**Новосибирский государственный технический университет**

Лабораторная работа №3 по дисциплине программирование

Тема: Организация ввода-вывода в С++

|  |  |
| --- | --- |
| Группа: АВТ-907  Студент: Смоленский И.В. | Преподаватель: Булатов А. Д. |

Новосибирск, 2020

**Задание**:

Вариант 14.

Изучить работу потоков ввода-вывода и реализацию перегрузки потоков ввода-вывода на стандартные устройства и в файл для разработанных классов.

Для класса из лаб. работы №2 перегрузить операции ввода/вывода, позволяющие осуществлять ввод и вывод в удобной фоpме объектов классов:

* ввод с клавиатуры объекта и вывод на экран;
* запись объектов в текстовый файл и загрузка из текстового файла;
* запись объектов в двоичный файл и загрузка из двоичного файла.

Теоретические сведения:

**Потоковые классы в С++**

***Поток***— это абстрактное понятие, относящееся к любому переносу данных от источника к приемнику. По виду устройств, с которыми работает поток, можно разделить потоки на ***стан­дартные***, ***файловые* и *строковые***. Библиотека потоковых классов С++ построена на основе двух базовых классов: **ios** и **streambuf** .

Класс streambuf обеспечивает организацию и взаимосвязь буферов ввода-вывода, размещаемых в памяти, с физическими устройствами ввода-вывода. Методы и данные класса streambuf программист явно обычно не использует. Этот класс нужен другим классам библиотеки ввода-вывода. Он доступен и программисту для создания новых классов на основе уже существующих.

**Схема иерархии**

ios + sreambuf

istream ostream

iostream

istrstream ostrstream

ifstream ofstream

strstream

fstream

istream – класс входных потоков;

ostream – класс выходных потоков;

iostream – класс ввода-вывода;

istrstream – класс входных строковых потоков;

ifstream – класс входных файловых потоков и т.д.

Потоковые классы, их методы и данные становятся доступными в программе, если в неё включен нужный заголовочный файл.

iostream.h – для ios, ostream, istream.

strstream.h – для strstream, istrstream, ostrstream

fstream.h – для fstream, ifstream, ofstream

С помощью перегрузки операторов ввода и вывода можно заставить компилятор распознавать в операторах ввода-вывода один или несколько необходимых классов. Рассмотрим, как создаются потоковые классы.

**Перегрузка операторов вывода в поток**

Обычно потоки вывода поддерживают только стандартные типы данных. С помощью перегрузки оператора вывода в поток << можно заставить выражения выводить объекты собственных классов.

std::ostream& operator<<(std::ostream& os,const Bankomat &p)

{

os << "ID of ATM: " << p.id << "\nsum in ATM: " << p.sumBank << "\nMax sum of Get: " << p.sumMaxGet << endl;

return os;

}

Поток вывода перегружен в виде дружественной функции. Так как функция возвращает ссылку на osteram, то можно выводить несколько объектов в одном операторе вывода в поток.

cout << p1 << p2 << p3 << p4;

**Перегрузка операторов ввода из потока**

С помощью перегрузки оператора ввода из потока >> можно читать объекты класса.

std::istream& operator>>(std::istream& is, Bankomat& p)

{

is >> p.id >> p.sumBank >> p.sumMaxGet;

return is;

}

**Файловые потоки**

Файловые потоки представляют объектно-ориентированный способ чтения и записи информации в дисковые файлы. Библиотека файловых потоков разработана для работы только с текстовыми файлами. Однако можно применить ее и для чтения и записи двоичных значений, текстов, структур или любых других объектов.

Класс файловых потоков подключается заголовочными файлами fstream.h и iostream.h. Нужно учесть несколько моментов при использовании файловых потоков.

* все классы файловых потоков, за исключением filebuf, - производные от класса ios. Благодаря своей наследственности они могут использовать функции-члены, манипуляторы, флажки состояния и другие методы обработки потоков из файла iostream.h.
* для чтения данных из файлов используйте класс ifstream, производный от класса istream.
* для выходных файлов используйте класс ofstream, производный от класса ostream.
* для чтения и записи в один файл используйте класс fstream.
* класс filebuf предоставляет буферизованный сервис ввода-вывода для этих классов.

***Потоки текстовых файлов***

Чтобы создать новый текстовый файл, определите объект класса ofstream и передайте конструктору класса два аргумента: имя файла и значение режима открытия:

ofstream ofs("Newfile.txt", ios::out);

Если файл уже существует, то это приведет к перезаписыванию этого файла. Аргумент ios::out определяет режим доступа к файлу. Это константы перечислимого типа open\_mode. Можно задать сразу несколько опций путем объединения констант в выражениях операцией ИЛИ "|".

Константы open\_mode:

**app** - операция записи добавляет новую информацию в конец файла;

**ate** - при открытии разыскивает конец файла;

**binary** - открывает файл в двоичном формате;

**in** - открывает файл для чтения;

**nocreate** - если файл не существует, то новый не открывается;

**noreplace** -если файл уже существует, то не перезаписывается;

**out** - открывает файл для записи.

Вывод:   
  
Мной были перегружены операции ввода/вывода, позволяющие осуществлять ввод и вывод в удобной фоpме объектов классов: