**Министерство науки и высшего образования РФ**

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

«Чувашский Государственный Университет им. И.Н.Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

Отчет Лабораторной работы №10

По дисциплине: «Объектно-ориентированное программирование»

Вариант 14

Выполнил студент группы КТ-43-21

Казаков А.Ю.

Проверил ст. преподаватель:

Мытникова Е. А.

Чебоксары, 2022

**Цель.** Научиться программировать ввод и вывод в С++, используя объекты потоковых классов стандартной библиотеки С++.

MAIN.cpp

#include "Osadki.h"

string PATH = "Osadki.txt";

int AmountObjInFile();

void WriteFile(vector<Osadki>& vect);

void WriteFileToDelete(vector<Osadki>& vect);

vector<Osadki> ReadFile();

/\*

\* Программа #1

\* Написать программу #1 для создания объектов пользовательского класса

\* (ввод исходной информации с клавиатуры с использованием перегруженной операции “>>“)

\* и сохранения их в потоке (файле). Предусмотреть в программе вывод сообщения о

\* количестве сохраненных объектов и о длине полученного файла в байтах.

\*/

void CreateObjects()

{

cout << "////////////////////////////////////////////////////////////////////////////////////////////\n";

int size;

cout << "Введите количестов объектов : ";

cin >> size; cout << endl;

vector<Osadki> osadki(size);

// Заполняем вектор

int n = 0;

for (auto& i : osadki)

{

cout << "\tЭкземпляр номер - " << n + 1 << '\n';

cin >> i;

++n;

}

WriteFile(osadki);

cout << "\nКоличество сохраненных объектов = " << osadki.size() << endl;

cout << "Файл занимает : " << sizeof(osadki) << " байт\n";

cout << "\n////////////////////////////////////////////////////////////////////////////////////////////\n";

cout << endl;

}

/\*\*

\* Программа #2

\* Написать программу #2 для чтения объектов из потока,

\* сохранения их в массиве и просмотра массива. Для просмотра объектов использовать

\* перегруженную для cout операцию << и свой манипулятор.

\* Предусмотреть программе вывод сообщения о количестве прочитанных объектов и байтов.

\*/

void ReadObjects()

{

cout << "////////////////////////////////////////////////////////////////////////////////////////////\n";

vector<Osadki> osadky = ReadFile();

for (int i = 0; i < osadky.size(); i++)

{

cout << "\tЭкземпляр номер - " << i + 1 << '\n';

cout << osadky[i] << '\n';

}

cout << "\nКоличество объектов в файле : " << AmountObjInFile() << endl;

cout << "Файл занимает : " << sizeof(osadky) << " байт\n" << endl;

cout << "////////////////////////////////////////////////////////////////////////////////////////////\n";

}

/\*

\* Программа #3

\* Написать программу #3 для добавления объектов в поток.

\*/

void AddObject()

{

cout << "////////////////////////////////////////////////////////////////////////////////////////////\n";

cout << "Введите, какое количество обьектов добавить в поток : ";

int \_amountObj;

cin >> \_amountObj;

vector<Osadki> osadki = ReadFile();

ofstream output;

vector<Osadki> addOsadki(\_amountObj);

for (auto& i : addOsadki)

cin >> i;

WriteFile(osadki);

output.open(PATH, ios::app);

int n = 0;

output << endl;

for (auto i : addOsadki)

{

if (n != addOsadki.size() - 1)

output << i << endl;

else

output << i;

n++;

}

output.close();

cout << "\n////////////////////////////////////////////////////////////////////////////////////////////\n";

}

/\*\*

\* Программа #4

\* Написать программу #4 для удаления объектов из файла.

