Студент группы ИС-26 Мазманов А.Ф

Практическое занятие № 6

Тема: составление программ cо списками в IDE PyCharm Community.

Цели: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, с использованием списков. Приобрести навыки составление со списками в IDE PyCharm Community.

Вариант20

Номер 1

Блок-схема для данного кода на Python:

graph TD

A[Начало] --> B{Ввод N, K, L};

B -- 1 < K < L < N --> C[Проверка условия];

B -- Условие не выполняется --> D[Вывод ошибки];

C --> E[Инициализация elements = [], summ = 0, count = 0];

E --> F{Цикл i от 0 до N-1};

F --> G[Ввод элемента и добавление в elements];

G --> F;

F --> H{Цикл i от K-1 до L-1};

H --> I[summ += elements[i]; count++];

I --> H;

H --> J{count > 0?};

J -- Да --> K[summ / count];

J -- Нет --> L[Возврат 0];

K --> M[Вывод результата];

D --> M;

L --> M;

M[Конец]

Код программы

*'''*  
*Дан список размера N и целые числа К и L (1 < K < L < N ).*  
*Найти среднее арифметическое элементов список с номерами от К до L включительно.*  
*'''*  
  
def arithmetic\_mean(numN, numK, numL):  
 elements = []  
 linit1 = numK  
 linit2 = numL  
 summ = int(0)  
 count = int(0)  
  
 for i in range(numN):  
 elements.append(int(input(f"Введи число {i+int(1)} \n")))  
  
 for j in range(linit1, linit2 + 1):  
 summ += elements[j]  
 count += int(1)  
  
 return float(summ / count)  
  
  
try:  
 N = int(input("Введи N размер списка:\n"))  
 K = int(input("Введи K (1 < K < L < N):\n"))  
 L = int(input("Введи L (1 < K < L < N):\n"))  
 if not (int(1) < K < L < N):  
 raise ValueError("Условие 1 < K < L < N не выполняется")  
 result = arithmetic\_mean(N, K, L)  
 print(f"Результат --, {result}")  
except ValueError as va:  
 print(va)

Протокол работы программы номер 1

Введи N размер списка:

10

Введи K (1 < K < L < N):

2

Введи L (1 < K < L < N):

5

Введи число 1

1

Введи число 2

2

Введи число 3

3

Введи число 4

4

Введи число 5

5

Введи число 6

6

Введи число 7

7

Введи число 8

8

Введи число 9

9

Введи число 10

10

4.5

Номер 2

Блок-схема для данного кода на Python:

graph TD

A[Начало] --> B{Ввод N, K, L};

B -- 1 < K < L < N --> C[Проверка условия];

B -- Условие не выполняется --> D[Вывод ошибки];

C --> E[Инициализация elements = [], summ = 0, count = 0];

E --> F{Цикл i от 0 до N-1};

F --> G[Ввод элемента и добавление в elements];

G --> F;

F --> H{Цикл i от K-1 до L-1};

H --> I[summ += elements[i]; count++];

I --> H;

H --> J{count > 0?};

J -- Да --> K[summ / count];

J -- Нет --> L[Возврат 0];

K --> M[Вывод результата];

D --> M;

L --> M;

M[Конец]

Код программы

Код программы

*'''*  
*Дан целочисленный список размера N. Найти максимальное количество его*  
*одинаковых элементов.*  
*'''*  
  
def max\_count(number):  
 elements = []  
 element\_counts = {}  
  
 for i in range(number):  
 elements.append(int(input(f"Введи число {i+int(1)} \n")))  
  
 orig\_arr = elements.copy()  
  
 for element in elements:  
 if element in element\_counts:  
 element\_counts[element] += int(1)  
 else:  
 element\_counts[element] = int(1)  
  
  
 MAX\_COUNT = max(element\_counts.values())  
 return MAX\_COUNT, orig\_arr  
  
  
try:  
 N = int(input("Введи размер списка \n"))  
 if N < int(0):  
 raise ValueError("Только целые, положительный числа. \n")  
 resalt = max\_count(N)  
 print(f"Максимально одинаковых элементов - {resalt[int(0)]}")  
 print(f"Изначальный список - {resalt[int(1)]}")  
except ValueError as va:  
 print(va)

Протокол работы программы номер 2

Введи размер списка

6

Введи число 1

1

Введи число 2

3

Введи число 3

2

Введи число 4

3

Введи число 5

2

Введи число 6

3

Максимально одинаковых элементов - 3

Изначальный список - [1, 3, 2, 3, 2, 3]

Номер 3

Блок-схема для данного кода на Python:

graph TD

A[Начало] --> B{Ввод N};

B -- N < 0 --> C[Вывод ошибки: "Только целые, положительные числа"];

B -- N >= 0 --> D[Инициализация elements = []];

D --> E{Цикл i от 0 до N-1};

E --> F[Ввод элемента и добавление в elements];

F --> E;

E --> G[arr = len(elements)];

G --> H[arr\_orig = elements.copy()];

H --> I{Цикл j от 0 до arr-1};

I --> J{Цикл elem от 0 до arr-j-1};

J --> K{elements[elem] > elements[elem+1]?};

K -- Да --> L[elements[elem] и elements[elem+1] меняются местами];

K -- Нет --> J;

J --> I;

I --> M[Вывод: "Изначальный список - arr\_orig"];

M --> N[Вывод: "Отсортированный список - elements"];

C --> O[Конец];

N --> O;

Код программы

*'''*  
*Дан список размера N, все элементы которого, кроме одного, упорядочены по убыванию.*  
*Сделать список упорядоченным, переместив элемент, нарушающий упорядоченность, на новую позицию.*  
*'''*  
  
def bubble\_sort(number):  
 elements = []  
 for i in range(number):  
 elements.append(int(input(f"Введи число {i+int(1)} \n")))  
  
 arr = len(elements)  
 arr\_orig = elements.copy()  
  
 for j in range(arr):  
 for elem in range(int(0), arr-j-int(1)):  
 if elements[elem] > elements[elem + int(1)]:  
 elements[elem + int(1)] ,elements[elem] = elements[elem], elements[elem+1]  
 return elements, arr\_orig  
  
try:  
 N = int(input("Введи размер списка \n"))  
 if N < int(0):  
 raise ValueError("Только целые, положительный числа. \n")  
 resalt = bubble\_sort(N)  
 print(f"Изначальный список - {resalt[int(1)]}")  
 print(f"Отсортированный список - {resalt[int(0)]}")  
except ValueError as va:  
 print(va)

Протокол работы программы номер 3

Введи размер списка

6

Введи число 1

4

Введи число 2

1

Введи число 3

3

Введи число 4

2

Введи число 5

5

Введи число 6

6

Изначальный список - [4, 1, 3, 2, 5, 6]

Отсортированный список - [1, 2, 3, 4, 5, 6]

Тип задачи: Линейный

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ с использованиями списков в ID PyCharm Community.