|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет  имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатика, искусственный интеллект и системы управления» (ИУ) |

|  |  |
| --- | --- |
| КАФЕДРА | «Информационная безопасность» (ИУ8) |

Лабораторная работа № 4

ПО КУРСУ

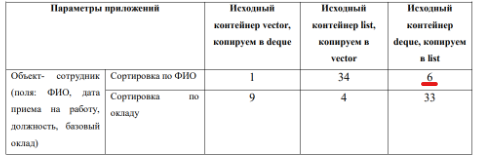
«Алгоритмические языки»

на тему «Использование объектов своих классов в последовательных контейнерах библиотеки STL »

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-24 |  |  |  | Иванова А. А |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | Барыкин Д. В |
|  |  |  |  |  | (И.О. Фамилия) |

**Цель работы:**

Наработка навыков по использованию объектов своих классов в последовательных контейнерах библиотеки STL



**Текст программы:**

#include <iostream>

#include <deque>

#include <list>

#include <algorithm>

#include <fstream>

#include <string>

class Employee {

public:

std::string name;

std::string hire\_date;

std::string position;

double base\_salary;

Employee() : name(""), hire\_date(""), position(""), base\_salary(0.0) {}

Employee(const std::string& n, const std::string& h, const std::string& p, double s)

: name(n), hire\_date(h), position(p), base\_salary(s) {}

friend std::ostream& operator<<(std::ostream& os, const Employee& e) {

os << e.name << ": " << e.hire\_date << ", " << e.position << ", " << e.base\_salary << "$";

return os;

}

bool operator<(const Employee& other) const {

return name > other.name;

}

};

bool compareByName(const Employee& a, const Employee& b) {

return a.name > b.name;

}

int main() {

std::deque<Employee> employees;

std::ifstream input("input.txt");

std::string name, hire\_date, position;

double base\_salary;

while (getline(input, name)) {

getline(input, hire\_date);

input >> position >> base\_salary;

input.ignore();

employees.push\_back(Employee(name, hire\_date, position, base\_salary));

}

input.close();

std::ofstream output("output.txt");

output << "Original container:\n";

std::cout << "Original container:\n";

for (const auto& e : employees) {

output << e << std::endl;

std::cout << e << std::endl;

}

std::deque<Employee> originalEmployees = employees;

std::list<Employee> copiedEmployees;

copiedEmployees.resize(originalEmployees.size());

std::copy(originalEmployees.begin(), originalEmployees.end(), copiedEmployees.begin());

std::sort(employees.begin(), employees.end());

std::list<Employee> employees\_list(employees.begin(), employees.end());

output << "\nSorted container:\n";

std::cout << "\nSorted container:\n";

for (const auto& e : employees\_list) {

output << e << std::endl;

std::cout << e << std::endl;

}

output << "\nCopied container:\n";

std::cout << "\nCopied container:\n";

for (const auto& e : copiedEmployees) {

output << e << std::endl;

std::cout << e << std::endl;

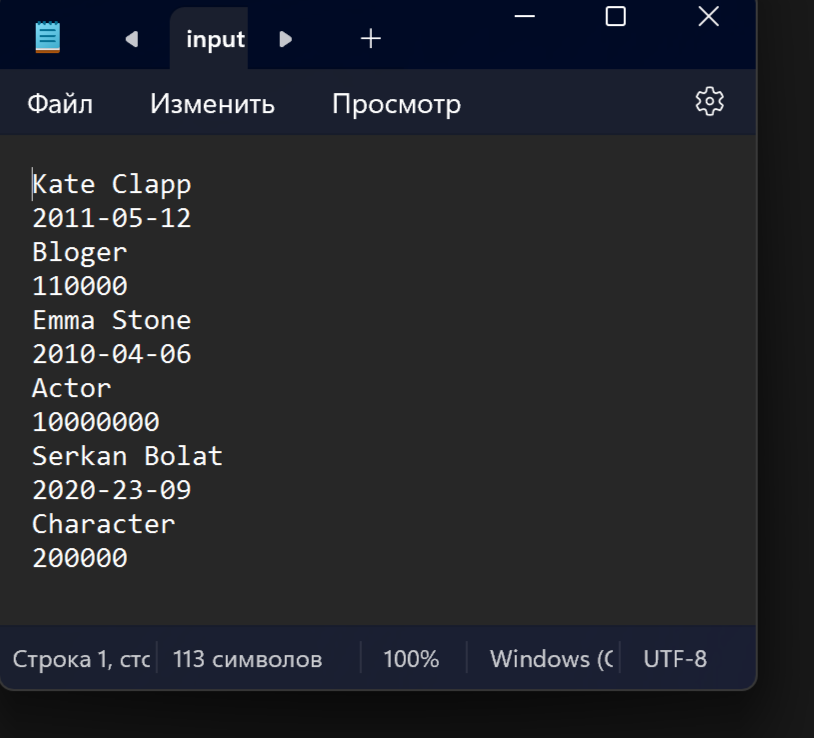
}

output.close();

