**Практикум 12**

**Вариант 4**

<https://github.com/VsevolodTilta/Practic>

**Задание 1.**

В класс Money добавить:

1. Индексатор, позволяющий по индексу 0 обращаться к полю first, по индексу 1 – к полю second, при других значениях индекса выдается сообщение об ошибке.
2. Перегрузку:
   * операции ++ (--): одновременно увеличивает (уменьшает) значение полей first и second;
   * операции !: возвращает значение true, если поле second не нулевое, иначе false;
   * операции бинарный +: добавляет к значению поля second значение скаляра;
   * преобразования типа Money в string (и наоборот).

**Листинг приложения**

static void Main(string[] args)

{

Money cash = new Money();

Console.Write("Введите номер индекса: ");

int n = int.Parse(Console.ReadLine());

try

{

Console.WriteLine(cash[n]);

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.WriteLine(cash);

Console.Write("Введите стоимость желаемого товара: ");

int price = int.Parse(Console.ReadLine());

Console.WriteLine(cash.CheckAvailability(price));

Console.Write("Введите кол-во желаемого товара: ");

int sumItem = int.Parse(Console.ReadLine());

Console.WriteLine(cash.CalculationOfQuantity(sumItem));

Console.WriteLine(cash++);

Console.WriteLine(cash--);

if(!cash)

{

Console.WriteLine("Поле \"second\" не нулевое");

}

else

{

Console.WriteLine("Поле \"second\" нулевое");

}

Console.WriteLine(cash+35);

Console.WriteLine((string)cash);

Console.WriteLine((Money)"1 2");

Console.ReadKey();

}

**class Money**

{

int first;

int second;

int totalCash;

public Money()

{

first = 222;

second = 322;

}

public Money(int value, int counts)

{

this.first = value;

this.second = counts;

this.totalCash = value \* counts;

}

public double this[int index]

{

get

{

if (index == 0) return first;

else if (index == 1) return second;

else

{

throw new Exception("Введите верный индекс!");

}

}

set

{

if (index == 0) first = Convert.ToInt32(value);

else if (index == 1) second = Convert.ToInt32(value);

else

{

throw new Exception("Введите верный индекс!");

}

}

}

public static Money operator ++(Money ob) //одновременно увеличивает значение полей first и second

{

ob.first += 1;

ob.second += 1;

return ob;

}

public static Money operator --(Money ob) //одновременно уменьшает значение полей first и second

{

try

{

ob.first -= 1;

ob.second -= 1;

if (ob.second < 0 | ob.first < 0) throw new System.ArithmeticException();

}

catch (System.ArithmeticException)

{

Console.WriteLine("Уменьшение не возможно !");

ob.first +=1;

ob.second += 1;

}

return ob;

}

public static bool operator !(Money ob) //возвращает значение true, если поле second не нулевое, иначе false

{

if (ob.second != 0) return true;

else return false;

}

public static Money operator +(Money ob, double n) //добавляет к значению поля second значение скаляра

{

ob.second = Convert.ToInt32(ob.second + n);

return ob;

}

public static explicit operator string(Money ob) //Преобразование типа Money в string

{

return string.Concat(string.Concat(ob.first, ' '), ob.second);

}

public static explicit operator Money(string str) //Преобразование типа Money в string

{

return new Money(Convert.ToInt32(str.Split(' ')[0]), Convert.ToInt32(str.Split(' ')[1]));

}

public override string ToString()

{

return $"У Вас в кармане {second} куп. номиналом {first} руб.";

}

public string CheckAvailability(double costOfGoods)

{

totalCash = first \* second;

if (costOfGoods < totalCash)

{

return $"У Вас достаточно наличных ({totalCash} руб.) для покупки товара стоимость {costOfGoods} руб.";

}

else

{

return $"К сожалению, для покупки товара стоимостью {costOfGoods} Вам не хватает {costOfGoods - totalCash} руб. ";

}

}

public string CalculationOfQuantity(double costOfGoods)

{

int quantityGoods = (int)(totalCash / costOfGoods);

return $"На имеющиеся {totalCash} руб. можно купить {costOfGoods} ед. товара стоимостью {quantityGoods} руб.";

