Министерство науки и высшего образования РФ

Федеральное бюджетное государственное образовательное учреждение высшего профессионального образования

Пермский национальный исследовательский политехнический университет (ПНИПУ)

Факультет: Электротехнический (ЭТФ)

Направление: 09.03.01 «Информатика и вычислительная техника (Автоматизированные системы обработки информации и управления)»

Кафедра: «Информационные технологии и автоматизированных систем» (ИТАС)

Основы алгоритмизации

Лабораторная работа №10

Сохранение данных в файле с использованием потоков

Студент: Балтаев Э. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Группа: АСУ-20-1бзу

Работу проверил:

доцент кафедры ИТАС

Полякова О. А.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись:\_\_\_\_\_\_\_\_\_\_\_

Пермь 2021г.

Постановка задачи:

1. Создать пользовательский класс с минимальной функциональностью

2. Написать функцию для создания объектов пользовательского класса (ввод исходной информации с клавиатуры) и сохранения их в потоке (файле)

3. Написать функцию для чтения и просмотра объектов из потока

4. Написать функцию для удаления объектов из потока в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

5. Написать функцию для добавления объектов в поток соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

6. Написать функцию для изменения объектов в потоке в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

7. Для вызова функций в основной программе предусмотреть меню

Задание варианта:

Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой. Реализовать:

- деление сумм,

- умножение суммы на дробной число.

**Задание**

\* Удалить все записи не равные заданному значению

\* Уменьшить все записи с заданным значением в два раза

\* Добавить К записей в начало файла

Описание класса

1) Сумма

#pragma once

#include <iostream>

#include <fstream>

using namespace std;

class money

{

private:

long rub;

int cop;

public:

money();

money(long r, int c);

money(const money& m);

~money() {}

long get\_rub() { return rub; }

long get\_cop() { return cop; }

void set\_rub(long r);

void set\_cop(int c);

money& operator= (const money& m);

money operator/ (int val);

bool operator== (const money& m);

friend istream& operator>> (istream& in, money& m);

friend ostream& operator<< (ostream& out, const money& m);

friend ofstream& operator<< (ofstream& fout, const money& m);

friend ifstream& operator>> (ifstream& fin, money& m);

};

2) Потоки

#pragma once

#include "money.h"

void save();

void read();

void del();

void update();

void insert();

Определение компонентных функций

1) Сумма

#include "money.h"

money::money()

{

rub = 0;

cop = 0;

}

money::money(long r, int c)

{

set\_rub(r);

set\_cop(c);

}

money::money(const money& m)

{

rub = m.rub;

cop = m.cop;

}

void money::set\_rub(long r)

{

if (r < 0)

rub = 0;

else

rub = r;

}

void money::set\_cop(int c)

{

if (c < 0)

cop = 0;

else {

rub += c / 100;

cop = c % 100;

}

}

money& money::operator=(const money& m)

{

if (&m == this)

return \*this;

rub = m.rub;

cop = m.cop;

return \*this;

}

money money::operator/(int val)

{

money m(rub / val, cop / 2);

return m;

}

bool money::operator==(const money& m)

{

if (rub == m.rub && cop == m.cop)

return true;

return false;

}

istream& operator>>(istream& in, money& m)

{

long r;

int c;

cout << "Введите сумму через пробел: ";

in >> r; m.set\_rub(r);

in >> c; m.set\_cop(c);

return in;

}

ostream& operator<<(ostream& out, const money& m)

{

out << m.rub << "," << m.cop;

return out;

}

ofstream& operator<<(ofstream& fout, const money& m)

{

fout << m.rub << " " << m.cop;

return fout;

}

ifstream& operator>>(ifstream& fin, money& m)

{

fin >> m.rub >> m.cop;

return fin;

}

2) Потоки

#include "stream.h"

#pragma once

#include "money.h"

void save() {

ofstream out;

out.open("file1.txt");

if (!out.is\_open())

throw exception("Файл не открылся");

int n;

cout << "Количество сумм? "; cin >> n;

if (n < 1)

throw exception("Введите значенме больше 0");

money m;

