**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Державний університет інфраструктури та технологій

Коледж морського і річкового флоту

**Потрібен клас для спаткування**

**ЗВІТ**

**з лабораторної роботи № 8** **«Програмування модифікаторів доступу в аксесорах»**

з дисципліни **«Об’єктно-орієнтованого програмування»**

**Варіант № 11**

студент групи **ПР-318**

спеціальності **«ІПЗ»**

Пронь Дмитро Вікторович

**Викладач:** **Горбаченко В.А**.

**Київ – 2020**

**Мета роботи**: Навчитися створювати модифікатори доступу в аксесорах «get» та «set» на прикладі Windows Forms.

**Порядок виконання роботи**

1. На основі програмного коду ЛР№6 та індивідуального варіанту завдання об’єкту реального світу (Велосипед) реалізувано клас потомок вкладеного класу в головному класі.
2. Написано код ініціалізації конструкторів потомків вкладених класів з коментарями пророблених дій.
3. Написано код деініціалізації деструкторів потомків вкладених класів та головного класу з коментарями пророблених дій.
4. Проведено тестування всіх частин програмного коду на коректність роботи
5. Висновки: В даній лабораторній роботі був додан делегат з навою CallBack(), метод CallBack() виводить на екран повідомлення з ім’ям автора.

**Контрольні запитання**

* 1. **Що таке спадкування в мові програмування C#?**Спадкування є одним з трьох основних принципів об'єктно-орієнтованого програмування, оскільки воно допускає створення ієрархічних класифікацій.
  2. **В чому переваги використання спадкування?**Завдяки спадкуванню можна створити загальний клас, в якому визначаються характерні особливості, властиві безлічі пов'язаних елементів. Від цього класу можуть потім спадкувати інші, більш конкретні класи, додаючи в нього свої індивідуальні особливості.
  3. **Перелічити нюанси реалізації кодів конструкторів та деструкторів потомків вкладених класів в головному класі.**Створювати конструктор і деструктор не потрібно, оскільки класс, який наслідує інший класс також наслідує конструктор і деструктор.
  4. **Наскільки реалізований клас в C# був змінений в порівнянні з попереднім кодом? Що саме було реалізовано?**Код був суттєво змінений, був додан класс, який спадкує інший класс.
  5. **Чи відбулися в класі(-ах) кардинальні зміни в структурі? Якщо так, то чому саме?**Попередні класси не чіпалися, створювался лише 1 новий клас, який наслідував інший.

**Реалізація коду програми C#**

using k2020\_lr7;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace k2020\_lr7

{

class TWheel

{

// Клас «Колесо»

public int spokeCout; //Кількість спиць

public int diametr; //Діаметр колес

public void ScanParams()

{

Console.BackgroundColor = ConsoleColor.Red;

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("\n\n[TWheel] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Кiлькiсть спиць: ");

try

{

spokeCout = Convert.ToInt32(Console.ReadLine()); // Число

}

catch (Exception e)

{

Console.WriteLine("ERROR: Data entry error! The value will be zero!");

Console.WriteLine("DESCRIPTION: " + e.Message);

spokeCout = 0;

}

Console.Write("Дiаметр колес : ");

try

{

diametr = Convert.ToInt32(Console.ReadLine()); // Число

}

catch (Exception e)

{

Console.WriteLine("ERROR: Data entry error! The value will be zero!");

Console.WriteLine("DESCRIPTION: " + e.Message);

diametr = 0;

}

}

public void PrintParams()

{

Console.BackgroundColor = ConsoleColor.DarkGreen;

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\n\n[TWheel] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Кiлькiсть спиць : " + spokeCout);

Console.WriteLine("Дiаметр колес : " + diametr);

Console.WriteLine();

}

}

class Tframe

{

// Класс «Рама»

public string colour; //колір рами

public string type; //тип рами

public string material; //матеріал

public void ScanParams()

{

Console.BackgroundColor = ConsoleColor.Red;

Console.ForegroundColor =

ConsoleColor.Yellow;

Console.WriteLine("\n\n[Tframe] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Колiр рами: ");

colour = Console.ReadLine(); // Текст

Console.WriteLine("\n\n[Tframe] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Тип рами: ");

type = Console.ReadLine(); // Текст

Console.WriteLine("\n\n[Tframe] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Матеріал рами: ");

material = Console.ReadLine(); // Текст

}

public void PrintParams()

{

Console.BackgroundColor = ConsoleColor.DarkGreen;

Console.ForegroundColor =

ConsoleColor.White;

Console.WriteLine("\n\n[Tframe] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Колiр рами : " + colour);

Console.WriteLine();

Console.WriteLine("\n\n[Tframe] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Тип рами : " + colour);

Console.WriteLine();

Console.WriteLine("\n\n[Tframe] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Матеріал рами : " + colour);

Console.WriteLine();

}

}

class Trudder

{

// Клас «Руль»

public string colour; //колір руля

public int height; //Висота руля

public void ScanParams()

{

Console.BackgroundColor = ConsoleColor.Red;

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("\n\n[Trudder] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Колiр руля: ");

colour = Console.ReadLine(); // Текст

Console.Write("Висота руля : ");

try

{

height = Convert.ToInt32(Console.ReadLine()); // Число

}

catch (Exception e)

