САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Практическая работа №15

«Использование STL»

Выполнил: Мороз И.О.

Группа К3120

Проверил: Осипов Н.А.

Санкт-Петербург

2018 г.

# Упражнение 1.

1. На основе исходных файлов были реализованы новые методы для класса Group

int Group::getSize() {

return masSt.size();

}

void Group::addStudent(Student newStudent) {

masSt.push\_back(newStudent);

}

void Group::delStudent(Student oldStudent) {

masSt.remove(oldStudent);

}

void Group::GroupSort() {

masSt.sort();

}

void Group::GroupOut() {

iter = masSt.begin();

while (iter != masSt.end()) {

(\*iter++).display();

}

}

Student Group::findStudent(string searchName, string searchLastName) {

Student temp(searchName, searchLastName);

iter = find(masSt.begin(), masSt.end(), temp);

return (\*iter);

}

1. Для класса Student были переопределены операторы

bool operator== (const Student& p1, const Student& p2)

{

return (p1.name == p2.name && p1.last\_name == p2.last\_name) ? true : false;

}

bool operator< (const Student& p1, const Student& p2)

{

if (p1.last\_name == p2.last\_name)

return (p1.name < p2.name) ? true : false;

return (p1.last\_name < p2.last\_name) ? true : false;

}

bool operator!= (Student& p1, Student& p2)

{

return !(p1 == p2);

}

bool operator> (Student& p1, Student& p2)

{

return !(p1 < p2) && !(p2 == p2);

}

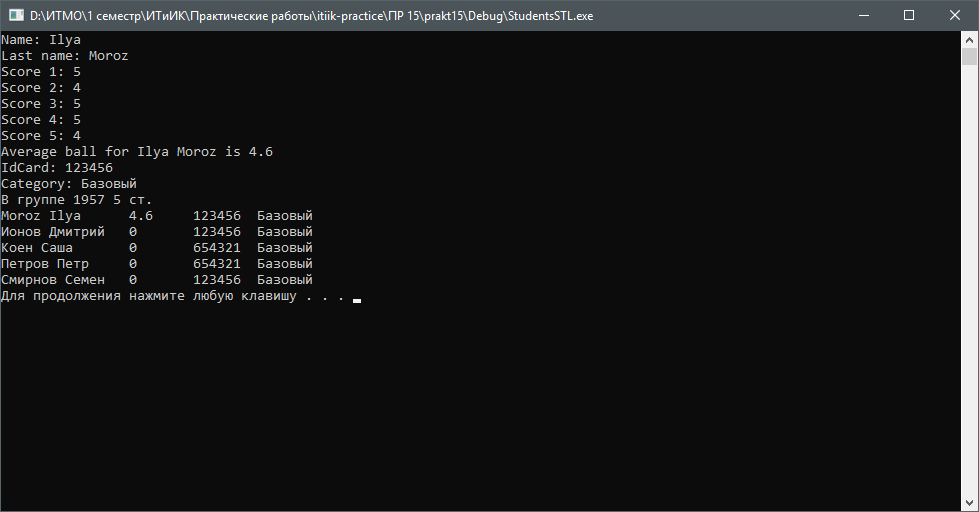


Рисунок 1 – проверка работы программы

1. Был реализован поиск студента по имени и фамилии и его удаление

gr1957.delStudent(gr1957.findStudent("Семен", "Смирнов"));

gr1957.GroupOut();

