**Лабораторна робота №5**

***Ієрархія об'єктів і групи. Агрегація.Композиція.***

**Мета.** Одержати практичні навички створення об'єктів-груп (агрегація, композиція) .

**Постановка задачі:**

**Завдання**

*Створити класси на яких можливо детально переглянути*

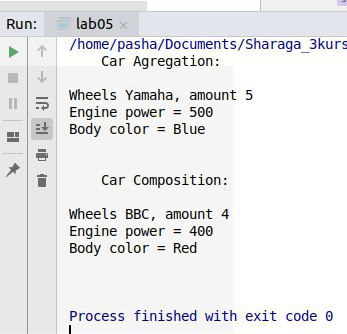
*механізм Агрегації і Композиції*

**Код пограми С++:**

#include **<bits/stdc++.h>  
  
using namespace** std;  
  
**class** Wheels  
{  
 **public**:  
 **int** amount;  
 **virtual void** Info() = 0;  
};  
  
**class** Yamaha : **public** Wheels  
{  
 **private**:  
 **int** amount;  
   
 **public**:  
 Yamaha(**int** amount)  
 {  
 **this**->amount = amount;  
 }  
  
 **void** Info()  
 {  
 cout << **"Wheels Yamaha, amount "** << amount << endl;  
 }  
  
};  
**class** BBC : **public** Wheels  
{  
 **private**:  
 **int** amount;  
   
 **public**:  
 BBC(**int** amount)  
 {  
 **this**->amount = amount;  
 }  
   
 **void** Info()  
 {  
 cout << **"Wheels BBC, amount "** << amount << endl;  
 }  
};

**class** Engine  
{  
 **private**:  
 **int** power;  
   
 **public**: Engine(**int** power)  
 {  
 **this**->power = power;  
 }  
 **void** Info()  
 {  
 cout << **"Engine power = "** << power << endl;  
 }  
};  
  
**class** Body  
{  
 **private**:  
 string color;  
  
 **public**:  
 Body(string color)  
 {  
 **this**->color = color;  
 }  
  
  
 **void** Info()  
 {  
 cout << **"Body color = "** << color << endl;  
 }  
};  
  
**class** Car\_Agregation  
{  
**private**:  
 Wheels \*wheels;  
 Engine \*engine;  
 Body \*body;  
  
**public**:  
 Car\_Agregation(Wheels \*wheels, Engine \*engine, Body \*body) {  
 **this**->wheels = wheels;  
 **this**->engine = engine;  
 **this**->body = body;  
 }  
  
 **void** Info()  
 {  
 cout << **"\tCar Agregation:\n"** << endl;  
 wheels->Info();  
 engine->Info();  
 body->Info();  
 cout << endl << endl;  
 }  
};  
  
**class** Car\_Composition  
{  
**private**:  
 Wheels \*wheels;  
 Engine \*engine;  
 Body \*body;  
  
**public**:  
 Car\_Composition()  
 {  
 wheels = **new** BBC(4);  
 *//wheels = new Yamaha(7);* engine = **new** Engine(400);  
 body = **new** Body(**"Red"**);  
 }  
  
 **void** Info()  
 {  
 cout << **"\tCar Composition:\n"** << endl;  
 wheels->Info();  
 engine->Info();  
 body->Info();  
 cout << endl << endl; }  
};  
  
  
  
  
**int** main() {  
  
 Wheels \*w = **new** Yamaha(5);  
 *//Wheels \*w2 = new BBC(7);* Engine \*e = **new** Engine(500);  
 Body \*b = **new** Body(**"Blue"**);  
  
 Car\_Agregation \*carAgregation = **new** Car\_Agregation(w, e, b);  
 carAgregation->Info();  
  
 Car\_Composition \*carComposition = **new** Car\_Composition();  
 carComposition->Info();  
 **return** 0;  
}

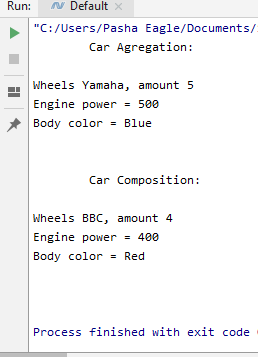
**Результат:**



**Код пограми С#:**

using System;  
  
namespace lab05  
{  
 internal class Program  
 {  
 abstract class Wheels  
 {  
 protected int amount;  
 public abstract void Info();  
 }  
  
 class Yamaha : Wheels  
 {  
 private int amount;  
  
 public Yamaha(int amount)  
 {  
 this.amount = amount;  
 }  
   
 public override void Info()  
 {  
 Console.WriteLine("Wheels Yamaha, amount " + amount);  
 }  
   
 }  
   
 class BBC : Wheels  
 {  
 private int amount;  
  
 public BBC(int amount)  
 {  
 this.amount = amount;  
 }  
  
 public override void Info()  
 {  
 Console.WriteLine("Wheels BBC, amount " + amount);  
 }  
 }  
  
 class Engine  
 {  
 private int power;  
  
 public Engine(int power)  
 {  
 this.power = power;  
 }  
   
 public void Info()  
 {  
 Console.WriteLine("Engine power = " + power);  
 }  
 }  
  
 class Body  
 {  
 private string color;  
  
 public Body(string color)  
 {  
 this.color = color;  
 }  
   
 public void Info()  
 {  
 Console.WriteLine("Body color = " + color);  
 }  
 }  
   
 class Car\_Agregation  
 {  
 private Wheels wheels;  
 private Engine engine;  
 private Body body;  
  
 public Car\_Agregation(Wheels wheels, Engine engine, Body body)  
 {  
 this.wheels = wheels;  
 this.engine = engine;  
 this.body = body;  
 }  
   
 public void Info()  
 {  
 Console.WriteLine("\tCar Agregation:\n");  
 wheels.Info();  
 engine.Info();  
 body.Info();  
 Console.WriteLine("\n");  
 }  
 }  
  
 class Car\_Composition  
 {  
 private Wheels wheels;  
 private Engine engine;  
 private Body body;  
  
 public Car\_Composition()  
 {  
 wheels = new BBC(4);  
 //wheels = new Yamaha();  
   
 engine = new Engine(400);  
 body = new Body("Red");  
 }  
  
 public void Info()  
 {  
 Console.WriteLine("\tCar Composition:\n");  
 wheels.Info();  
 engine.Info();  
 body.Info();  
 Console.WriteLine("\n");  
 }  
 }  
   
   
 public static void Main(string[] args)  
 {  
 Wheels w = new Yamaha(5);  
 Wheels w2 = new BBC(7);  
 Engine e = new Engine(500);  
 Body b = new Body("Blue");  
   
 Car\_Agregation carAgregation = new Car\_Agregation(w, e, b);  
 carAgregation.Info();  
   
 Car\_Composition carComposition = new Car\_Composition();  
 carComposition.Info();  
   
 }  
 }  
}

**Результат:**



**Висновки:** отже, на цій лабораторній роботі я одержав практичні навички створення об'єктів-груп (агрегація, композиція).