УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**“ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО”**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

по дисциплине «Объектно-ориентированное программирование»

На тему **«**Особенности разработки и верификации в объектно-ориентированном языке**»**

Выполнил:

студент гр. ИТП-21

Клевцов А.О.

Принял: ассистент

Гуменников Е.Д.

Гомель 2022

**Цель работы:** освоить особенности разработки и верификации в объектно-ориентированном языке.

**Задание:**

* 1. Необходимо разработать тип и выполнить для него перегрузку операций, согласно варианта (таблица 2)
  2. Тип должен быть размещён в библиотеке классов.
  3. Весь код должен быть снабжён элементами документирования
  4. Разработать не менее модульные тестов для тестирования, созданного класса
  5. Создать консольное приложение с интерфейсом пользователя
  6. Подключить созданную библиотеку в консольное приложение
  7. Сгенерировать XML-файл с документацией по проекту
  8. Верифицировать консольное приложение

Таблица 1

|  |  |
| --- | --- |
| **Вариант** | **Условие Задачи** |
| 11. | Разработать тип для хранения информации о результатах сдачи студентом экзамена, при этом известны ФИО студента и полученная оценка на экзамене.  Реализовать методы, позволяющие находить сумму и разность между  двумя типами, представляющие собой тип, содержащий средний бал на экзамене. Предусмотреть возможность выполнения операций. |

**Ход работы**

В ходе лабораторной работы было написано консольное приложение, библиотека классов для хранения информации о студенте и его отметке. Для класса были перегружены операторы сложения и вычитания, написана функция для нахождения средней оценки n-ого количества студентов. Были созданы 3 конструктора класса. Написан тест для перегруженных операций.

Листинг программ будет представлен в приложении А

На рисунках будут представлены результат выполнения тестов и информация, выдаваемая в консоль после запуска приложения.

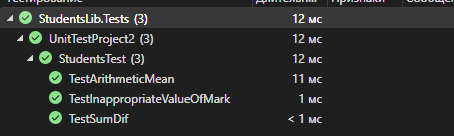


Рисунок 1 ­­– Успешное выполнение тестирования

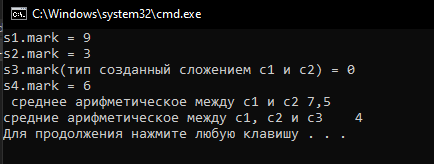


Рисунок 2 ­­– Результат выполнения приложения

**Вывод:** в ходе лабораторной работы были освоены основы синтаксиса объектно-ориентированного языка программирования на языке *c#*. Также были закреплены знания о то, что такое классы, экземпляры классов, конструкторы классов, а также получены знания о том, что такое перегрузка методов и как её делать.

**Приложение А**

*Main*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB2

{

public class Program

{

public static float ArithmeticMean(params Student []students)

{

float result = 0;

foreach (Student student in students)

{

result += student.mark;

}

return result/students.Length;

}

static void Main(string[] args)

{

Student s1 = new Student(9);

Student s2 = new Student(3);

Student s3 = s1 + s2;

Student s4 = s1 + s3 - s2;

Console.WriteLine("s1.mark = " + s1.mark);

Console.WriteLine("s2.mark = " + s2.mark);

Console.WriteLine("s3.mark = " + s3.mark);

Console.WriteLine("s4.mark = " + s4.mark);

Console.WriteLine(ArithmeticMean(s1,s4));

Console.WriteLine(ArithmeticMean(s1, s2, s3));

}

}

}

*Student*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB2

{

public class Student

{

string lastName, firstName, middleName;

public int mark;

public Student(string lastName, string firstName, string middleName, int mark)

{

this.lastName = lastName;

this.firstName = firstName;

this.middleName = middleName;

this.mark = mark;

}

public Student()

{

lastName = "";

firstName = "";

middleName = "";

mark = 1;

}

public Student(int mark)

{

lastName = "";

firstName = "";

middleName = "";

if(mark > 0 && mark <=10)

this.mark = mark;

}

public static Student operator +(Student s1, Student s2)

{

return new Student(s1.mark + s2.mark);

}

public static Student operator -(Student s1, Student s2)

{

return new Student(s1.mark - s2.mark);

}

}

}

*test*

using System;

using LAB2;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace UnitTestProject2

{

[TestClass]

public class StudentsTest

{

[TestMethod]

public void TestSumDif()

{

int firstMark = 5;

int secondMark = 2;

int expectedSum = firstMark + secondMark;

int expectedDif = firstMark - secondMark;

Student student1 = new Student(firstMark);

Student student2 = new Student(secondMark);

Student studentSum = student1 + student2;

Student studentDif = student1 - student2;

Assert.AreEqual(expectedSum, studentSum.mark);

Assert.AreEqual(expectedDif, studentDif.mark);

}

}

}