МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»

(НГТУ)

Институт \_радиоэлектроники и информационных технологий\_\_\_\_\_\_\_

Кафедра «Прикладная математика и информатика»

Лабораторная работа №1

**Выполнил:**

Яковлев Иван Игоревич, 19-ПМ-1

**Проверил:**

Доцент кафедры «Прикладная математика»

Чернов Антов Григорьевич

Нижний Новгород

2020

Содержание:

Стр.

1. Постановка задачи…………………………………………………………………...3
2. Описание классов..............................................................................................4
3. Код………………………………………………......................................…………..5-17
4. Результат работы………………………………….......................................……...18
5. Вывод…………………………………………………………………………………....19

Постановка задачи

Создать на C++ справочную систему игроков с названием команды и набранных ими очки с реализацией классов и функциями удаления, поиска, добавления и редактирования.

Описание классов

Есть класс Players, он содержит имя игрока, его группу/команду его очки и указатель на следующего игрока . Также этот класс содержит геттеры и сеттеры для всех полей.

Класс List является односвязным списком, содержит указатель на первого игрока и размер списка. Содержит методы для добавления, удаления и редактирования игроков

Код

Players.hpp

#include <string>

using namespace std;

class Players{

private:

string \_name;

string \_group;

int \_ball;

Players\* \_next;

public:

Players();

void setBall(int ball);

void setName(string \_name);

void setGroup(string \_group);

void setNext(Players\* next);

void print();

Players\* next();

string name();

string group();

int ball();

friend ostream& operator<< (ostream &out, Players &player);

friend istream& operator>> (istream &in, Players &player);

friend ofstream& operator<< (ofstream &fout, Players &player);

friend ifstream& operator>> (ifstream &fin, Players &player);

};

Players.cpp

#include "Players.hpp"

#include <iostream>

#include <fstream>

using namespace std;

Players::Players(){

\_name = "";

\_group = "";

\_ball = 0;

\_next = NULL;

}

void Players::setBall(int ball){

\_ball = ball;

}

void Players::setName(string name){

\_name = name;

}

void Players::setGroup(string group){

\_group = group;

}

void Players::setNext(Players\* next){

\_next = next;

}

void Players::print(){

cout << "Name: " << \_name << endl;

cout << "Group: " << \_group << endl;

cout << "ball:" << \_ball << endl;

}

Players\* Players::next(){

return \_next;

}

string Players::name(){

return \_name;

}

string Players::group(){

return \_group;

}

int Players::ball(){

return \_ball;

}

ostream& operator<< (ostream &out, Players &player){

out << "Name: " << player.\_name << endl;

out << "Group: " << player.\_group << endl;

out << "ball:" << player.\_ball << endl;

return out;

}

istream& operator>> (istream &in, Players &player){

cout << "Name: ";

in >> player.\_name;

cout << "Group: ";

in >> player.\_group;

cout << "ball: ";

in >> player.\_ball;

return in;

}

ofstream& operator<< (ofstream &fout, Players &player){

fout << player.\_name << endl;

fout << player.\_group << endl;

fout << player.\_ball << endl;

return fout;

}

ifstream& operator>> (ifstream &fin, Players &player){

fin >> player.\_name;

fin >> player.\_group;

fin >> player.\_ball;

List.hpp

#include "Players.hpp"

class List{

private:

Players\* head;

int \_size;

public:

List();

~List();

void add(Players\* t);

void popHead();

void remove(int n);

void printAll();

void readFromFile(string filename);

void writeToFile(string filename);

void find(string name);

void find1(string group);

void find(string name, string group);

void find(int ball);

void changeName(int n, string name);

void changeGroup(int n, string group);

void changeBall(int n, int ball);

friend ostream& operator<< (ostream &out, List &list);

friend ofstream& operator<< (ofstream &fout, List &list);

friend ifstream& operator>> (ifstream &fin, List &list);

};

List.cpp

#include "List.hpp"

#include <fstream>

#include <iostream>

using namespace std;

List::List(){

head = NULL;

\_size = 0;

}

List::~List(){

while(head != NULL){

Players\* temp = head->next();

delete head;

head = temp;

}

}

void List::add(Players\* temp){

temp->setNext(head);

head = temp;

\_size++;

}

void List::popHead(){

Players\* temp = head->next();

delete head;

head = temp;

\_size--;

}

void List::remove(int n){

Players\* temp = head;

if(head == NULL || n > \_size)

return;

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

temp = temp->next();

Players\* del = temp->next();

temp->setNext(del->next());

delete del;

\_size--;

}

void List::printAll(){

Players\* temp = head;

for(int i = 0; i < \_size; i++){

cout << "Number of element: " << i << endl;

temp->print();

cout << endl;

temp = temp->next();

}

}

void List::readFromFile(string filename){

ifstream input;

input.open(filename);

if(!input.is\_open()){

cout << "Can't open this file" << endl;

return;

