числа K
        K = int(input("Введите значение K: "))
        #Вызываем функцию для сдвига списка
        shift_list_left(A, K)
    except ValueError as e:
        print(f"Произошла ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")
```

Протокол работы программы:

Введите число R: 3

Введите элементы списка A через пробел: 1 2 3 4 5 6 7 8 9

Исходный список: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Ближайшая пара к 3.0: [1, 2]

Process finished with exit code 0

Студент группы ИС-27 Ковалев.Р.Д.

Постановка задачи 3:

Дан список размера N и целое число K (1 < K < N). Осуществить сдвиг элементов списка влево на K позиций (при этом AN перейдет в AN-K, AN-1 — в AN-K-1, ..AK+1 — в A1, а исходное значение K первых элементов будет потеряно). Последние K элементов полученного списка положить равными 0.

Тип алгоритма: Циклический

Текст программы:

```python
#Дан список A размера N. Вывести вначале его элементы с четными номерами (в
#порядке возрастания номеров), а затем – элементы с нечетными номерами (также в
#порядке возрастания номеров): A2, A4, A6, . . ., A1, A3, A5, ... . Условный оператор не
#использовать.
def print_even_and_odd_elements(A):  1 usage  Thegame2022 *
    try:
        #Проверка на пустой список
        if len(A) == 0:
            raise ValueError("Список пуст.")
        #Получаем элементы с четными индексами
        even_elements = A[1::2]
        #Получаем элементы с нечетными индексами
        odd_elements = A[::2]
        #Объединяем срезы и выводим результат
        result = even_elements + odd_elements
        print("Результирующий список:", result)

    except ValueError as e:
        print(f"Произошла ошибка: {e}")


if __name__ == "__main__":
    try:
        #Запрашиваем ввод элементов для списка A
        B = input("Введите элементы списка A: ")
        #Если цифры введены без пробелов,то обрабатываются как отедльные цифры
        if ' ' not in B:
            A = []
            for num in B:
                A.append(int(num))
        else:
            A = B.split()
            A = list(map(int, A))

        print("Исходный список:", A)

        print_even_and_odd_elements(A)

    except ValueError as e:
        print(f"Произошла ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")
```

Студент группы ИС-27 Ковалев.Р.Д.

Протокол работы программы:

Введите элементы списка A через пробел: 1 2 3 4 5 6 7 8 9

Введите значение K: 3

Исходный список: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Сдвинутый список: [4, 5, 6, 7, 8, 9, 0, 0, 0]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ циклической 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.