{

Console.WriteLine("ERROR: Data entry error! The value will be zero!");

Console.WriteLine("DESCRIPTION: " + e.Message);

height = 0;

}

}

public void PrintParams()

{

Console.BackgroundColor = ConsoleColor.DarkGreen;

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\n\n[Trudder] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Колiр руля : " + colour);

Console.WriteLine("Висота руля : " + height);

Console.WriteLine();

}

}

class TBike

{

// Класс «Велосипед»

public string model; //Модель

public string firm; //Назва фірми яка його виготовила

public string DateOfCreation; //дата виробництва

public bool isworked; //Працюючому стані

public TWheel wheel; // Клас «Колесо»

public Tframe frame; // Класс «Рама»

public Trudder rudder; // Клас «Руль»

public bool CheckDate(string dtstr)

{ // метод який перевіряє правильність вводу дати.

// Якщо дата невірна повертає false. Якщо вірна - true; Приймає дату в типі

bool date\_correct = true;

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

try

{

DateTime.Parse(dtstr);

}

catch (Exception e)

{

date\_correct = false;

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

return date\_correct;

}

public bool CheckName(string namestr)

{ // метод який перевіряє правильність вводу імені.

// Якщо ім'я введено некорректно, то повертає false, інакше - true; Приймає ім'я в типі string

bool name\_correct = true;

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

for (var i = 0; i < 38; i++)

{

if (!Regex.IsMatch(namestr, @"^[A-Za-z]+$"))

{

name\_correct = false;

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

return name\_correct;

}

public void Separator(int count, char symb)

{

var separator = "";

for (var i = 0; i < count; i++)

{

separator += symb;

}

Console.WriteLine(separator);

}

public void InputSeparatorDate(ref int count, ref char symb)

{

Console.WriteLine(" Enter count of symbols and symbol" + System.Environment.NewLine + " symbol: ");

try

{

symb = Convert.ToChar(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(" ERROR: Dta entry error!The value will be = ");

Console.WriteLine(" DESCRIPTION: " + e.Message);

symb = '=';

}

Console.WriteLine(" Count of symbols: ");

try

{

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

}

catch (Exception e)

{

Console.WriteLine(" ERROR: Dta entry error!The value will be zero!");

Console.WriteLine(" DESCRIPTION: " + e.Message);

count = 0;

}

}

public TBike()

{

// Constructor

wheel = new TWheel();

frame = new Tframe();

rudder = new Trudder();

}

~TBike()

{

// Destructor

}

public void ScanParams()

{

Console.BackgroundColor = ConsoleColor.Red;

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("[TBike] :: SCANPARAMS");

Console.ResetColor();

Console.Write("Модель : ");

model = Console.ReadLine(); // Текст

Console.Write("Фiрма : ");

firm = Console.ReadLine(); // Текст

if (CheckName(firm))

{

Console.WriteLine("Iм'я введено корректно!");

}

else

{

Console.WriteLine("Iм'я введено не корректно!");

}

Console.Write("Дата : ");

DateOfCreation = Console.ReadLine(); // Текст

if (CheckDate(DateOfCreation))

{

Console.WriteLine("Дата введена корректно!");

}

else

{

Console.WriteLine("Дата введена не корректно!");

}

Console.Write("Працюючому станi : ");

string tmp\_work = Console.ReadLine().ToUpper();

isworked = (tmp\_work == "YES" || tmp\_work == "Y" ? true : false); // bool

}

public void PrintParams()

{

Console.BackgroundColor = ConsoleColor.DarkGreen;

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("[TBike] :: PRINTPARAMS");

Console.ResetColor();

Console.WriteLine("Модель : " + model);

Console.WriteLine("Фiрма : " + firm);

Console.WriteLine("Дата виготовлення : " + DateOfCreation);

Console.WriteLine("В працюючому станi : " + isworked);

}

public void ScanAllParams(int count, char symb)

{

Separator(count, symb);

ScanParams();

Separator(count, symb);

wheel.ScanParams();

Separator(count, symb);

frame.ScanParams();

Separator(count, symb);

rudder.ScanParams();

Separator(count, symb);

}

public void PrintAllParams(int count, char symb)

{

PrintParams();

Separator(count, symb);

wheel.PrintParams();

Separator(count, symb);

frame.PrintParams();

Separator(count, symb);

rudder.PrintParams();

Separator(count, symb);

}

public delegate void CallBack();

public void ShowAuthor(CallBack obj)

{

obj();

}

}

}

class Program

{

public static int count = 12;

public static char symb = '=';

private static void CallBack()

{

Console.WriteLine("Author: Dmytro Pron ");

}

private static void Main(string[] args)

{

var bike = new TBike();

bike.InputSeparatorDate(ref count, ref symb);

bike.ScanAllParams(count, symb); // зчитуємо всі необхідні дані для заповнення полів класів

bike.Separator(count, symb);

bike.PrintAllParams(count, symb); // виводимо всі дані полів класів

bike.Separator(count, symb);

bike.ShowAuthor(CallBack);

Console.WriteLine("PROGRAM STOP!");

Console.Beep();

Console.ReadLine();

}

}