}

string s\_temp;

int i\_temp;

do{

Players\* temp = new Players;

input >> s\_temp;

temp->setName(s\_temp);

input >> s\_temp;

temp->setGroup(s\_temp);

input >> i\_temp;

temp->setBall(i\_temp);

this->add(temp);

} while(!input.eof());

input.close();

this->popHead();

}

void List::writeToFile(string filename){

ofstream output;

output.open(filename);

if(!output.is\_open()){

cout << "Can't open thise file" << endl;

return;

}

Players\* temp = head;

while(temp != NULL){

output << temp->name() << endl;

output << temp->group() << endl;

output << temp->ball() << endl;

temp = temp->next();

}

output.close();

}

void List::find(string name){

Players\* temp = head;

int num = 0;

while(temp != NULL){

if(temp->name() == name){

cout << "Number of element: " << num << endl;

temp->print();

}

temp = temp->next();

num++;

}

}

void List::find1(string group){

Players\* temp = head;

int num = 0;

while(temp != NULL){

if(temp->group() == group){

cout << "Number of element: " << num << endl;

temp->print();

}

temp = temp->next();

num++;

}

}

void List::find(string name, string group){

Players\* temp = head;

int num = 0;

while(temp != NULL){

if(temp->name() == name && temp->group() == group){

cout << "Number of element: " << num << endl;

temp->print();

}

temp = temp->next();

}

}

void List::find(int ball){

Players\* temp = head;

int num = 0;

while(temp != NULL){

if(temp->ball() == ball){

cout << "Number of element: " << num << endl;

temp->print();

cout << endl;

}

temp = temp->next();

num++;

}

}

void List::changeName(int n, string name){

Players\* temp = new Players;

temp = head;

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

temp = temp->next();

temp->setName(name);

}

void List::changeGroup(int n, string group){

Players\*temp = new Players;

temp = head;

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

temp = temp->next();

temp->setGroup(group);

}

void List::changeBall(int n, int ball){

Players\* temp = new Players;

temp = head;

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

temp = temp->next();

temp->setBall(ball);

}

ostream& operator<< (ostream &out, List &list){

Players\* temp = list.head;

for(int i = 0; i < list.\_size; i++){

out << "Number of element: " << i << endl;

out << \*temp;

out << endl;

temp = temp->next();

}

return out;

}

ofstream& operator<< (ofstream &fout, List &list){

Players\* temp = list.head;

while(temp != NULL){

fout << \*temp;

temp = temp->next();

}

return fout;

}

ifstream& operator>> (ifstream &fin, List &list){

do{

Players\* temp = new Players;

fin >> \*temp;

list.add(temp);

} while(!fin.eof());

list.popHead();

return fin;

}

main.cpp

#include "List.hpp"

#include <iostream>

#include <fstream>

using namespace std;

int main(int argc, char\* argv[]){

bool flag = true;

int action;

int n, num;

string s\_temp, s\_temp1;

int ball;

ofstream output;

if(argc != 2){

cout << "Incorrect number of arguments" << endl;

return -1;

}

List playList;

//studList.readFromFile(argv[1]);

ifstream input(argv[1]);

if(!input.is\_open()){

cout << "Can't open thise file" << endl;

return -1;

}

input >> playList;

input.close();

while(flag){

cout << "Choose what you want " << endl;

cout << "1. Add new player" << endl;

cout << "2. Remove player" << endl;

cout << "3. Change player's data" << endl;

cout << "4. Find a player" << endl;

cout << "5. Print the whole database" << endl;

cout << "6. Save the list to the file" << endl;

cout << "7. Exit" << endl;

cin >> action;

Players\* temp;

switch(action){

case 1:

temp = new Players;

cin >> \*temp;

playList.add(temp);

break;

case 2:

cout << "Enter player's number: ";

cin >> n;

if(n == 0)

playList.popHead();

else

playList.remove(n);

break;

case 3:

cout << "Enter player's number: ";

cin >> num;

cout << "What would you like to change?" << endl;

cout << "1. Name" << endl;

cout << "2. Group" << endl;

cout << "3. Ball" << endl;

cin >> n;

switch(n){

case 1:

cout << "Enter new name: ";

cin >> s\_temp;

playList.changeName(num, s\_temp);

break;

case 2:

cout << "Enter new group: ";

cin >> s\_temp;

playList.changeGroup(num, s\_temp);

break;

case 3:

cout << "Enter ball: ";

cin >> ball;

playList.changeBall(num, ball);

break;

}

break;

case 4:

cout << "Enter the way to find" << endl;

cout << "1. Name" << endl;

cout << "2. Group" << endl;

cout << "3. Name and group" << endl;

cout << "4. ball" << endl;

cin >> n;

switch(n){

case 1:

cout << "Enter name: ";

cin >> s\_temp;

playList.find(s\_temp);

break;

case 2:

cout << "Enter group: ";

cin >> s\_temp;

playList.find1(s\_temp);

break;

case 3:

cout << "Enter name: ";

cin >> s\_temp;

cout << "Enter group: ";

cin >> s\_temp1;

playList.find(s\_temp, s\_temp1);

break;

case 4:

cout << "What ball do you want to find?" << endl;

cin >> ball;

playList.find(ball);

break;

