**ФГБОУ ВО**

**Национальный исследовательский университет**

**«МЭИ»**

­

**Лабораторная работа № 6**

по курсу

«Технология программирования»

**ООП C#. Множественное наследование.**

**Выполнил:**

студент группы А-04-23

Осипов Алексей

Вариант 11

**Москва, 2024**

**Лабораторная работа №6**

**Вариант №11**

**ООП C#. Множественное наследование.**

**Условие задачи**

1. Построить иерархию классов в соответствии с вариантом задания.
2. Разработать методы и свойства для каждого из определяемых классов.
3. Реализовать программу на C# в соответствии с вариантом задания.
4. Реализовать “дружественный - статичный” метод для нескольких классов.
5. Реализовать обработку исключительных ситуаций
6. Рассмотреть и описать (в вашей работе где), какие проблемы возникли при реализации множественного наследования ("алмаз смерти").

Вариант классов:

Автомобиль, лодка, транспортное средство, плавающий автомобиль

**1. Постановка задачи**

Реализовать класс транспортного средства (далее т.с.), с полями массы и максимальной скорости. В классе должны быть get-set методы, не допускающие отрицательных значений там, где физически это невозможно, а также дружественная (статичная) функция нахождения импульса т.с.

Класс-Лодка будет наследоваться от класса т.с. и добавит поле ватерлинии. Будут get-set методы с проверкой не отрицательность, а также своя реализация задания максимальной скорости, с проверкой не отрицательности, максимального значения и сохранением до целого **делимого на 10** для наглядности уникальности метода.

Класс-Машина наследуется от т.с. и добавит поле-флаг наличия автопилота, а также количества колес. Должны быть get-set методы с аналогичными проверками как у предшествующих классов, а также метод задания скорости с сохранением до целого **делимого на 5.**

Класс-Плавающая машина наследуется от класса машины и лодки. Добавит поле угла погружения т.с. в воду и get-set методы для этого поля. Функции задания скорости будет возвращать среднюю скорость родительских классов.

**2. Разработка программы**

**2.1 Разработка структуры классов**

****

**2.2 Реализация функций и процедур классов**

Дружественная (статичная) функция для подсчета импульса. На вход принимает любой объект, наследуемый от класса т.с. Возвращает значение импульса как double.

public static double getImpuls(Vehicle vehicle);

## Функции get типа реализованы через автоматически реализованные свойства {get;}.

Методы set типареализованы через автоматически реализованные свойства {set;} с добавлением проверки верного диапазона значения. Принимают значение для записи в объект, взамен ничего на дают. Метод setSpeed() реализован в каждом классе по разному, но выполняет одну и ту же функцию.

public virtual void setSpeed(double speed);

**2.3 Реализация основного цикла программы**

В функции main демонстрируются основные методы и функции классов. Создается по одному объекту от каждого из написанных классов и задается одинаковая скорость. Далее вызывается уведомление с подсчетом импульса объекта т.с. После закрытия уведомления, вызывается форма демонстрации различной реализации установки скорости у разных объектов (set\_speed). Это заметно по округлению значений до ближайшего делимого на 10, 5 и 1. Вывод скорости гибрида машины и лодки подсчитывается как среднее значение скоростей родителей. Скорости можно менять, и будет видна иерархическая зависимость: скорость т.с. не зависит от остальных объектов, в то время как при изменении скорости класса-родителя, все дочерние классы изменят значения своих скоростей по иерархии вниз.

**3 Возникшие трудности**

Невозможность множественного наследования в синтаксисе языка. Невозможность перегружать поля родительских классов при использовании множественного наследования с помощью интерфейсов. Возникает проблема точного определения дублирующихся полей и функций у класса, наследующегося от 2х родителей. Требуется явное указание класса, от которого берется метод. Также возникала проблема невозможности наследования дружественной функции вниз по иерархии и неявного ее вызова, поскольку дружественная функция-статична и нужно явное обращение к классу, содержащему функцию, но это решается приведением типов вверх по иерархии.

**4 Код основной программы**

namespace Lab6

{

internal static class Program

{

static void Main()

{

Vehicle veh = new Vehicle(3, 5);

MessageBox.Show("Инпульс через 'друж.' функцию: "+Convert.ToString(Vehicle.getImpuls(veh)));

ApplicationConfiguration.Initialize();

Application.Run(new Form1());

}

}

}

namespace Lab6

{

public partial class Form1 : Form

{

static Vehicle v = new Vehicle(2, -3);

static Car c = new Car();

static Boat b = new Boat();

static FloatingCar f = new FloatingCar();

public Form1()

{

InitializeComponent();

}

private void VehicleChanged(object sender, KeyEventArgs e)

{

if (e.KeyCode != Keys.Enter) { return; }

try { v.setSpeed(Convert.ToDouble(textBox1.Text)); }

catch {

textBox1.Text = v.Speed.ToString();

return;

}

c.setSpeed(v.Speed);

b.setSpeed(v.Speed);

textBox2.Text = c.Speed.ToString();

textBox3.Text = b.Speed.ToString();

CarChanged(sender, e);

BoatChanged(sender, e);

}

private void CarChanged(object sender, KeyEventArgs e)

{

if (e.KeyCode != Keys.Enter) { return; }

try { c.Speed = Convert.ToDouble(textBox2.Text); }

catch

{

textBox2.Text = c.Speed.ToString();

return;

}

f.setSpeed(c.Speed, b.Speed);

textBox4.Text = ((c.Speed + b.Speed) / 2).ToString();

FLChanged(sender, e);

}

private void BoatChanged(object sender, KeyEventArgs e)