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

cin >> m; // вводим объект

out << m << endl; // записываем в файл

}

out.close();

cout << "Успешно!" << endl;

}

void read() {

ifstream in;

in.open("file1.txt");

if (!in.is\_open())

throw exception("Файл не открылся");

money m;

int c = 0;

do {

in >> m; // читаем строку

if (in.eof())

break;

c++;

cout << m << endl; // выводим на экран

} while (true);

in.close();

if (c == 0)

cout << "Пусто" << endl;

}

void del() {

money key;

cin >> key;

ifstream in;

in.open("file1.txt");

if (!in.is\_open())

throw exception("Файл для чтения не открылся");

ofstream out;

out.open("file1\_temp.txt");

if (!out.is\_open())

throw exception("Файла на запись не открылся");

money m;

do {

in >> m;

if (in.eof())

break;

// если равны копируем

if (m == key)

out << m << endl;

} while (true);

in.close();

out.close();

remove("file1.txt");

rename("file1\_temp.txt", "file1.txt");

cout << "Успешно!" << endl;

}

void update() {

money key;

cin >> key;

ifstream in;

in.open("file1.txt");

if (!in.is\_open())

throw exception("Файл для чтения не открылся");

ofstream out;

out.open("file1\_temp.txt");

if (!out.is\_open())

throw exception("Файла на запись не открылся");

money m;

do {

in >> m;

if (in.eof())

break;

// если равны копируем

if (m == key)

m = m / 2;

out << m << endl;

} while (true);

in.close();

out.close();

remove("file1.txt");

rename("file1\_temp.txt", "file1.txt");

cout << "Успешно!" << endl;

}

void insert() {

ifstream in;

in.open("file1.txt");

if (!in.is\_open())

throw exception("Файл для чтения не открылся");

ofstream out;

out.open("file1\_temp.txt");

if (!out.is\_open())

throw exception("Файла на запись не открылся");

money m;

int K;

cout << "Количество сумм? "; cin >> K;

if (K < 1)

throw exception("Введите значенме больше 0");

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

cin >> m;

out << m << endl;

}

do {

in >> m;

if (in.eof())

break;

out << m << endl;

} while (true);

in.close();

out.close();

remove("file1.txt");

rename("file1\_temp.txt", "file1.txt");

cout << "Успешно!" << endl;

}

Определение главного файла

#include <iostream>

#include "stream.h"

int main()

{

system("chcp 1251");

int x;

while (true) {

try {

cout << "\n1. Сформировать файл";

cout << "\n2. Прочитать файл";

cout << "\n3. Удалить все записи не равные заданному значению";

cout << "\n4. Уменьшить все записи с заданным значением в два раза";

cout << "\n5. Добавить К записей в начало файла";

cout << "\n6. Очистить";

cout << "\n7. Выход";

cout << "\n>";

cin >> x;

switch (x) {

case 1:

save();

break;

case 2:

read();

break;

case 3:

del();

break;

case 4:

update();

break;

case 5:

insert();

break;

case 6:

system("cls");

break;

case 7:

return 0;

break;

}

}

catch (exception& ex) {

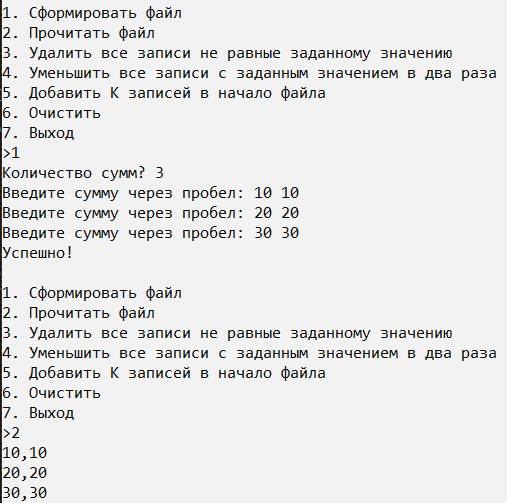
cout << "Ошибка: " << ex.what() << endl;

