Колядко А, Пигарев А 221703

**Код до изменения:**

#include <iostream> // Подключение стандартной библиотеки

#include <string>

using namespace std;

int numArr[1000], cnt;

int main() //Основное тело программы

{

char arr[1000] = { ' ' };

int currNum = 0;

int k = 1;

cout << "Enter a numeric string: "; // Вывод сообщения

char c = getchar();

while (c != '\n') {

arr[k++] = c;

c = getchar();

}

cout << '\n'; // Перевод каретки

int q = 1;

for (int i = k; i >= 0; i--) {

if (isdigit(arr[i])) {

currNum += q \* (arr[i] - '0');

q \*= 10;

}

else if (q != 1) {

numArr[cnt++] = currNum;

currNum = 0;

q = 1;

}

}

}

**Код после изменения:**

#include <iostream> // Подключение стандартной библиотеки

#include <string>

using namespace std;

int SolveCurrNum(int q, char arri){

return q \* (arri - '0');

}

char\* getString(int\* k){

char arr[1000] = { ' ' };

printf ("Enter a numeric string: "); // Вывод сообщения

char c = getchar();

while (c != '\n') {

arr[\*k++] = c;

c = getchar();

}

return arr;

}

int numArr[1000], cnt;

int main() //Основное тело программы

{

int currNum = 0;

int k = 1;

char\* arr = getString(&k);

printf("\n"); // Перевод каретки

int q = 1;

for (int i = k; i >= 0; i--) {

if (isdigit(arr[i])) {

currNum += SolveCurrNum(q, arr[i]);

q \*= 10;

}

else if (q != 1) {

numArr[cnt++] = currNum;

currNum = 0;

q = 1;

}

}

}

Таблица 1 – Метрика ABC

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 19 | 7 | 3 |
| = – 12  -- – 1  ++ - 2  += - 1  \*= - 1  Инициализация переменных (в форме без оператора присваивания) – 2 | Вызов польз. функций – 2  printf – 2  isdigit – 1  getchar - 2 | != – 2  >= – 1 |

Diff =

Метрика Кафура

Информационная сложность модуля относительно структуры данных J (arr)

*J = W \* R + W \* RW + RW \*R + RW \* (RW - 1)*

*W =1 (*getString*)*

*R =2(isdigit,* SolveCurrNum*)*

*RW= 1(main)*

*J = 1 \* 2 + 1 \* 1 + 1 \*2 + 1 \* (1 - 1) = 5*

*Где W — число процедур, которые только обновляют структуру данных;*

*R — только читают информацию из структуры данных;*

*RW — и читают, и обновляют информацию в структуре данных*