{

if (e.KeyCode != Keys.Enter) { return; }

try { b.Speed = Convert.ToDouble(textBox3.Text); }

catch

{

textBox3.Text = b.Speed.ToString();

return;

}

f.setSpeed(c.Speed, b.Speed);

textBox4.Text = ((c.Speed + b.Speed) / 2).ToString();

FLChanged(sender, e);

}

private void FLChanged(object sender, KeyEventArgs e)

{

if (e.KeyCode != Keys.Enter) { return; }

try { f.setSpeed(Convert.ToDouble(textBox4.Text)); }

catch

{

textBox4.Text = f.Speed.ToString();

return;

}

}

}

}

namespace Lab6

{

partial class Form1

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

private void InitializeComponent()

{

label1 = new Label();

label2 = new Label();

label3 = new Label();

label4 = new Label();

textBox1 = new TextBox();

textBox2 = new TextBox();

textBox3 = new TextBox();

textBox4 = new TextBox();

SuspendLayout();

//

// label1

//

label1.AutoSize = true;

label1.Font = new Font("Segoe UI", 14F);

label1.Location = new Point(34, 34);

label1.Margin = new Padding(25);

label1.Name = "label1";

label1.Size = new Size(154, 32);

label1.TabIndex = 0;

label1.Text = "Скорость т.с.";

//

// label2

//

label2.AutoSize = true;

label2.Font = new Font("Segoe UI", 14F);

label2.Location = new Point(34, 116);

label2.Margin = new Padding(25);

label2.Name = "label2";

label2.Size = new Size(216, 32);

label2.TabIndex = 2;

label2.Text = "Скорость машины";

//

// label3

//

label3.AutoSize = true;

label3.Font = new Font("Segoe UI", 14F);

label3.Location = new Point(34, 198);

label3.Margin = new Padding(25);

label3.Name = "label3";

label3.Size = new Size(189, 32);

label3.TabIndex = 3;

label3.Text = "Скорость лодки";

//

// label4

//

label4.AutoSize = true;

label4.Font = new Font("Segoe UI", 14F);

label4.Location = new Point(34, 280);

label4.Margin = new Padding(25);

label4.Name = "label4";

label4.Size = new Size(213, 32);

label4.TabIndex = 4;

label4.Text = "Скорость гибрида";

//

// textBox1

//

textBox1.Font = new Font("Segoe UI", 14F);

textBox1.Location = new Point(287, 34);

textBox1.Margin = new Padding(0);

textBox1.Name = "textBox1";

textBox1.Size = new Size(201, 39);

textBox1.TabIndex = 6;

textBox1.Text = "0";

textBox1.KeyDown += VehicleChanged;

//

// textBox2

//

textBox2.Font = new Font("Segoe UI", 14F);

textBox2.Location = new Point(287, 116);

textBox2.Margin = new Padding(0);

textBox2.Name = "textBox2";

textBox2.Size = new Size(201, 39);

textBox2.TabIndex = 8;

textBox2.Text = "0";

textBox2.KeyDown += CarChanged;

//

// textBox3

//

textBox3.Font = new Font("Segoe UI", 14F);

textBox3.Location = new Point(287, 198);

textBox3.Margin = new Padding(0);

textBox3.Name = "textBox3";

textBox3.Size = new Size(201, 39);

textBox3.TabIndex = 9;

textBox3.Text = "0";

textBox3.KeyDown += BoatChanged;

//

// textBox4

//

textBox4.Font = new Font("Segoe UI", 14F);

textBox4.Location = new Point(287, 280);

textBox4.Margin = new Padding(0);

textBox4.Name = "textBox4";

textBox4.Size = new Size(201, 39);

textBox4.TabIndex = 10;

textBox4.Text = "0";

textBox4.KeyDown += FLChanged;

//

// Form1

//

AutoScaleDimensions = new SizeF(8F, 20F);

AutoScaleMode = AutoScaleMode.Font;

BackColor = SystemColors.ScrollBar;

ClientSize = new Size(548, 448);

Controls.Add(textBox4);

Controls.Add(textBox3);

Controls.Add(textBox2);

Controls.Add(textBox1);

Controls.Add(label4);

Controls.Add(label3);

Controls.Add(label2);

Controls.Add(label1);

Name = "Form1";

Text = "Form1";

ResumeLayout(false);

PerformLayout();

}

#endregion

private Label label1;

private Label label2;

private Label label3;

private Label label4;

private TextBox textBox1;

private TextBox textBox2;

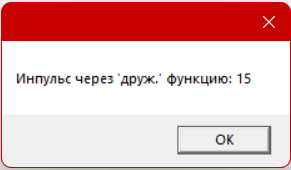
private TextBox textBox3;

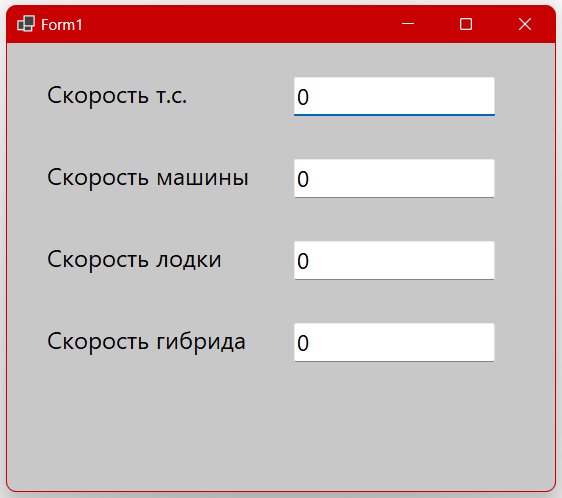
private TextBox textBox4;

}

}

**5 Вид формы программы**

****

****