МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Лабораторна робота №1

з дисципліни «Об'єктно оріентовне програмування»

Тема: «Шаблони проектування. Породжуючі шаблони»

Виконав:

студенти групи ІП-81

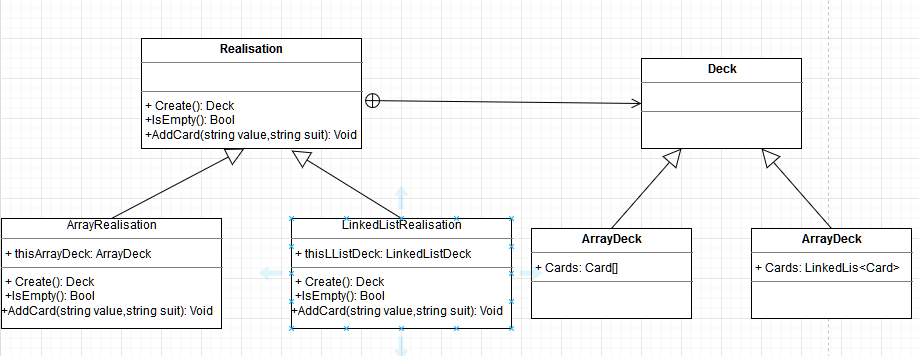
Балачін Петро

Київ-2020

**Мета роботи**: ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.

**Хід роботи:**

**Варіант 3**



class Program

{

static void Main(string[] args)

{

Realization array = new ArrayRealization();

array.Create();

Realization LList = new LinkedListRealization();

LList.Create();

Console.WriteLine(array.isEmpty());

Console.WriteLine(LList.isEmpty());

array.AddCard("Ace", "Spades");

array.AddCard("Six", "Hearts");

LList.AddCard("Seven", "Clubs");

}

}

abstract class Realization

{

abstract public Deck Create();

abstract public bool isEmpty();

abstract public void AddCard(string val,string suit);

}

class ArrayRealization : Realization

{

public ArrayDeck thisArrayDeck;

public override Deck Create()

{

thisArrayDeck = new ArrayDeck();

return thisArrayDeck;

}

public override bool isEmpty()

{

return (thisArrayDeck.Cards.Length == 0);

}

public override void AddCard(string val,string suit)

{

if(Enum.IsDefined(typeof(Values),val) && Enum.IsDefined(typeof(Suits), suit))

{

Array.Resize(ref thisArrayDeck.Cards, thisArrayDeck.Cards.Length + 1);

thisArrayDeck.Cards[thisArrayDeck.Cards.Length - 1] = new Card(suit, val);

Console.WriteLine("Card has been added.Now Array Deck is :");

foreach( Card el in thisArrayDeck.Cards)

{

Console.WriteLine(el.Value + " of " + el.Suit);

}

}

else

{

Console.WriteLine("Enter valid name of card");

}

}

}

class LinkedListRealization : Realization

{

public LinkedListDeck thisLListDeck;

public override Deck Create()

{

thisLListDeck = new LinkedListDeck();

return thisLListDeck;

}

public override bool isEmpty()

{

return (thisLListDeck.Cards.Count == 0);

}

public override void AddCard(string val, string suit)

{

if (Enum.IsDefined(typeof(Values), val) && Enum.IsDefined(typeof(Suits), suit))

{

thisLListDeck.Cards.AddLast(new Card(suit, val));

Console.WriteLine("Card has been added.Now Linked List Deck is :");

foreach (Card el in thisLListDeck.Cards)

{

Console.WriteLine(el.Value + " of " + el.Suit);

}

}

else

{

Console.WriteLine("Enter valid name of card");

}

}

}

abstract class Deck

{

}

class ArrayDeck : Deck

{

public Card[] Cards;

public ArrayDeck()

{

Cards = new Card[] { };

Console.WriteLine("ArrayDeck has created.");

}

}

class LinkedListDeck : Deck

{

public System.Collections.Generic.LinkedList<Card> Cards;

public LinkedListDeck()

{

Cards = new System.Collections.Generic.LinkedList<Card> { };

Console.WriteLine("LinkedListDeck has created.");

}

}

enum Suits

{

Hearts,Diamonds,Clubs,Spades

}

enum Values

{

Six,Seven,Eight,Nine,Ten,Jack,Queen,King,Ace

}

class Card

{

public Suits Suit { get; private set;}

public Values Value { get; private set; }

public Card(string suit,string value)

{

if (Enum.IsDefined(typeof(Values), value) && Enum.IsDefined(typeof(Suits), suit))

{

this.Suit = (Suits)Enum.Parse(typeof(Suits), suit);

this.Value = (Values)Enum.Parse(typeof(Values), value);

}

else

{

Console.WriteLine("Enter valid name of card");

}

}

}

**Висновок:**Я засвоїв теоретичний матеріал з відповідної теми та закріпив його,виконавши практичні завдання.