Обобщенные сведения ООП

Объекты и классы

Объектно-ориентированная программа строится как набор взаимодействующих **объектов**, а не функций. Функции, естественно, также присутствуют, но как соответствующие компоненты объектов. Объекты же являются основными структурообразующими сегментами (частями) программы.

Объекты содержат:

- - **функции,** необходимые для выполнения поступающих объекту команд,
- - **данные,** требующиеся для этих функций.

Каждый объект имеет тип, являясь экземпляром определенного класса, – термин «класс» по сути является синонимом термина «тип».

Класс – это составной структурированный тип, введенный программистом-пользователем, включающий в себя данные и функции, а также набор ограничений на доступ к этим данным и функциям. Класс описывает общие свойства (данные и функции) порождаемых объектов и предоставляет своего рода схему (шаблон) и механизм для создания объектов класса.

Объединение в классе данных и функций, работающих с этими данными в сочетании с ограничениями доступа называется *инкапсуляцией*. Данные и функции называют компонентами класса или членами класса.

Говорят, что данные задают состояние объекта, а функции – его поведение.

Для компонентов класса используются и другие термины: для данных – поля или атрибуты, для функций – методы, функции-члены, функции-элементы; в современной терминологии чаще применяются «поля» и «методы».

В простейшем случае синтаксис определения класса следующий:

class *Имя_класса{список_членов_класса};*

где

- ***Имя_класса*** – произвольно выбираемый идентификатор;
- ***список_членов_класса*** – объявления и определения полей (типизированных данных) и принадлежащих классу методов (функций-элементов).

Класс позволяет ограничивать доступ к своим внутренним компонентам, представляя их как открытые (**public**), закрытые (**private**) и защищенные (**protected**).

Класс устанавливает четко определенный интерфейс для взаимодействия объектов этого типа с остальными сегментами программы. Закрытые коды или данные доступны только внутри этого объекта, т.е. методы класса имеют неограниченный доступ к полям класса независимо от того, закрыты поля или нет.

Для объявления объекта класса (экземпляра класса) используется конструкция

Имя_класса имя_объекта;

Например, пусть мы объявили класс CMoney:

```
class CMoney
{ /*... */ };
```

Тогда мы можем объявлять такие переменные:

```
CMoney    t;       // переменная
CMoney *p;         // указатель
CMoney m[100];     // массив
```

```
1)   struct date                          // дата
{
int month,day,year; }; // поля: месяц, день, год
void set(int,int,int);          // метод – установить дату
void get(int*,int*,int*);    // метод – получить дату
void next();                    //метод – установить следующую дату
void print();                   //метод – вывести дату
```

```
2) class complex                    // комплексное число
{
double re,im;
public:
double real()     {return(re);}
double imag()    {return(im);}
void set(double x,double y)      {re = x; im = y;}
void print()   {cout << “re = “ << re; cout << “im = “ << im;}
};
```

Объявление объектов:

```
date today, my_birthday;
date *point = &today;        // указатель на объект типа date
date clim[30];               // массив объектов
date &name = my_birthday
```

Обращение к полям объекта

1) **С помощью “квалифицированных” имен:**

имя_объекта. имя_данного

имя_объекта. имя_функции

Например:

```
complex x1,x2;
x1.re = 1.24;
x1.im = 2.3;
x2.set(5.1,1.7);
x1.print();
```

2) Второй способ доступа использует указатель на объект

указатель_на_объект–>имя_компонента

```
complex *point = &x1;  //  или  point = new complex;
point –>re = 1.24;
point –>im = 2.3;
point –>print();
```

Лабораторная работа №6

Целью лабораторной работы является получение практических навыков при использовании простых классов и объектов в языке C++.

Разработать алгоритм и программу, выполняющую следующие операции.

1) Создать класс, описывающий указанную в задании сущность и включающий не менее 4 ÷ 5 полей, в том числе наименование объекта.

2) Разработать набор 4 ÷ 5 методов, которые служат для обработки данных – обязательно включить в набор методы для ввода значений с клавиатуры, вывода на экран.

3) Создать в программе несколько переменных (объектов) с типом указанного класса и продемонстрировать работу методов класса.

ЗАДАНИЕ

Вариант	***Класс***
0	Автомобиль (модель, бензобак, передвижение, . . .)