\*/

void DeteleObject()

{

ifstream input(PATH);

string \_findName;

cout << "Введите имя объекта, которое требуется удалить : "; cin >> \_findName;

string \_type\_osadki;

string \_time;

double \_day;

int iternum = 0;

bool isFind = false;

while (!input.eof())

{

input >> \_type\_osadki >> \_time >> \_day;

if (\_findName == \_type\_osadki)

{

isFind = true;

break;

}

++iternum;

}

if (!isFind)

{

cout << "Указанное имя не найдено!\n";

exit;

}

input.seekg(0, ios::beg);

vector<Osadki> osadki(AmountObjInFile() - 1);

int \_pos = 0;

int i = 0;

while (!input.eof())

{

++\_pos;

if (\_pos - 1 == iternum)

{

input >> \_type\_osadki >> \_time >> \_day;

continue;

}

input >> \_type\_osadki >> \_time >> \_day;

osadki[i].Set(\_type\_osadki, \_day, \_time);

++i;

}

WriteFile(osadki);

cout << "Номер объекта = " << iternum;

input.close();

}

/\*\*

\* Программа #5

\* Написать программу #5 для корректировки(т.е.замены) записей в файле.

\*/

void ChangeObject()

{

int \_deleteObjNum;

cout << "Введите номер объекта, которое требуется заменить : "; cin >> \_deleteObjNum;

if (\_deleteObjNum > AmountObjInFile())

{

cout << "Номер слишком большой. Out of range)";

exit;

}

string \_type\_osadki;

string \_time;

string \_day;

vector<Osadki> osadki = ReadFile();

string \_newName, \_newform, \_newsquare;

cout << "Новое название типа осадков - "; cin >> \_type\_osadki;

cout << "Новое время года - "; cin >> \_time;

cout << "Новый день - "; cin >> \_day;

int iternum = 0;

ofstream output(PATH);

for (int i = 0; i < osadki.size(); i++)

{

if (i == \_deleteObjNum && i == osadki.size() - 1)

{

output << \_type\_osadki << '\n';

output << \_time << '\n';

output << \_day;

continue;

}

if (i == \_deleteObjNum)

{

output << \_type\_osadki << '\n';

output << \_time << '\n';

output << \_day << '\n';

continue;

}

if (i == osadki.size() - 1)

{

output << osadki[i].GetTypeOsadki() << '\n';

}

else

{

output << osadki[i].GetTypeOsadki() << '\n';

}

}

output.close();

}

int main()

{

setlocale(LC\_ALL, "rus");

while (1)

{

char c = '=';

unsigned int \_action;

bool \_doExit = false;

cout << "\n\t\t\t\t\tЛабораторная работа - (10) Класс - Осадки\n\n";

cout << "\nСпособы взаимодействия : " << endl

<< "(1) Создать файл и заполнить его объектами" << endl

<< "(2) Ввести объекты из файла и вывести их" << endl

<< "(3) Добавить объект в поток (файл)" << endl

<< "(4) Удалить объект по имени" << endl

<< "(5) Изменить объект по номеру" << endl

<< "(0) Выйти" << endl

<< "\tВведите номер действия : "; cin >> \_action;

switch (\_action)

{

case 1 :

system("cls");

cout << endl;

CreateObjects();

cout << endl;

system("pause");

system("cls");

break;

case 2:

system("cls");

ReadObjects();

system("pause");

system("cls");

break;

case 3:

system("cls");

AddObject();

system("pause");

system("cls");

break;

case 4:

system("cls");

DeteleObject();

system("pause");

system("cls");

break;

case 5:

system("cls");

ChangeObject();

system("pause");

system("cls");

break;

case 0:

cout << "\n\n\n";

\_doExit = true;

break;

default:

system("cls");

cout << "Неправильно введен номер действия, введите снова" << endl;

system("pause");

system("cls");

}

if (\_doExit)

break;

}

return EXIT\_SUCCESS;

}

int AmountObjInFile()

{

ifstream inputf(PATH);

string \_str = "";

int sumElements = 0;

while (!inputf.eof())

{

inputf >> \_str;

sumElements++;

