МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ

ГОУ ВПО

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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

***Факультет автоматики и вычислительной техники***

**Лабораторная работа № 3  
по курсу «Программирование»**

**«Переопределение операций»**

**Вариант №2**

Студентов 2-го курса  
факультета АВТФ группы АБ-521  
Рудковского А.А.

Крошакова Е.В.

Проверил Бычков М.И.

Новосибирск-2016г

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

**Задание**:

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

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

Изменить демонстpационную пpогpамму. В файл иметь возможность сохранять любое количество созданных объектов. При загрузке в программу создавать объекты с параметрами, записанными в файл.

**Ход работы**:

1. Для начала перегрузим два оператора, через функции с ключевыми словами operator << и operator>>.
2. Возвращаемое значение и левый операнд, зависит от выполняемой оператором функции.
3. При выполнении операции поток данных передается аргументу функции Далее в функции происходит вывод необходимых данных в этот поток. Затем этот поток передается в возвращаемое значение метода. Это необходимо для реализации последовательного вывода объектов в одном операторе.
4. Для реализации перегрузки операторов, мы использовали четыре вида потоков: ostream, istream, ofstream, ifstream.
5. Потоки ostream и istream мы использовали для ввода и вывода объектов с консоли.
6. Потоки ofstream и ifstream мы использовали для ввода и вывода объектов в файл и из файла.
7. Для работы с файлом, нам надо добавить два дополнительных метода, так как нужно создать файловый поток, произвести с ним необходимые действия, а затем закрыть его.
8. В методах будем использовать перегруженные нами операторы ввода/вывода в файл.
9. Для демонстрации работы программы, напишем тестовые методы.

**Вывод:**

Изучили работу потоков для записи и чтения информации с помощью потоков. Работа с потоками является более универсальным средством для ввода/вывода информации, а так же имеет более удобный и читаемый синтаксис. Используя потоки , мы научились записывать и читать информацию из файла, а так же работать с консолью.

Приложение 1. Заголовочный файл string.h

## Защита лабораторной работы. Вопросы к защите лабораторной работы

1. Что такое поток ввода-вывода?
2. Что такое файл? В каких форматах сохраняется информация в файлах?
3. Зачем нужно переопределять операции ввода/вывода >> и << для объектов классов?
4. Какие потоки используются при работе с файлами? Где они определены?
5. С какими атрибутами открывается файл?
6. Как открыть файл по чтению, по записи, по чтению и записи?
7. Как прочитать информацию из файла?
8. Как записать информацию в файл?
9. Как закрыть файл?
10. Как переместить внутренний указатель в файле?

Ответы:

1. ***Поток ввода-вывода***— это абстрактное понятие, относящееся к любому переносу данных от источника к приемнику
2. **Файл** — именованная область данных на носителе информации. Информации сохраняется в текстовом и двоичном формате.
3. Обычно потоки вывода поддерживают только стандартные типы данных. С помощью перегрузки оператора вывода в поток << можно заставить выражения выводить объекты собственных классов. С помощью перегрузки оператора ввода из потока >> можно читать объекты класса
4. Для работы с файлом, мы используем потоки ifstream и ofstream. Они определены в библиотеке iostream.
5. Атрибуты открытия файла:

**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;

1. Для открытия файла есть два варианта: в первом мы используем двусторонний поток fstream и указываем для него атрибуты: **in** - для чтения , **out** – для записи. **in|out** для записи и чтения. Второй вариант использовать потоки ifstream для записи, ofstream для чтения.
2. Чтобы прочитать информацию из файла используем перегруженный оператор >> или функцию потока getline.
3. Чтобы записать информацию из файла используем перегруженный оператор << или функция потока write.
4. Для закрытия файла мы используем функцию close в потоке.
5. Для перемещения указателя внутри файла служат функции seekp() и seekg(), которые позволяют переходить на произвольную позицию в файле. Для файла, открытого для чтения, используется функция seekg(), для файла, открытого для записи – seekp().

.