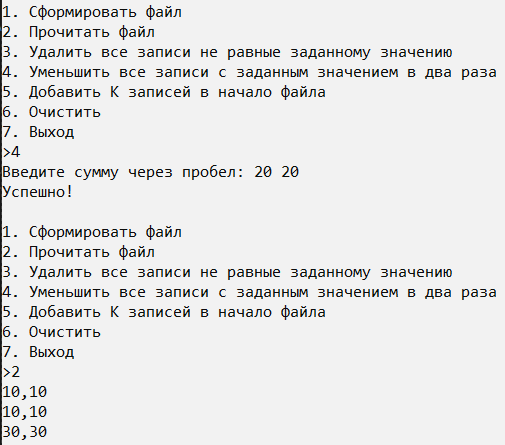
}

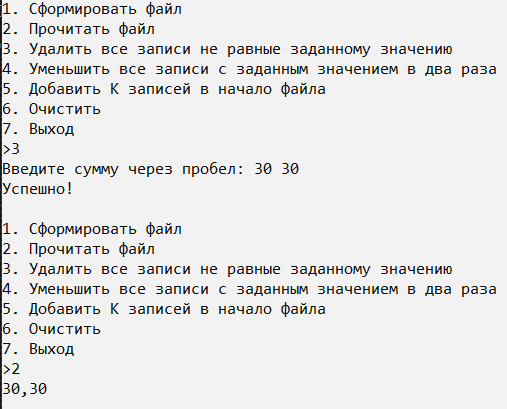
}

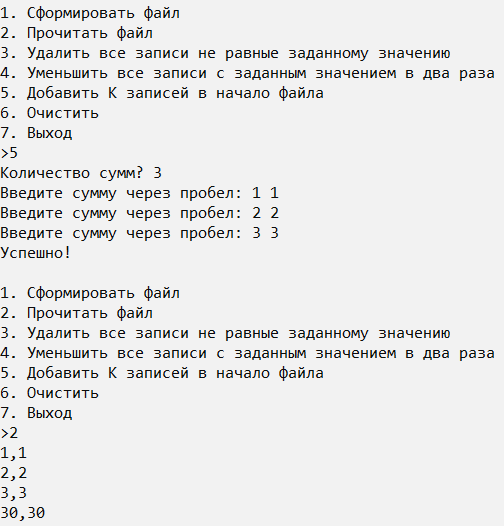
}

Результат работы программы









Ответы на контрольные вопросы:

1) Поток – определяется как последовательность байтов и не зависит от конкретного устройства, с которым производится обмен.

2) Стандартные, строковые, файловые

3) <iostream>

4) <fstream>

5) <sstream>

6) operator <<

7) operator >>

8) put(c); write(&c); write(s, strlen(s));

9) get(c); read(&c, 1); get(s, 50);

10) in, out, trunk, app, ate, binary

11) app

12) in

13) ate

14) out | trunk

15) Вывод и добавление в конец

16) Вывод и очищение файла

17) Вывод, ввод и очищение

18) ifstream s; s.open(path) / ifstream s(path);

19) ofstream s; s.open(path) / ofstream s(path);

20) in / out | trunk / out | binary / app …

21) char c; while(s.get(c)) { cout<<c; }

22) string s = ‘hello world’; fout << s << endl;

23)

Открыть файл1 на чтение

Открыть файл2 на запись

Читаем файл1 и проверяем нужно ли удалять

Если не нужно то записываем в файл2

Продолжаем

Закрываем оба файла

Удаляем файл1, переименовываем файл2

24)

Открыть файл1 на чтение в режиме «добавление в конец»

В цикле добавляем значения

Продолжаем

Закрываем файл1

25)

Открыть файл1 на чтение

Открыть файл2 на запись

Читаем файл1 и проверяем нужно ли изменить

Если нужно то меняем

Записываем в файл2

Продолжаем

Закрываем оба файла

Удаляем файл1, переименовываем файл2