Design Patterns / Шаблони за дизайн

Flyweight / Миниобект - структурен шаблон за дизайн

Шаблонът за дизайн Миниобект ви позволява да преизползвате паметта в приложението, когато имате много обекти, които са почти идентични по своя характер. Например, ако пишете игра за смартфон,където паметта е много ограничена и трябва да се покажат много извънземни, които са идентични по форма, може да има само едно място, което пази формата на чужденеца вместо да пази всеки идентична форма в скъпоценната паметта.

В шаблона Миниобект , съществува концепцията на външните(присъщи) и вътрешните(придобити) състояния. Присъщи състояния са тези неща, които са постоянни и се съхраняват в паметта. Външните състояния са неща, които не са постоянни и е необходимо да се изчисляват в движение, и поради това не се съхраняват в паметта.

Например, в играта, която бихме искали да създадем, формите на извънземните са едни и същи, но цветът им ще се променя в зависимост от това колко ядосани са. Формите на извънземните ще бъдат присъщи, а цветът на чужденеца ще бъде придобит.

public enum Color { Green, Red, Blue }

class Program

{

static void Main(string[] args)

{

*//create Aliens and store in factory*

AlienFactory factory = new AlienFactory();

factory.SaveAlien(0, new LargeAlien());

factory.SaveAlien(1, new LittleAlien());

*//now access the flyweight objects*

IAlien a = factory.GetAlien(0);

IAlien b = factory.GetAlien(1);

*//show intrinsic states, all accessed in memory without calculations*

Console.WriteLine("Showing intrinsic states...");

Console.WriteLine("Alien of type 0 is " + a.Shape);

Console.WriteLine("Alien of type 1 is " + b.Shape);

*//show extrinsic states, need calculations*

Console.WriteLine("Showing extrinsic states...");

Console.WriteLine("Alien of type 0 is " + a.GetColor(0).ToString());

Console.WriteLine("Alien of type 0 is " + a.GetColor(1).ToString());

Console.WriteLine("Alien of type 1 is " + b.GetColor(0).ToString());

Console.WriteLine("Alien of type 1 is " + b.GetColor(1).ToString());

}

}

public interface IAlien

{

string Shape { get; } *//intrinsic state*

Color GetColor(int madLevel); *//extrinsic state*

}

public class LargeAlien : IAlien

{

private string shape = "Large Shape"; *//intrinsic state*

string IAlien.Shape

{

get { return shape; }

}

Color IAlien.GetColor(int madLevel) *//extrinsic state*

{

if (madLevel == 0)

return Color.Green;

else if (madLevel == 1)

return Color.Red;

else

return Color.Blue;

}

}

public class LittleAlien : IAlien

{

private string shape = "Little Shape"; *//intrinsic state*

string IAlien.Shape

{

get { return shape; }

}

Color IAlien.GetColor(int madLevel) *//extrinsic state*

{

if (madLevel == 0)

return Color.Red;

else if (madLevel == 1)

return Color.Blue;

else

return Color.Green;

}

}

public class AlienFactory

{

private Dictionary<int,> list = new Dictionary<int,>();

public void SaveAlien(int index, IAlien alien)

{

list.Add(index, alien);

}

public IAlien GetAlien(int index)

{

return list[index];

}

}