}

public int Nominal

{

get => first;

set => first = value;

}

public int QuantityNotes

{

get => second;

set => second = value;

}

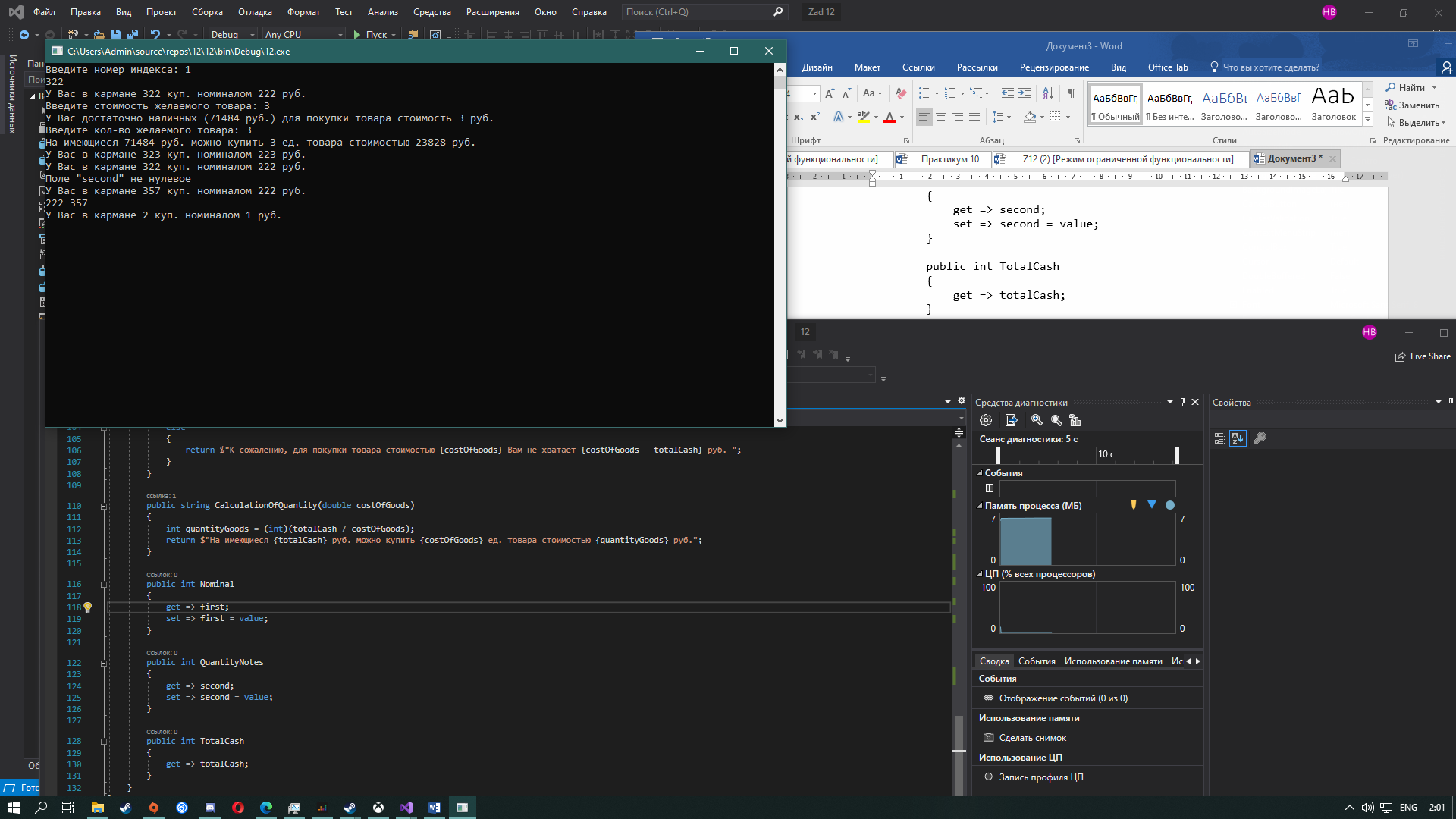
public int TotalCash

{

get => totalCash;