}

inputf.close();

return sumElements / 3;

}

void WriteFile(vector<Osadki>& vect)

{

ofstream output(PATH);

int n = 0;

for (int i = 0; i < vect.size(); ++i)

{

if (n != vect.size() - 1)

output << vect[i] << endl;

else

output << vect[i];

++n;

}

output.close();

}

void WriteFileToDelete(vector<Osadki>& vect)

{

int amountIterations = AmountObjInFile() - 1;

ofstream output(PATH);

int n = 0;

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

{

if (n != amountIterations)

output << vect[i] << endl;

else

output << vect[i];

++n;

}

output.close();

}

vector<Osadki> ReadFile()

{

int size = AmountObjInFile();

vector<Osadki> osadki(size);

ifstream input(PATH);

int i = 0;

while (!input.eof())

{

input >> osadki[i];

++i;

}

input.close();

return osadki;

}

ANIMAL.cpp

#include "Osadki.h"

Osadki::Osadki(){}

Osadki::Osadki(string Type\_osadki, double Day, string Time)

{

this->day = Day;

this->time = Time;

this->type\_osadki = Type\_osadki;

}

Osadki::Osadki(const Osadki& obj)

{

this->day = obj.day;

this->time = obj.time;

this->type\_osadki = obj.type\_osadki;

}

Osadki::~Osadki(){}

void Osadki::Set(string Type\_osadki, double Day, string Time)

{

this->day = Day;

this->time = Time;

this->type\_osadki = Type\_osadki;

}

void Osadki::SetTypeOsadki(string Type\_osadki)

{

this->type\_osadki = Type\_osadki;

}

void Osadki::SetTime(string Time)

{

this->time = Time;

}

void Osadki::SetDay(double Day)

{

this->day = Day;

}

string Osadki::GetTypeOsadki()

{

return type\_osadki;

}

string Osadki::GetTime()

{

return time;

}

double Osadki::GetDay()

{

return day;

}

void Osadki::print\_info()

{

cout << "Тип осадков - " << type\_osadki;

cout << "\nВремя - " << time;

cout << "\nДень - " << day << "\n\n";

}

istream& operator >> (istream& in, Osadki& obj)

{

cout << "type : "; in >> obj.type\_osadki;

cout << "time : "; in >> obj.time;

cout << "day : "; in >> obj.day;

return in;

}

ostream& operator << (ostream& out, const Osadki& obj)

{

out << "Type = " << obj.type\_osadki

<< "\ntime = " << obj.time

<< "\nday = " << obj.day << '\n';

return out;

}

ifstream& operator >> (ifstream& inf, Osadki& obj)

{

inf >> obj.type\_osadki;

inf >> obj.time;

inf >> obj.day;

return inf;

}

ofstream& operator<<(ofstream& outf, Osadki& obj)

{

outf << obj.type\_osadki << endl

<< obj.time << endl

<< obj.day;

return outf;

}

OSADKI.h

#pragma once

#ifndef ANIMAL\_H\_

#define ANIMAL\_H\_

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

#include "AnimalManipulator.h"

using namespace std;

class Osadki

{

string time;

string type\_osadki;

int day;

public:

Osadki();

Osadki(string Type\_osadki, double Day, string Time);

Osadki(const Osadki& osadki);

~Osadki();

void Set(string Type\_osadki, double Day, string Time);

void SetTypeOsadki(string type\_osadki);

void SetDay(double day);

void SetTime(string time);

string GetTypeOsadki();

double GetDay();

string GetTime();

void print\_info();

// Перегрузка операторов ввода, вывода, записи, чтения

friend istream& operator >> (istream& in, Osadki& obj);

friend ostream& operator << (ostream& out, const Osadki& obj);

friend ifstream& operator >> (ifstream& inf, Osadki& obj);

friend ofstream& operator << (ofstream& outf, Osadki& obj);

};

#endif // ANIMAL\_H\_



