Практическое занятие №1

ПАТТЕРН ИНТЕРФЕЙС

1. Есть программа, где создан класс DUCK (утка) со следующими методами – quak(крякать), swim(плавать) и display(показать) и два наследуемых класса – MALLARDDUCK(дикая утка) и REDHEADDUCK (красноголовая утка), которые соответственно наследуют все методы класса DUCK, диаграмма классов данной программы представлена на рисунке 1.

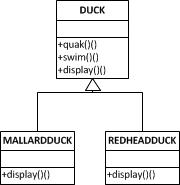


Рисунок -1.

1. Представим, что в класс DUCK добавили новый метод fly (летать) (Рисунок 2).

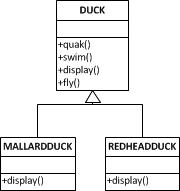
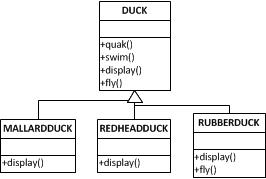


Рисунок -2.

1. А теперь, представим, что нам необходимо в программу добавить новый класс (новый тип уток) RUBBERDUCK (резиновая утка) (Рисунок 3). Резиновая утка не летает, поэтому метод fly() придется переопределить.



1. А если у нас будет тысяча видов уток и у каждой свой набор свойств, то нам придется переопределять для каждого класса какой-то или какие-то методы. Для этого существует следующий паттерн, который поможет избежать этого.
2. Создайте новый класс “Duck.cs” и новый класс “MallardDuck.cs”.

class Duck

{

protected FlyBehavior flybehavior;

protected QuackBehavior quackbehavior;

public Duck() { }

public void performFly(){

flybehavior.fly();

}

public void performQuack()

{

quackbehavior.quack();

}

public void swim() {}

}

class MallardDuck : Duck

{

public MallardDuck()

{

quackbehavior = new Quack();

flybehavior = new FlyWithWings();

System.Console.WriteLine("I MallardDuck");

}

}

1. Создайте интерфейс “FlyBehavior .cs” и два новых класса “FlyWithWings.cs” “FlyNoWay.cs”.

interface FlyBehavior

{

void fly();

}

class FlyWithWings : FlyBehavior

{

public void fly()

{

System.Console.WriteLine("I can fly");

}

}

class FlyNoWay : FlyBehavior

{

public void fly()

{

System.Console.WriteLine("I can`t fly");

}

}

1. Создайте интерфейс “QuakBehavior .cs” и два новых класса “Quak.cs” “Silence.cs”.

interface QuackBehavior

{

void quack();

}

class Quack : QuackBehavior

{

public void quack()

{

System.Console.WriteLine("Quack");

}

}

class Slience : QuackBehavior

{

public void quack()

{

System.Console.WriteLine("Slience");

}

}

1. В метод Main вставьте следующий код:

static void Main(string[] args)

{

Duck mallard = new MallardDuck();

mallard.performQuack();

mallard.performFly();

Duck wood = new WoodDuck();

wood.performQuack();

wood.performFly();

System.Console.ReadKey();

}

1. Создайте новый класс, характеризующий новую утку с новыми свойствами, придумайте интерфейс для нового свойства. Добавьте это свойство к уже существующим классам уток и выведите результаты на консоль.