

«Санкт-Петербургский государственный политехнический университет»

Институт информационных технологий и управления

Кафедра «Распределенные вычисления и компьютерные сети»

КУРСОВАЯ РАБОТА

Создание простейших классов на языке C++

По дисциплине «Основы программирования»

Выполнил студент
гр. 13507/2

С.С. Недобуга

Руководитель
Ст.преподаватель

Т.Н. Самочадина

Санкт-Петербург

2014

Содержание

1 Введение	2
2 Постановка задачи	3
3 Описание разрабатываемых классов. Обоснование способа представления данных	4
3.1 Класс Stringl	4
3.2 Класс Base	4
4 Структура проекта	5
5 Особенности реализации	5
6 Тестирование	6
7 Заключение	7
8 Список используемой литературы	8
9 Список приложений	8

1. Введение

Объектно-ориентированное программирование - это результат эволюции более ранних методологий программирования. Тенденция увеличения количества и усложнение приложений привела программы в ненадежное состояние, что является следствием недостатка способностей языков программирования. Этим и обусловлена потребность в ООП.

ООП – это иерархически связанная модель данных, выраженная классами. Переход от традиционного программирования к ООП на начальном этапе характерен тем, что под объектами в программе подразумеваются конкретные физические объекты. В этом случае легче дается понимание различных действий над ними. В качестве примера: можно выбрать простые графические фигуры, типа квадрата, поскольку каждая фигура представляет реальный объект на экране. Впоследствии превратим его в строку ввода, например. Его всегда можно отобразить и, тем самым, проверить моделируемые действия над объектом в явном виде. После того, как определены простейшие графические объекты, достаточно легко можно формировать более сложные на основе уже имеющихся. ООП же предоставляет готовые сложные графические объекты, с соответствующим списком примитивных объектов. [1]

В данной курсовой работе показан пример создания классов с методами работы и операциями над объектами созданных классов.

2. Постановка задачи

Требуется написать программу для хранения и обработки информации о продуктах в магазинах. В ходе написания программы должен быть описан класс, содержащий следующие поля:

- название продукта
- магазин
- цена

Работа так же должна содержать:

- реализацию классов, включающую:
 - конструкторы
 - деструктор
 - перегрузку операторов (ввода\вывода, бинарных, унарных, операторов отношения и т.д.).

3. Описание разрабатываемых классов. Обоснование способа представления данных

3.1. Класс stringl

Для удобства хранения и обработки данных был создан класс stringl – упрощенная (не полная) версия класса string. (string little – stringl);

Данный класс хранит в себе массив типа char. Данные можно склеивать, резать, удалять, реализовано это посредством различных методов и перегрузок операторов, таких как сложения и т.п.

Приватные поля класса

- 1) data_ типа char* – для хранения указателей на нулевой элемент
- 2) length_ типа int – для хранения длины «слова»

Конструкторы: обычный, принимающий указатель на массив; конструктор копий; пустой; деструктор.

Методы и операторы:

- char * getch();
- friend istream& operator >> (istream &is, stringl &s);
- friend ostream& operator << (ostream &os, stringl &s);
- stringl operator += (const stringl &a);
- stringl operator = (const stringl &s);

3.2. Класс Base

Данный класс является реализацией поставленной задачи. В нем хранятся закрытые поля:

- Name_ типа stringl – имя продукта
- Shop_ типа stringl – название магазина
- Price_ типа stringl – цена товара

Класс Base имеет:

- конструкторы с параметрами (задается имя, магазин, цена),
- без параметров (все равно нулю),
- конструктор копирования,
- методы и перегруженные операторы:

```
stringl getName() const;
stringl getShop() const;
double getPrice() const;
void setName(const stringl &N);
void setShop(const stringl &N);
void setPrice(const double &N);
friend ostream& operator <<(ostream & os, Base &B);
friend istream& operator >>(istream & is, Base &B);
Base operator + (const double& d);
Base operator += (const Base& d);
Base operator - (const double& d);
Base operator -= (const Base& d);
```

```
Base operator ++ ();  
Base operator -- ();  
bool operator > (const Base& B);  
bool operator < (const Base& B);  
bool operator >= (const Base& B);  
bool operator <= (const Base& B);  
bool operator == (const Base& B);
```

4. Структура проекта

Каждый класс описывается в отдельном заголовочном файле: «Base.h» и «stringl.h».

Реализация соответствующих классам методов находятся в одноименных файлах расширения cpp.

Манипулирование классами и программой происходит в «main.cpp»

5. Особенности реализации

Реализованные в данной работе методы тривиальны и не требуют дополнительной документации.

Класс stringl – является классом-помощником (необязательным).

Обработаны исключения при вводе и выводе данных.

Возможно считать базу данных из файла input.txt по средством функции ReadFile.

Возможность вывести базу, представленную массивом класса Base, с помощью функции ShowBase.

6. Тестирование

Считываем данные из файла `..\Kurs1\Files\input.txt` используя побочную функцию `readFiles`. Первый параметр данной функции принимает указатель на массив типа `Base`, вторым параметром задается количество продуктов (считанных строк).

```
readFiles(B,4)
```

Как можно заметить из вызова функции, должно считаться 4 строки кода.

Выведем считанную базу на экран, с помощью функции `showBase(B,4)`; Второй аргумент – количество строк.

Получим:

```
*****
*   Product|      Shop|      Price*
*****
*   Grechka|   Slavyanka|    123.45*
*****
* Cheeseburger| McDonalds|    35.5 *
*****
*   Makarohi|   Karleone|     77*
*****
*    Kamen|    Osina|    300*
*****
```

Так же, для примера, сравним стоимость первого и второго элемента: гречки из Славянки и чизбургера из МакДональдса.

Как видно из таблицы, и фрагмента кода мы должны будем увидеть на экране данные о первом товаре – что, в конечном счете, и имеем:

```
Grechka|   Slavyanka|    123.45
```

Вывод: программа работает стабильно, правильно обрабатывает и выводит данные. Также работают перегруженные операторы.

Дополнительно: внеся код в обработчик ошибок, и например, заставить программу делать «невозможное» (считать 5 строк, из 4-ех строк базы), увидим:

Error. Unexpected end of file.

Input.txt

```
Grechka Slavyanka 123.45
Cheeseburger McDonalds 35.5
Makarohi Karleone 77.00
Kamen Osina 300.00
```

Фрагмент кода:

```
if(B[0]>B[1])
    cout<<B[0];
    else
    cout<<B[1];
```

7. Заключение

В данной курсовой работе применяется изученный материал об объектно-ориентированном программировании (ООП). В данной курсовой работе применяется изученный материал об объектно-ориентированном программировании (ООП).

ООП дает возможность заниматься разработкой, не следя за всеми нюансами. Достаточно просто знать чего вам нужно и представить это абстрактным объектом. При этом отпадает надобность «удерживать» все отдельные детали под контролем.

Итогом данной работы является реализованная база данных продуктов, содержащее, конечно, название, магазин, и цену продукта, а также элементы управления ей.

8. Список используемой литературы

1. Лаворе Р., Объектно-ориентированное программирование в С++. СПб.: Питер, 2011. – 928 с.: ил.
2. Павловская Т.А., С\С++. Программирование на языке высокого уровня. СПб.: Питер, 2007 – 464с.

9. Список приложений

1. Приложение 1. Текст программы.
2. Приложение 2. Презентация.