В классе CL_car (Автомобиль) предусмотрим 5 полей:

string model - название модели,

double rasst - пройденное расстояние,

double benzobak – объем бензобака,

double rashod - расход бензина,

double speed - средняя скорость передвижения;

```
  include <iostream.h>
#include <string.h>
#define LINE cout<<"--------------------------------\n"
using namespace std;

char str2[60];     /* для вывода кириллицы */
char* cyr(char* str1) {CharToOem(str1,str2); return str2;}

class CL_car {
string model;
double rasst, benzobak, rashod, speed;
public:
CL_car(string model) {// Constructor
benzobak=100.0; rasst = 0.0; rashod = 15.0; speed=65.0;
cout << " Constructor " << this << model << endl;
}
```

```
~CL_car()
{cout << " Destructor " << this << model << endl;} // Destructor
// Присвоить наименование
void SetName(string name)
{model = name;}                                  //  Ввести наименование с клавиатуры
void InpName()
{cout << cyr(" Введите марку (для объекта ")<<this<< "): ";
cin >> model;}
void GetName()
{ cout << model << endl;} //  Вывести марку автомбиля
//  Заправка - добавить бензин – прототип, определение далее вне класса
void Zaprav(double benzin);
//  Проехать - "пробег" – здесь прототип, определение далее вне класса
void Motion(double probeg);
//  Вывести состояниевoid Fields(){
cout << rasst << "  " << benzobak << "   " << speed << "  " << model <<
endl;
printf("%5.1f %6.1f %5.1f    ",rasst,benzobak,speed); cout<<model<<endl;
}};
```

```
int main() {
system("color f0");  // установить черный цвет
 CL_car car1(" c1 "), car2(" c2 "), car3(" c3 ");
car1.SetName(cyr("ПОБЕДА"));
car2.SetName(cyr("Москвич"));
car3.InpName();
car1.Motion(160); car1.Fields();
car2.Motion(110); car2.Fields();
car3.Motion(200); car3.Fields();
```

```
while (otvet) {              // цикл с выводом меню для отладки
cout << cyr(" 1. Пробег\n") ;
cout << cyr(" 2. Добавить бензин\n");
cout << cyr(" 3. Вывести состояние\n");
cout << cyr(" 4. Очистить экран\n");
cout << cyr(" 5. Выход\n");
cout << " > ";

cin >> kodm;                         // ввод кода меню
switch(kodm) {
case 1:                      // Пробег
car1.Motion(555);  break;
case 2:                      // Добавить бензин
car1.Zaprav(77);  break;
case 3:                      // Вывести состояние
LINE;cout<<cyr("Пробег Запас Скорость Модель  \n");LINE;
car1.Fields();
car2.Fields();
car3.Fields();
LINE; break;
case 4: system("cls"); break;
case 5: otvet = 0;  break;
default: cout << cyr("Неверный код меню\n"); break;
}}
```

```
//    Завершение работы – и посмотрим вызовы деструкторов ...
car1.GetName();
car2.GetName();
car3.GetName();
return 0;
}

//  Проехать - "пробег"  - Определение метода вне класса
void CL_car::Motion(double probeg){
benzobak -= probeg/rashod;
if (benzobak <= 35.)
cout << model <<cyr(" В баке мало бензина: ") << benzobak  <<  endl;
rasst += probeg;
}
//  Заправка - добавить бензин - Определение метода вне класса
void CL_car::Zaprav(double benzin){
benzobak += benzin;
}
```

```
"C:\Program Files\Microsoft Visual Studio\MyProjec
 Constructor 0012FF44 c1
 Constructor 0012FF14 c2
 Constructor 0012FEE4 c3
 Введите марку: ВОЛГА
160.0    89.3   65.0     ПОБЕДА
110.0    92.7   65.0     Москвич
200.0    86.7   65.0     ВОЛГА
 1. Пробег
 2. Добавить бензин
 3. Вывести состояние
 4. Очистить экран
 5. Выход
 > 3
-----------------------------------
Пробег  Запас  Скорость  Модель
-----------------------------------
160.0    89.3   65.0     ПОБЕДА
110.0    92.7   65.0     Москвич
200.0    86.7   65.0     ВОЛГА
-----------------------------------
 1. Пробег
 2. Добавить бензин
 3. Вывести состояние
 4. Очистить экран
 5. Выход
 > 1
 1. Пробег
 2. Добавить бензин
 3. Вывести состояние
 4. Очистить экран
 5. Выход
 > 3
```

```
-----------------------------------
Пробег  Запас  Скорость  Модель
-----------------------------------
715.0    52.3   65.0     ПОБЕДА
110.0    92.7   65.0     Москвич
200.0    86.7   65.0     ВОЛГА
-----------------------------------
 1. Пробег
 2. Добавить бензин
 3. Вывести состояние
 4. Очистить экран
 5. Выход
 > 5
ПОБЕДА
Москвич
ВОЛГА
 Destructor 0012FEE4ВОЛГА
 Destructor 0012FF14Москвич
 Destructor 0012FF44ПОБЕДА
Press any key to continue
```