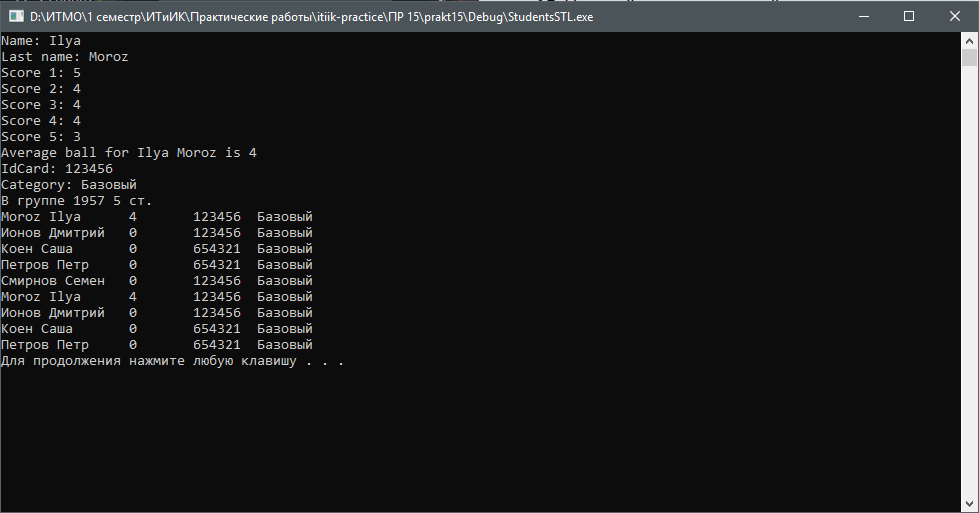


Рисунок 2 – вывод группы без одного студента

# Упражнение 2.

1. Класс Group был изменён так, чтобы список студентов хранился в мультимножестве

multiset<Student\*, compareStudent> masSt;

multiset<Student\*, compareStudent>::iterator iter;

1. Методы класса были изменены для работы с мультимножеством

int Group::getSize() {

return masSt.size();

}

void Group::addStudent(Student\* newStudent) {

masSt.insert(newStudent);

}

void Group::delStudent(Student\* oldStudent) {

masSt.erase(oldStudent);

}

void Group::GroupSort() {

}

void Group::GroupOut() {

iter = masSt.begin();

while (iter != masSt.end()) {

(\*iter++)->display();

}

}

Student\* Group::findStudent(string searchName, string searchLastName) {

Student \*temp = new Student(searchName, searchLastName);

iter = masSt.lower\_bound(temp);

delete temp;

return (\*iter);

}

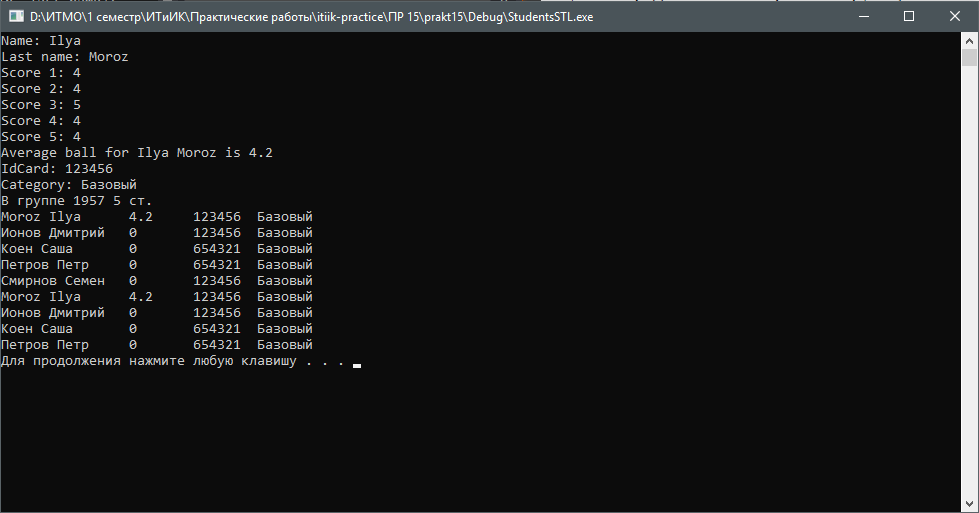


Рисунок 3 – результат работы программы

# Вывод

В данной практической работе были рассмотрены возможности хранения студентов в списках и мультимножествах. При этом использовалась передача как объектов, так и ссылок.

# Контрольные вопросы