Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Логика и основы алгоритмизации

в инженерных задачах»

на тему «Динамические списки»

Выполнили:

студенты группы 22ВВВ1

Дунюшкин В.А.

Ивлюшин М.Д.

Проверили:

к.т.н, доцент Юрова О.В.

к.э.н, доцент Акифьев И.В.

Пенза 2023

**Цель работы:**

Цель данной лабораторной работы заключается в изучении и практическом применении динамических списков в программировании .

**Лабораторное задание:**

1. Реализовать приоритетную очередь, путём добавления элемента в список в

соответствии с приоритетом объекта (т.е. объект с большим приоритетом

становится перед объектом с меньшим приоритетом).

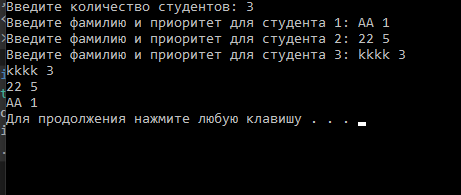
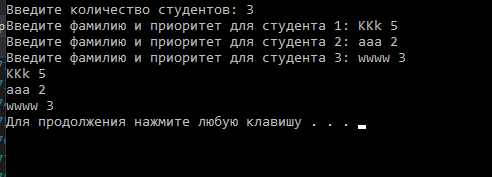
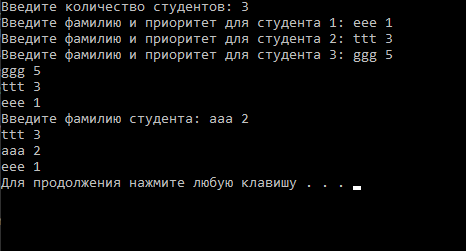
2. \* На основе приведенного кода реализуйте структуру данных Очередь.

3. \* На основе приведенного кода реализуйте структуру данных Стек.

**Ход работы:**

Инициализируем структуры данных Student, Queue и Stack для реализации приоритетной очереди, очереди и стека соответственно.  
 Далее реализуем функции для работы с ними (сравнение для приоритетной очереди, добавление и извлечение эл-та для очереди и стека)  
 Дальше в функции main мы выбираем какую структуру данных будем использовать.  
 Приоритетная очередь:  
 -Считывается количество студентов  
 -Выделяется память для массива студентов  
 -В цикле считываются данные о студентах  
 -Вызывается сортировка qsort по приоритету  
 -Выводятся отсортированные студенты  
 Обычная очередь:  
 -Создается очередь заданного размера  
 -В цикле считываются студенты и добавляются в -очередь через enqueue  
 -В цикле извлекаются студенты из очереди через dequeue и выводятся  
 Стек:  
 -Создается стек заданного размера  
 -В цикле считываются студенты и добавляются в -стек через push  
 -В цикле студенты извлекаются из стека через pop и выводятся  
 Если выбрана неверная структура данных - выводится сообщение об ошибке.

**Результаты работы программы:**



**Вывод**

В ходе лабораторной работы научились создавать и работать с приоритетной очередью, структурой данных – Стек и структурой данных – Очередь.

**Листинг**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

// Структуры данных

struct Student {

string lastName;

int priority;

};

// Функция для сравнения студентов

bool compareStudents(const Student& a, const Student& b) {

return a.priority > b.priority;

}

int main() {

system("chcp 1251");

system("cls");

int choice;

cout << "Выберите структуру данных:" << endl;

cout << "1 - Приоритетная очередь" << endl;

cout << "2 - Обычная очередь" << endl;

cout << "3 - Стек" << endl;

cin >> choice;

system("cls");

if (choice == 1) {

int numStudents;

cout << "Введите количество студентов: ";

cin >> numStudents;

vector<Student> students(numStudents);

for (int i = 0; i < numStudents; i++) {

cout << "Введите фамилию и приоритет для студента " << i + 1 << ": ";

cin >> students[i].lastName >> students[i].priority;

}

sort(students.begin(), students.end(), compareStudents);

for (int i = 0; i < numStudents; i++) {

cout << students[i].lastName << " " << students[i].priority << endl;

}

int i = 0;

cout << "Введите фамилию студента: ";

cin >> students[i].lastName >> students[i].priority;

sort(students.begin(), students.end(), compareStudents);

for (int i = 0; i < numStudents; i++) {

cout << students[i].lastName << " " << students[i].priority << endl;

}

}

else if (choice == 2) {

vector<Student> q;

int n;

cout << "Введите количество студентов: ";

cin >> n;

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

Student s;

cout << "Введите фамилию и приоритет для студента " << i + 1 << ": ";

cin >> s.lastName >> s.priority;

q.push\_back(s);

}

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

Student s = q[i];

cout << s.lastName << " " << s.priority << endl;

}

}

else if (choice == 3) {

vector<Student> s;

int n;

cout << "Введите количество студентов: ";

cin >> n;

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

Student student;

cout << "Введите фамилию и приоритет для студента " << i + 1 << ": ";

cin >> student.lastName >> student.priority;

s.push\_back(student);

}

for (int i = n - 1; i >= 0; i--) {

Student temp = s[i];

cout << temp.lastName << " " << temp.priority << endl;

}

}

else {

cout << "Неверный выбор!" << endl;

}

system("pause");

}