}

break;

case 5:

cout << playList;

break;

case 6:

output.open(argv[1]);

if(!output.is\_open()){

cout << "Can't open file" << endl;

return -1;

}

output << playList;

output.close();

break;

case 7:

flag = false;

break;

default:

cout << "Incorrect action" << endl;

break;

}

}

}

Makefile

SRC = Players.o List.o main.o

TARGET = lab3

all: $(TARGET)

main.o: main.cpp

g++ -Wall -g -c -o main.o main.cpp

List.o: List.cpp

g++ -Wall -g -c -o List.o List.cpp

Players.o: Players.cpp

g++ -Wall -g -c -o Players.o Players.cpp

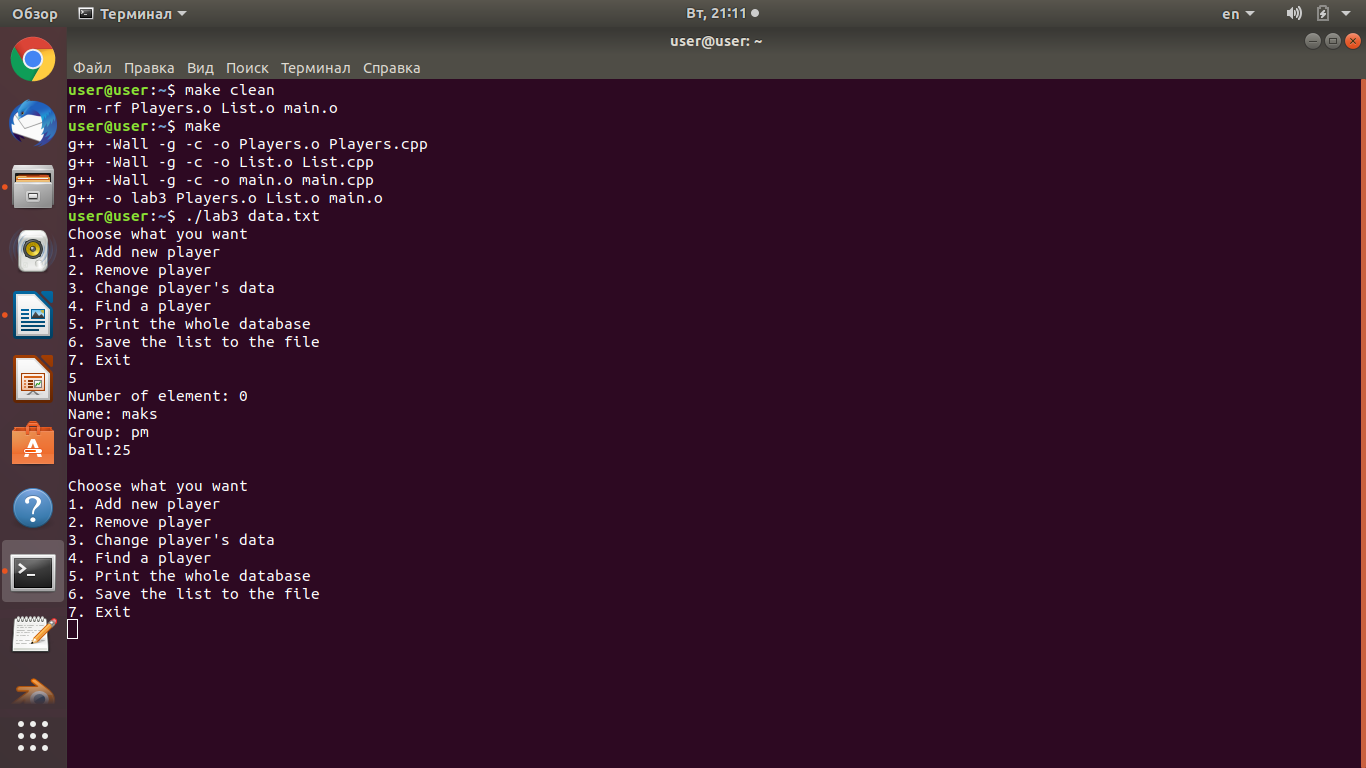
$(TARGET): $(SRC)

g++ -o $(TARGET) $(SRC)

clean:

rm -rf $(SRC)

Результат работы



Вывод

В результате выполнения практической работы, на языке С++ для ОС Linux была разработана

программа, позволяющая хранить и записывать справочную систему игроков с названием команды и набранных ими очки с реализацией классов и функциями удаления, поиска, добавления и редактирования.