Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЁТ**

**по дисциплине «Информатика»**

Семестр: 2

На тему: «Обработка исключительных ситуаций»

Выполнил студент ИВТ-22-2б:

Зайцев Павел Денисович

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

**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 в 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.

**Вариан 9:**

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:

— вычитание дробного числа из суммы

— операции сравнения (==,!=).

**Задание:**

* Удалить все записи равные заданному значению.
* Уменьшить все записи с заданным значением на 1 рубль 50 копеек. Значение

интервала не должно быть меньше 0 рублей 0 копеек.

* + Добавить К записей после элемента с заданным значением.

**Описание класса**

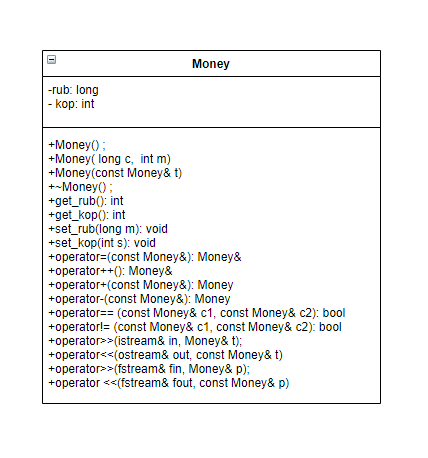
Класс Money.

Поля типа private: long rub – первое число длинное целого типа, Int b – второе число целого типа.

Методы типа public: 3 вида конструкторов и деструктор; геттеры и сеттеры для полей; перегрузка оператора присваивания operator=(const Money&); вычитания экземпляров класса operator-(const Money&); сложения для экземпляров класса operator+(const Money&); сложения для дробных чисел operator+(double);

перегруженный операторы сравнения экземпляров класса operator== (const Money& c1, const Money& c2), operator!= (const Money& c1, const Money& c2); перегруженные функции записи в файл operator>>(fstream& fin, Money& p), operator <<(fstream& fout, const Money& p), перегруженные функции потокового ввода-вывода operator>>(istream& in, Money& t), operator<<(ostream& out, const Money& t).

Также описание класса представлено на UML-диаграмме.



**Функции для работы с файлом**

int make\_file(const char\* f\_name)

{

fstream stream(f\_name, ios::out | ios::trunc); //открыть для записи

if (!stream)return -1;//ошибка открытия файла

int n;

Money p;

cout << "N?";

cin >> n;

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

{

cin >> p;//ввод атрибутов объекта из стандартного потока

stream << p << "\n";//запись объекта в файловый поток

}

stream.close();//закрыть поток

return n;//вернуть количество записанных объектов

}

int print\_file(const char\* f\_name)

{

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

Money p; int i = 0;

while (stream >> p)

{

cout << p << "\n";

i++;

}

stream.close();

return i;

}

int del\_fromfile(const char\* f\_name, Money number)

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i = 0;

Money p;

while (stream >> p)//пока нет конца файла выполняем чтение объекта

{

//если прочитан признак конца файла,то выход из цикла

if (stream.eof())break;

i++;

//если объект number не равен k, то записываем его во вспомогательый файл

if (p != number) {

temp << p;

}

else {

i--;

}

}

//закрыть файлы

stream.close();

temp.close();

remove(f\_name);//удалить старый файл

rename("temp", f\_name);// переименовать temp

return i;//количество прочитанных

}

int minus\_150(const char\* f\_name, Money number)

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i = 0;

Money p;

while (stream >> p)//пока нет конца файла выполняем чтение объекта

{

//если прочитан признак конца файла,то выход из цикла

if (stream.eof())break;

i++;

//если объект number равен p, то отнимаем от него 1ю50 руб и записываем его во вспомогательый файл

if (p == number)

{

p - 1.50;

temp << p;

}

else {

temp << p;

}

}

//закрыть файлы

stream.close();

temp.close();

remove(f\_name);//удалить старый файл

rename("temp", f\_name);// переименовать temp

return i;//количество прочитанных

}

int dobav(const char\* f\_name, int k)

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i = 0;

Money p;//Буферный экземпляр класса для считывания в него данных с файла

Money x;//Буферный экземпляр класса для записи новых элементов

int n=0;

do {

cout << "\nВведите колличество экземпляров которое вы хотите вставить:";

cin >> n;

} while (n <= 0);

while (stream >> p)//пока нет конца файла выполняем чтение объекта

{

//если прочитан признак конца файла,то выход из цикла

if (stream.eof())break;

i++;

//если объект number не равен k, то записываем его во вспомогательый файл

if (i == k) {

temp << p;

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

cout << endl<< c+1 << "-ый экземпляр";

cin >> x;//Инициализация нового экземпляра класса

temp << x;//Запись в файл нового экземпляра класса

i++;

}

}

else {

temp << p;

}

}

//закрыть файлы

stream.close();

temp.close();

remove(f\_name);//удалить старый файл

rename("temp", f\_name);// переименовать temp

return i;//количество прочитанных

}

**Функция main()**

void main()

{

setlocale(LC\_ALL,"rus");

Money p, buff;

int k, c,x;

char file\_name[30];

do

{

//Меню

cout << "\n1. Создать файл ";

cout << "\n2. Вывести файл";

cout << "\n3. Удалить элементы с файла ";

cout << "\n4. добавить К элементов";

cout << "\n5. Вычесть 1,50 руб из заданных элементов файла";

cout << "\n0. Выйти из программы\n";

cin >> c;

switch (c)

