**11 Полиморфизм**

Задание №1. Составить программу с одним родительским классом и двумя потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей, и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов (количество объектов> = 5).

Родительский класс – автомобиль (поле – название). Потомки: грузовой (поле – грузоподъёмность р в т), легковой (поле – объём двигателя V в см3). Полиморфные методы: расход горючего на 100км. Грузовой – . Легковой – .

Листинг программы:

public class Automobile

{

public String Name { get; set; }

public Automobile()

{

Name = "Car";

}

public virtual void Print()

{

Console.WriteLine(Name);

}

public virtual double GasConsumption()

{

return 0;

}

}

public class Truck : Automobile

{

public double p { get; set; }

public Truck(double P)

{

this.p = P;

}

public override void Print()

{

Console.WriteLine(Name);

Console.WriteLine("Consumption on 100km: " + GasConsumption());

Console.WriteLine("");

}

public override double GasConsumption()

{

return Math.Round(Math.Sqrt(p) \* 100, 4);

}

}

public class Car : Automobile

{

public double V { get; set; }

public Car(double v)

{

this.V = v;

}

public override void Print()

{

Console.WriteLine(Name);

Console.WriteLine("Gas consumption: " + GasConsumption());

Console.WriteLine("");

}

public override double GasConsumption()

{

return Math.Round(2.5 \* V,2);

}

}

class Program

{

public static void Main()

{

double p, v; string k; int R;

Console.WriteLine("Enter amount of auto: ");

R = Convert.ToInt32(Console.ReadLine());

Automobile[] Autos = new Automobile[R];

for (int i = 0; i < R; i++)

{

Console.WriteLine("Auto type:\n1 - Truck\n2 - Car\nEnter type: ");

k = Console.ReadLine();

if (k == "1")

{

Console.WriteLine("Enter load capacity: ");

p = Convert.ToInt32(Console.ReadLine());

Autos[i] = new Truck(p);

Console.WriteLine("Gas consumption on 100km - "+ " " + Autos[i].GasConsumption());

}

else if (k == "2")

{

Console.WriteLine("Enter engine volume: ");

v = Convert.ToInt32(Console.ReadLine());

Autos[i] = new Car(v);

Console.WriteLine("Gas consumption on 100km - " + " " + Autos[i].GasConsumption());

}

else if (k != "1" || k != "2")

{

Console.WriteLine("Invalid vehicle type!");

}

}

Таблица 11.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 1, 560, 2, 45 | 2366.4319, 112.5 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 11.1.

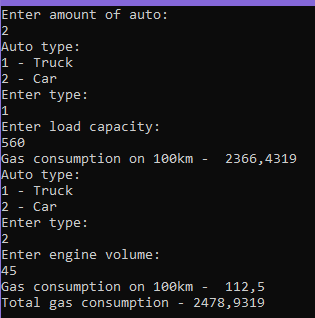


Рисунок 11.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Составить программу с абстрактным родительским классом и двумя объектами - потомками. Для этого модифицировать задание 2. Составить тестирующую программу с выдачей протокола на экран монитора. В ней нужно реализовать циклический вывод параметров объектов, используя полиморфный контейнер – массив объектов базового класса (количество объектов> = 5).

Организовать вычисление суммарного расхода горючего.

Листинг программы:

double sum = 0;

for (int i = 0; i < R; i++)

{

sum += Autos[i].GasConsumption();

}

Console.WriteLine($"Total gas consumption - {Math.Round(sum,4)}");

Console.ReadLine();

Таблица 11.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 2, 89, 1, 10 | 538.7278 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 11.2.

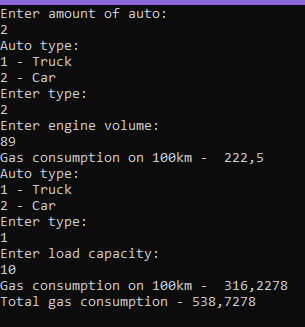


Рисунок 11.2 – Результат работы программы

Источник: собственная разработка