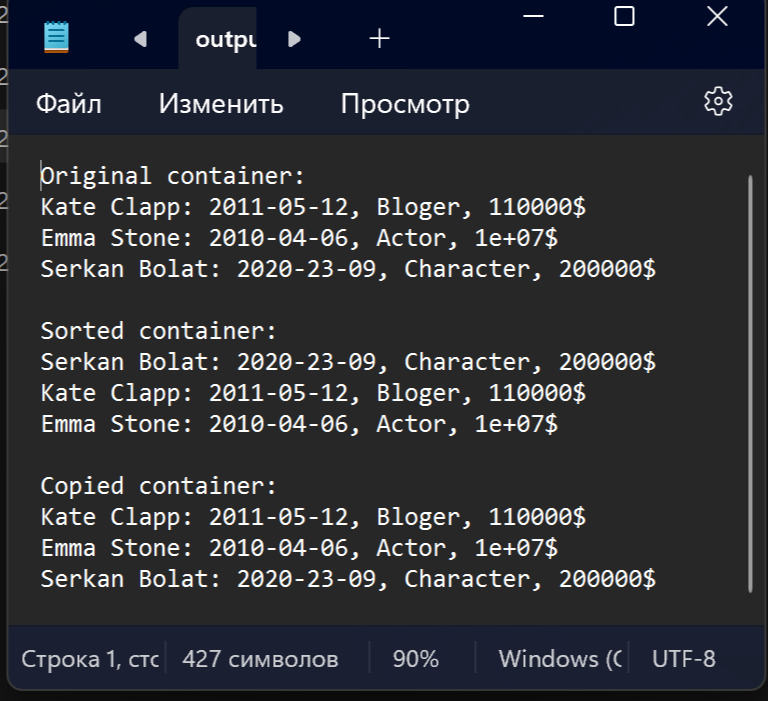
return 0;

}

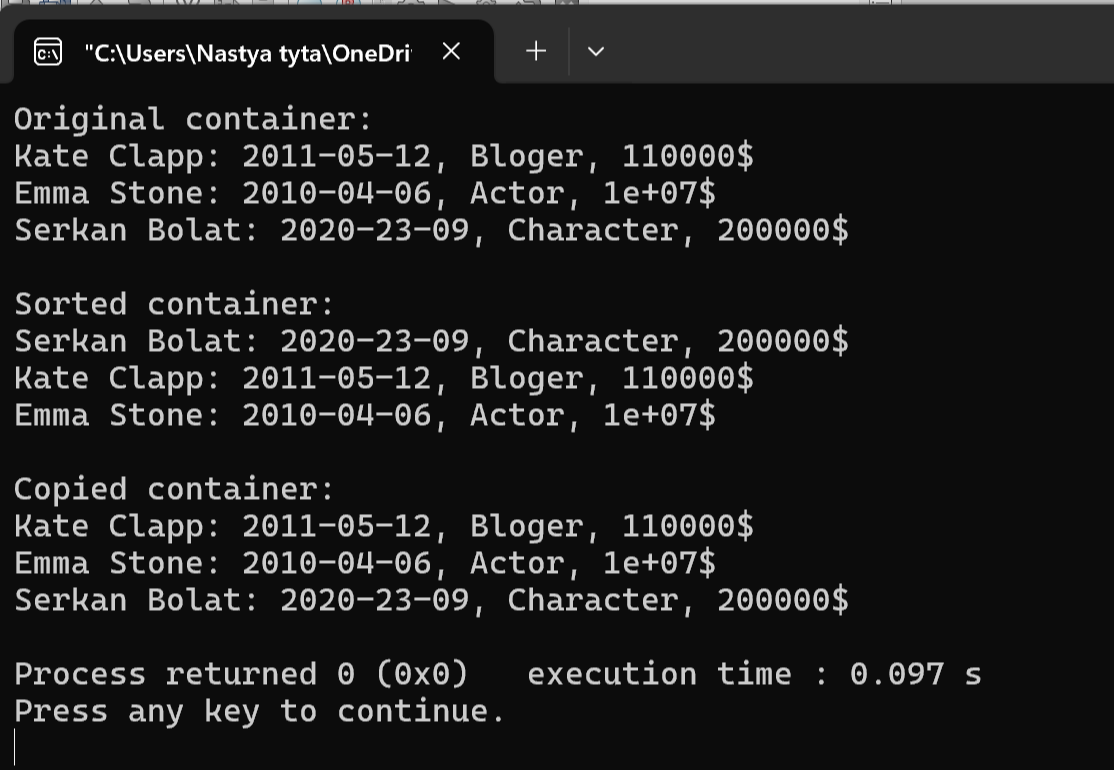
**Файл input.txt:**

****

**Файл output.txt:**



**Консоль:**



**Вывод:**

Выполнив лабораторную работу №4, я изучила тему «Использование объектов своих классов в последовательных контейнерах библиотеки STL» и улучшила свои навыки в этой теме.