{

case 1: cout << "Имя файла?";

cin >> file\_name;

k = make\_file(file\_name);

if (k < 0)cout << "Ошибка";

break;

case 2: cout << "Имя файла?"; cin >> file\_name;

k = print\_file(file\_name);

if (k == 0)cout << "Файл пуст\n";

if (k < 0)cout << "Файла не существует\n";

break;

case 3: cout << "Имя файла?"; cin >> file\_name;

cout << "ВВЕДИТЕ ЧИСЛО денег ? ";

cin >> buff;

k = del\_fromfile(file\_name, buff);

if (k < 0)cout << "Файла не существует";

break;

case 4: cout << "Имя файла?";

cin >> file\_name;

do {

cout << "ВВЕДИТЕ номер элемента, после которого хотите добавить?";

cin >> k;

} while (k<=0);

k = dobav(file\_name, k);

if (k < 0) { cout << "Файла не существует"; break; };

if (k == 0) { cout << "Can't read file"; break; };

break;

case 5: cout << "Имя файла?";

cin >> file\_name;

cout << "ВВЕДИТЕ ЧИСЛО денег?";

cin >> buff;

k = minus\_150(file\_name, buff);

if (k < 0) cout << "\nCan't read file";

if (k == 0) cout << "\nNot such record";

break;

}

} while (c != 0);

}

**Вывод программы**

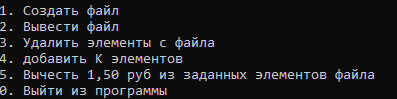


Рисунок 1 -Вывод меню программы

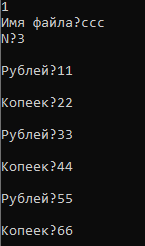


Рисунок 2 -Вывод 1 команды меню

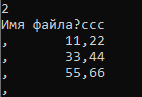


Рисунок 3 -Вывод 2 команды меню

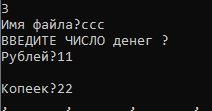


Рисунок 4 -Вывод 3 команды меню

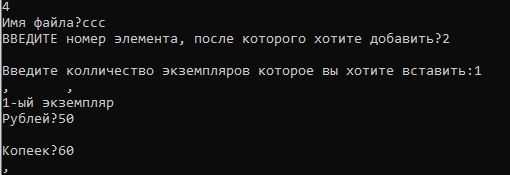


Рисунок 5 -Вывод 4 команды меню

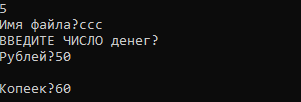


Рисунок 6 -Вывод 5 команды меню

**Контрольные вопросы**

1. *Что такое поток?*

Ответ: Поток - последовательность байтов, не зависящаяот конкретного устройства, с которым производится обмен

1. *Какие типы потоков существуют?*

Ответ: Стандартные, строковые и файловые. Все они делятся на однонаправленные и двунаправленные.

1. *Какую библиотеку надо подключить при использовании стандартных потоков?*

Ответ: <iostream>

1. *Какую библиотеку надо подключить при использовании файловых потоков?*

Ответ: <fstream>

1. *Какую библиотеку надо подключить при использовании строковых потоков?*

Ответ: <sstream>

1. *Какая операция используется при выводе в форматированный поток?*

Ответ: operator<<

1. *Какая операция используется при вводе из форматированных потоков?*

Ответ: operator>>

1. *Какие методы используются при выводе в форматированный поток?*

Ответ:



1. *Какие методы используется при вводе из форматированного потока?*

Ответ:





1. Какие режимы для открытия файловых потоков существуют?

Ответ:



1. *Какой режим используется для добавления записей в файл?*

Ответ: режим app

1. *Какой режим (комбинация режимов) используется в конструкторе*  
   ifstream file("f.txt")?

Ответ: in

1. *Какой режим (комбинация режимов) используется в конструкторе  
   fstream file("f.txt")?*

Ответ: in | out

1. *Какой режим (комбинация режимов) используется в конструкторе  
   ofstream file("f.txt")?*

Ответ: out

1. *Каким образом открывается поток в режиме ios::out|ios::app?*

Ответ: поток открывается на запись в конец файла.

1. *Каким образом открывается поток в режиме ios::out [ios::trunc?*

Ответ: поток открывается для записи, предварительно очистив файл

1. *Каким образом открывается поток в режиме ios::out ios::in/ios::trunk?*

Ответ: на ввод/ вывод данных с предварительной очисткой файла

1. *Каким образом можно открыть файл для чтения?*

Ответ: явно методом open и неявно при вызове конструктора

1. *Каким образом можно открыть файл для записи?*

Ответ: явно методом open и неявно при вызове конструктора

1. *Привести примеры открытия файловых потоков в различных режимах.*

Ответ: ofstream output(“file”); ifstream input(“file”); ofstream base(“file”, ios::trunc);

1. *Привести примеры чтения объектов из потока.*

Ответ: считывание строки getline(), считывание чисел, символов

«поток» » something,

1. *Привести примеры записи объектов в поток.*

Ответ: запись строки, чисел, символов

«поток» << something,

1. *Сформулировать алгоритм удаления записей из файла.*

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2.
4. Закрыть файлы.
5. Переименовать файл 2 в файл 1.
6. Сформулировать алгоритм добавления записей в файл.

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2.
4. За ними записать необходимые элементы
5. Закрыть файлы.
6. Переименовать файл 2 в файл 1.
7. Сформулировать алгоритм изменения записей в файле.

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2 c измененными данными.
4. Закрыть файлы.
5. Переименовать файл 2 в файл 1.