|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**ЛАБОРАТОРНАЯ РАБОТА №3**

**«Разработка программного кода. Расчет основных показателей надежности программного продукта.»**

**ДИСЦИПЛИНА: «Основы программной инженерии»**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-21Б | |  |  | ( | Суриков Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Амеличев Г. Э. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** знакомство с методами получения программного кода с заданной функциональностью и степенью качества программного модуля.

**Задачи:** Написать вариант программы, провести оптимизацию программы по выбранному параметру; сравнить характеристики исходной программы и модуля, полученной в результате оптимизации, анализируя время выполнения.

**Вариант 13:** Сортировка выбором.

**Листинг программы:**

1 *#include* *<iostream>*

2 *#include* *<chrono>*

3 *#include* *<vector>*

4

5 using namespace std;

6

7 void selectionSort(vector<int>& arr, int n) {

8 for (int i = 0; i < n-1; i++) {

9 int minIdx = i;

10 for (int j = i+1; j < n; j++)

11 if (arr[j] < arr[minIdx])

12 minIdx = j;

13 swap(arr[minIdx], arr[i]);

14 }

15 }

16

17 int main() {

18 for (int n : {50, 500, 1000}) {

19 vector<int> arr(n);

20 for (int i = 0; i < n; i++)

21 arr[i] = rand() % 1000;

22

23 auto start = chrono::high\_resolution\_clock::now();

24 selectionSort(arr, n);

25 auto end = chrono::high\_resolution\_clock::now();

26 auto duration = chrono::duration\_cast<chrono::milliseconds>(end - start).count();

27

28 cout << "Для " << n << " элементов время сортировки: " << duration << " мс" << endl;

29 }

30

31 return 0;

32 }

**Результат выполнения программы:**

Для 50 элементов время сортировки: 0 мс (7 микросекунд)

Для 500 элементов время сортировки: 0 мс (522 микросекунды)

Для 1000 элементов время сортировки: 2 мс (2038 микросекунд)

**Результаты выполнения замеров:**

|  |  |
| --- | --- |
| **Кол-во данных** | **Время в мс** |
| 50 | 0 |
| 500 | 0 |
| 1000 | 2 |

**Вывод:** в ходе данной лабораторной работы было произведено знакомство с методами получения программного кода с заданной функциональностью и степенью качества программного модуля.