}

**Консольное приложение**



**Листинг windows forms**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Money cash = new Money();

int n = int.Parse(textBox1.Text);

try

{

richTextBox1.Text += cash[n];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

richTextBox1.Text +=cash + "\n";

int price = int.Parse(textBox2.Text);

richTextBox1.Text += cash.CheckAvailability(price) + "\n";

int sumItem = int.Parse(textBox3.Text);

richTextBox1.Text += cash.CalculationOfQuantity(sumItem) + "\n";

richTextBox1.Text += cash++ + "\n";

richTextBox1.Text += cash-- + "\n";

if (!cash)

{

richTextBox1.Text += "Поле \"second\" не нулевое" + "\n";

}

else

{

richTextBox1.Text += "Поле \"second\" нулевое" + "\n";

}

richTextBox1.Text += cash + 35 + "\n";

richTextBox1.Text += (string)cash + "\n";

richTextBox1.Text += (Money)"1 2" + "\n";

}

}

**class Money**

{

int first;

int second;

int totalCash;

public Money()

{

first = 222;

second = 322;

}

public Money(int value, int counts)

{

this.first = value;

this.second = counts;

this.totalCash = value \* counts;

}

public double this[int index]

{

get

{

if (index == 0) return first;

else if (index == 1) return second;

else

{

throw new Exception("Введите верный индекс!");

}

}

set

{

if (index == 0) first = Convert.ToInt32(value);

else if (index == 1) second = Convert.ToInt32(value);

else

{

throw new Exception("Введите верный индекс!");

}

}

}

public static Money operator ++(Money ob) //одновременно увеличивает значение полей first и second

{

ob.first += 1;

ob.second += 1;

return ob;

}

public static Money operator --(Money ob) //одновременно уменьшает значение полей first и second

{

try

{

ob.first -= 1;

ob.second -= 1;

if (ob.second < 0 | ob.first < 0) throw new System.ArithmeticException();

}

catch (System.ArithmeticException)

{

Console.WriteLine("Уменьшение не возможно !");

ob.first += 1;

ob.second += 1;

}

return ob;

}

public static bool operator !(Money ob) //возвращает значение true, если поле second не нулевое, иначе false

{

if (ob.second != 0) return true;

else return false;

}

public static Money operator +(Money ob, double n) //добавляет к значению поля second значение скаляра

{

ob.second = Convert.ToInt32(ob.second + n);

return ob;

}

public static explicit operator string(Money ob) //Преобразование типа Money в string

{

return string.Concat(string.Concat(ob.first, ' '), ob.second);

}

public static explicit operator Money(string str) //Преобразование типа Money в string

{

return new Money(Convert.ToInt32(str.Split(' ')[0]), Convert.ToInt32(str.Split(' ')[1]));

}

public override string ToString()

{

return $"У Вас в кармане {second} куп. номиналом {first} руб.";

}

public string CheckAvailability(double costOfGoods)

{

totalCash = first \* second;

if (costOfGoods < totalCash)

{

return $"У Вас достаточно наличных ({totalCash} руб.) для покупки товара стоимость {costOfGoods} руб.";

}

else

{

return $"К сожалению, для покупки товара стоимостью {costOfGoods} Вам не хватает {costOfGoods - totalCash} руб. ";

}

}

public string CalculationOfQuantity(double costOfGoods)

{

int quantityGoods = (int)(totalCash / costOfGoods);

return $"На имеющиеся {totalCash} руб. можно купить {costOfGoods} ед. товара стоимостью {quantityGoods} руб.";

}

public int Nominal

{

get => first;

set => first = value;

}

public int QuantityNotes

{

get => second;

set => second = value;

}

public int TotalCash

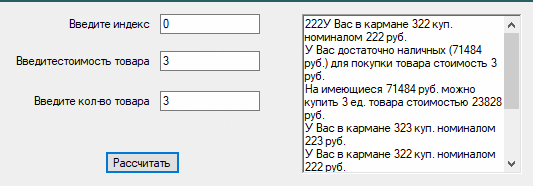
{

get => totalCash;

}

}

**Приложение windows forms**

****