**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра АПУ**

**ОТЧЕТ**

**по лабораторной работе №1**

**на тему: «Простые методы сортировки»**

**по дисциплине «Алгоритмы и структуры данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3391 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Николаев В.Ю. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Миненко М.В. |

Санкт-Петербург

2024 г.

## Цель работы

Получить практические навыки работы с простыми методами сортировки массивов структур.

## Задачи

1. Разработать структуру данных для хранения информации о студентах.
2. Реализовать два метода сортировки массива структур: сортировку обменом и выбором по убыванию курса.
3. Подсчитать и вывести на экран количество сравнений и обменов для каждого метода сортировки.
4. Обеспечить корректность ввода данных пользователем через меню программы.

## Описание программы

Программа реализует структуру данных «Студент», включающую следующие поля:

* ФИО
* Курс
* Факультет

## Алгоритмы сортировки

1. Сортировка обменом:

|  |
| --- |
| void bubble\_sort(const Student\* students, int student\_count) {  Student\* temp\_students = new Student[student\_count];  for (int i = 0; i < student\_count; ++i) {  temp\_students[i] = students[i];  }   int compares = 0, swaps = 0;  bool swapped;   for (int i = 0; i < student\_count - 1; i++) {  swapped = false;  for (int j = 0; j < student\_count - i - 1; j++) {  compares++;  if (temp\_students[j].course < temp\_students[j + 1].course) {  std::swap(temp\_students[j], temp\_students[j + 1]);  swaps++;  swapped = true;  }  }  if (!swapped) break;  }   print\_students(temp\_students, student\_count);  std::cout << "Пузырьковая сортировка: сравнений - " << compares << ", обменов - " << swaps << ".\n";   delete[] temp\_students; } |

1. Сортировка выбором:

|  |
| --- |
| void selection\_sort(const Student\* students, int student\_count) {  Student\* temp\_students = new Student[student\_count];  for (int i = 0; i < student\_count; ++i) {  temp\_students[i] = students[i];  }   int compares = 0, swaps = 0;  for (int i = 0; i < student\_count - 1; i++) {  int min\_idx = i;  for (int j = i + 1; j < student\_count; j++) {  compares++;  if (temp\_students[j].course > temp\_students[min\_idx].course) {  min\_idx = j;  }  }  if (min\_idx != i) {  std::swap(temp\_students[min\_idx], temp\_students[i]);  swaps++;  }  }   print\_students(temp\_students, student\_count);  std::cout << "Сортировка выбором: сравнений - " << compares << ", обменов - " << swaps << ".\n";   delete[] temp\_students; } |

## Меню программы

Программа имеет следующее меню для взаимодействия с пользователем:

1. Ввод данных
2. Вывод исходных данных
3. Сортировка пузырьком
4. Сортировка выбором
5. Выход

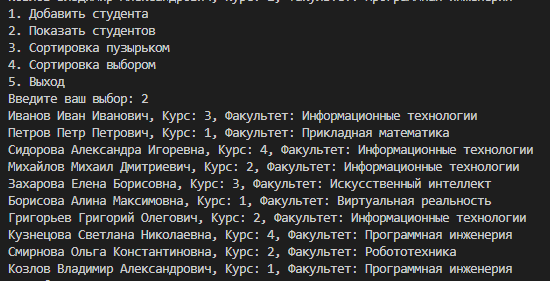
Кода меню:

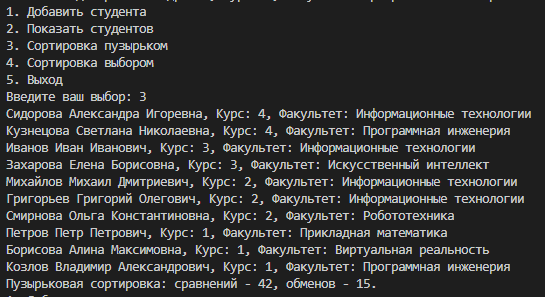
|  |
| --- |
| int main() {  int max\_students;  std::cout << "Введите максимальное количество студентов: ";  std::cin >> max\_students;  std::cin.ignore();   Student\* students = new Student[max\_students];  int student\_count = 0;   char command;  bool running = true;  while (running) {  std::cout << "1. Добавить студента\n2. Показать студентов\n3. Сортировка пузырьком\n4. Сортировка выбором\n5. Выход\nВведите ваш выбор: ";  std::cin >> command;  std::cin.ignore();  switch (command) {  case '1':  add\_student(students, student\_count, max\_students);  break;  case '2':  print\_students(students, student\_count);  break;  case '3':  bubble\_sort(students, student\_count);  break;  case '4':  selection\_sort(students, student\_count);  break;  case '5':  running = false;  break;  default:  std::cout << "Неверный ввод, попробуйте снова.\n";  }  }   delete[] students;  return 0; } |

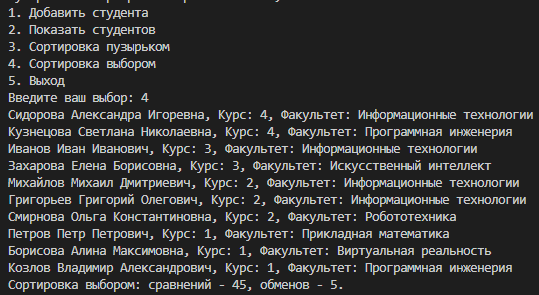
## Результаты работы программы

Пример ввода данных:  


|  |  |  |
| --- | --- | --- |
| ФИО | Курс | Факультет |
| Иванов Иван Иванович | 3 | Информационные технологии |
| Петров Петр Петрович | 1 | Прикладная математика |
| Сидорова Александра Игоревна | 4 | Информационные технологии |
| Михайлов Михаил Дмитриевич | 2 | Информационные технологии |
| Захарова Елена Борисовна | 3 | Искусственный интеллект |
| Борисова Алина Максимовна | 1 | Виртуальная реальность |
| Григорьев Григорий Олегович | 2 | Информационные технологии |
| Кузнецова Светлана Николаевна | 4 | Программная инженерия |
| Смирнова Ольга Константиновна | 2 | Робототехника |
| Козлов Владимир Александрович | 1 | Программная инженерия |

Исходный массив студентов:  


После сортировки пузырьком: 

После сортировки выбором: 

## Выводы

В ходе выполнения лабораторной работы были разработаны два алгоритма сортировки для массива структур. Оба метода успешно отсортировали студентов по убыванию курса. Сортировка выбором показала меньшее количество сравнений, чем сортировка обменом.