МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра СТ

Отчет

о выполнении лабораторной работы №3

«КЛАССЫ. ПЕРЕГРУЗКА МЕТОДОВ. НАСЛЕДОВАНИЕ»»

по дисциплине «Программирование под платформу .Net»

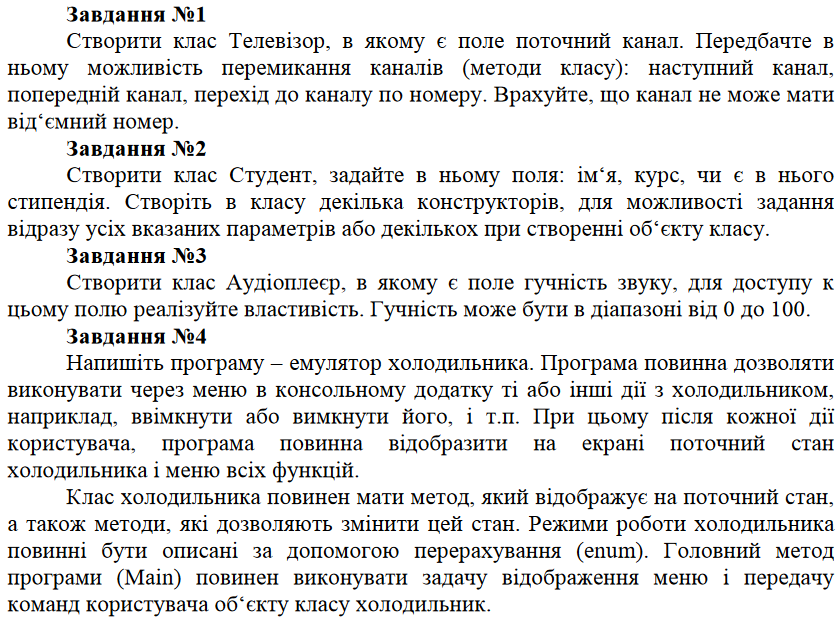
|  |  |
| --- | --- |
| Выполнил ст. гр. КН-15-2:  Антонюк М.В. | Проверила:  Жернова П.Е. |

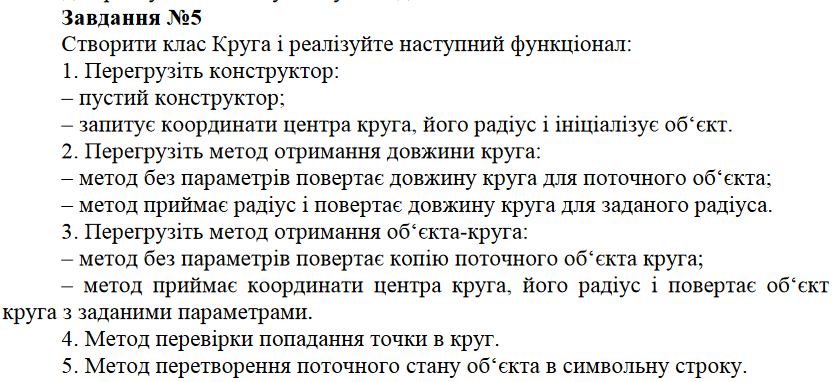
Харьков 2018

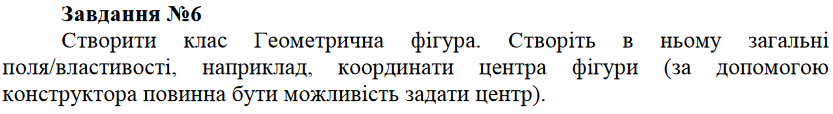
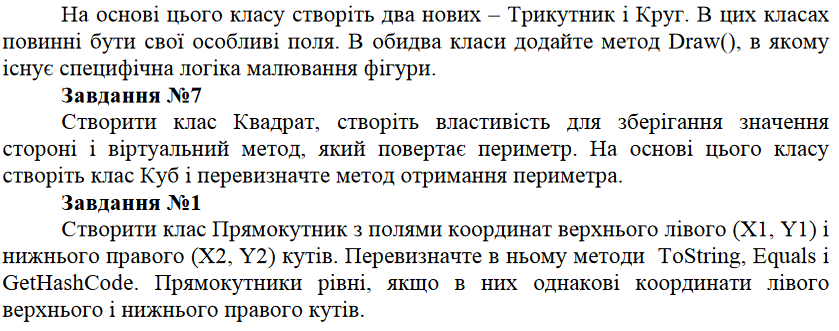
**Цель работы**

Научиться создавать собственные классы и объекты. Реализовать принцип наследования и перегрузку методов.

**Задание**





**Ход работы**

**Задание 1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class TV

{

private int current;

public TV() { current = 1; }

public void next()

{

current = (current == 100) ? 1 : current + 1;

}

public void previous()

{

current = (current == 1) ? 100 : current - 1;

}

public void setChanel(int chanel)

{

if (chanel > 0 && chanel < 101)

current = chanel;

}

}

}

**Задание 2**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Student

{

private string name;

private int course;

private bool grant;

public Student(string n, int c, bool g)

{

name = n;

course = c;

grant = g;

}

public Student(string n, int c)

{

name = n;

course = c;

}

public Student(string n, bool g)

{

name = n;

grant = g;

}

}

}

**Задание 3**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Audioplayer

{

private int volume;

public Audioplayer() { volume = 0; }

public int getVolume() { return volume; }

public void setVolume(int volume) { if (volume >= 0 && volume <= 100) this.volume = volume; }

}

}

**Задание 4**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Fridge

{

bool on;

public bool On{ get { return on; } }

bool lightOn;

public bool LightOn { get { return lightOn; } }

bool opened;

public bool Opened { get { return opened; } }

bool full;

public bool Full { get { return full; } }

public enum mode { economy, cold, supercold };

string currMode;

public Fridge()

{

on = false;

lightOn = false;

opened = false;

full = false;

currMode = mode.cold.ToString();

}

public void status()

{

Console.WriteLine("Текущее состояние холодильника: ");

Console.WriteLine("Включен: " + on.ToString());

Console.WriteLine("Свет включён: " + lightOn.ToString());

Console.WriteLine("Открыт: " + opened.ToString());

Console.WriteLine("Полный: " + full.ToString());

Console.WriteLine("Режим охлаждения: " + currMode);

Console.WriteLine();

}

public void turnOn()

{

on = true;

status();

}

public void turnOff()

{

on = false;

status();

}

public void open()

{

opened = true;

lightOn = true;

status();

}

public void close()

{

opened = false;

lightOn = false;

status();

}

public void fill()

{

full = true;

status();

}

public void empty()

{

full = false;

status();

}

public void changeMode(mode m)

{

currMode = m.ToString();

status();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class FridgeEmulator

{

static void Main(string[] args)

{

Fridge fr = new Fridge();

fr.status();

bool end = false;

string command, number;

do

{

Console.WriteLine((fr.On ? "Выключить" : "Включить") + " холодильник - нажмите " + ((fr.On) ? "2" : "1"));

Console.WriteLine((fr.Opened ? "Закрыть" : "Открыть") + " холодильник - нажмите " + (fr.Opened ? "4" : "3"));

Console.WriteLine((fr.Full ? "Опустошить" : "Наполнить") + " хололдильник - нажмите " + (fr.Full ? "6" : "5"));

Console.WriteLine("Изменить режим охлаждения - нажмите 7");

try

{

switch (int.Parse(Console.ReadLine()))

{

case 1:

fr.turnOn();

break;

case 2:

fr.turnOff();

break;

case 3:

fr.open();

break;

case 4:

fr.close();

break;

case 5:

fr.open();

fr.fill();

fr.close();

break;

case 6:

fr.open();

fr.empty();

fr.close();

break;

case 7:

Console.WriteLine("Режим 'economy' - введите 1 \nРежим 'cold' - введите 2 \nРежим 'supercold' - введите 3");

switch (int.Parse(Console.ReadLine()))

{

case 1:

fr.changeMode(Fridge.mode.economy);

break;

case 2:

fr.changeMode(Fridge.mode.cold);

break;

case 3:

fr.changeMode(Fridge.mode.supercold);

break;

}

break;

}

Console.WriteLine();

Console.WriteLine("Продолжить - нажмите 1 Закончить - нажмите 2");

switch (int.Parse(Console.ReadLine()))

{

case 1:

end = false;

break;

case 2:

end = true;

break;

default:

Console.WriteLine("Нет такого варианта");

break;

}

}

catch(FormatException)

{

Console.WriteLine("Ошибка. Неверный формат ввода");

}

finally { fr.status(); }

} while (!end);

}

}

}

**Задание 5**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class CircleTask5

{

double x;

double y;

double r;

public CircleTask5() { }

public CircleTask5(double x, double y, double r)

{

if (r < 0) throw new NegativeException(NegativeException.error);

this.x = x;

this.y = y;

this.r = r;

}

public double lenght()

{

return 2 \* r \* Math.PI;

}

public double lenght(double r)

{

if (r < 0) throw new NegativeException(NegativeException.error);

return 2 \* Math.PI \* r;

}

public CircleTask5 getCircle()

{

return new CircleTask5(this.x, this.y, this.r);

}

public CircleTask5 getCircle(double x, double y, double r)

{

return new CircleTask5(x, y, r);

}

public bool isOnCircle(double x, double y)

{

return Math.Sqrt(Math.Pow(this.x - x, 2) + Math.Pow(this.y - y, 2)) <= this.r;

}

public override string ToString()

{

return "Центр(" + x + ";" + y + ") Радиус = " + r;

}

}

}

**Задание 6**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public struct point { public double x; public double y; };

public class Figure

{

protected point center;

public Figure() { center.x = 0; center.y = 0; }

public Figure(point p) { center = p; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

class CircleTask6 : Figure

{

double r;

public CircleTask6() { r = 0; }

public CircleTask6(point p, double r) { center = p; this.r = r; }

public void Draw()

{

Console.WriteLine("Представьте что здесь нарисован круг");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

class TriangleTask6 : Figure

{

point a;

point b;

point c;

public TriangleTask6() { }

public TriangleTask6(point a, point b, point c, point center)

{

this.a = a;

this.b = b;

this.c = c;

this.center = center;

}

public void Draw()

{

Console.WriteLine("Представьте что здесь нарисован треугольник");

}

}

}

**Задание 7**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Square

{

protected double a;

public double A

{

get { return a; }

set

{

if (value < 0) throw new NegativeException(NegativeException.error);

a = value;

}

}

public Square() { }

public Square(double a)

{

this.a = a;

}

public virtual double perimeter()

{

return a \* 4;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Cube : Square

{

public Cube() { }

public Cube(double a) : base(a) { }

public override double perimeter()

{

return 12 \* a;

}

}

}

**Задание 8**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class Rectangle

{

point a;

point b;

public Rectangle(point a, point b)

{

if (a.x < b.x || a.y < b.y)

throw new Exception();

this.a = a; this.b = b;

}

public override string ToString()

{

return "Точка А (" + a.x + "; " + a.y + ") \nТочка В (" + b.x + "; " + b.y + ") ";

}

public override bool Equals(object obj)

{

if (obj == null)

return false;

Rectangle rec = obj as Rectangle;

if ((System.Object)rec == null)

return false;

return (a.x == rec.a.x) && (b.x == rec.b.x) && (a.y == rec.a.y) && (b.y == rec.b.y);

}

public override int GetHashCode()

{

int result = 1;

result += 31 \* result + (int)a.x;

result += 31 \* result + (int)a.y;

result += 31 \* result + (int)b.x;

result += 31 \* result + (int)b.y;

return result;

}

}

}

**Исключение для проверки на не отрицательность числа**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

public class NegativeException : Exception

{

public static readonly string error = "Ошибка. Число не может быть положительным!";

public NegativeException() { }

public NegativeException(string message) : base(message) { }

public NegativeException(string message, Exception inner) : base(message, inner) { }

}

}

**Результаты работы**

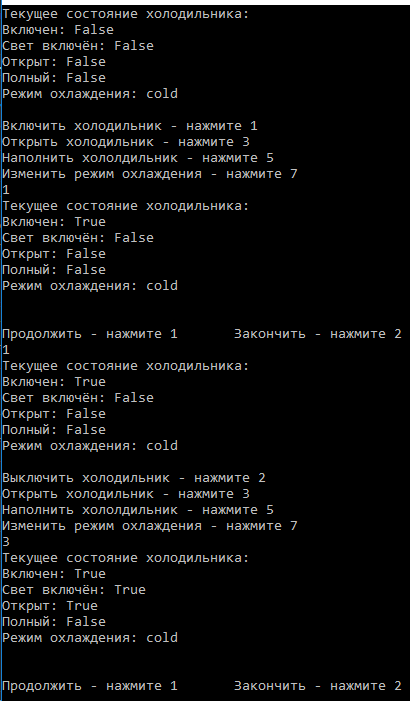
****

Рисунок 1 – Эмулятор холодильника

**Выводы.**

В ходе лабораторной работы были отработаны навыки работы со структурами. Также были проведены эксперименты с созданием классов, написанием методов, переопределением методов. Были изучены особенности переопределения виртуальных методов. Было изучено наследование и переопределение методов в наследниках.