МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Отчёт по лабораторной работе по дисциплине «Тестирование программного обеспечения»

**Параметризованные тесты**

|  |  |
| --- | --- |
| Выполнил: студент группы ФИб-4302-51-00 | / Д.А. Савин / |
| Проверил: ст. преподаватель каф. ПМиИ | / И.Н. Фищева / |

Киров 2022

Задание

Загрузите код параметризированного теста. Для этого откройте код Лабораторной работы №1. Найдите несколько тестов, которые можно переписать в один тест, пользуясь примерами из MSTest v2: Data tests И создайте эти же тесты используя TestCase() из NUnit.

Изображение выглядит как текст

Автоматически созданное описание

|  |
| --- |
| [DataTestMethod]  [DataRow(90, 0, -90, 0, "any")] //Для точек, лежащих на прямой, проходящей через центр Земли азимут = any  [DataRow(-90, 0, 90, 0, "any")]  [DataRow(90, 0, -90, 0, "any")] //Для точек на противоположных полюса (см 2) азимут = any  [DataRow(-90, 0, 90, 0, "any")]  public void TestGetAzimuthAny(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |
| [Test]  [TestCase(90, 0, -90, 0, "any")] //Для точек, лежащих на прямой, проходящей через центр Земли азимут = any  [TestCase(-90, 0, 90, 0, "any")]  [TestCase(90, 0, -90, 0, "any")] //Для точек на противоположных полюса (см 2) азимут = any  [TestCase(-90, 0, 90, 0, "any")]  public void TestGetAzimuthAny(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |

2.

|  |
| --- |
| [DataTestMethod]  [DataRow(80, 40, 80, 40, "none")] //Для совпадающих точек азимут = none  [DataRow(10, 10, 10, 10, "none")]  [DataRow(90, 0, 90, 0, "none")] //Для двух точек на одном полюсе азимут = none  [DataRow(-90, 0, -90, 0, "none")]  public void TestGetAzimuthNone(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |
| [Test]  [TestCase(80, 40, 80, 40, "none")] //Для совпадающих точек азимут = none  [TestCase(10, 10, 10, 10, "none")]  [TestCase(90, 0, 90, 0, "none")] //Для двух точек на одном полюсе азимут = none  [TestCase(-90, 0, -90, 0, "none")]  public void TestGetAzimuthNone(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |

3.

|  |
| --- |
| [DataTestMethod]  [DataRow(90, 0, 80, 40, "180")] //Для одной точки на северном полюсе азимут = 180  [DataRow(40, 10, 90, 0, "180")]  [DataRow(-90, 0, 15, 0, "180")] //Для одной точки на южном полюсе азимут = 180  public void TestGetAzimuth180(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |
| [Test]  [TestCase(90, 0, 80, 40, "180")] //Для одной точки на северном полюсе азимут = 180  [TestCase(40, 10, 90, 0, "180")]  [TestCase(-90, 0, 15, 0, "180")] //Для одной точки на южном полюсе азимут = 180  public void TestGetAzimuth180(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |

4.

|  |
| --- |
| [DataTestMethod]  [DataRow(0, 120, 0, 50, "270")] //Для точек на экваторе 90 или 270, смотря куда ближе  public void TestGetAzimuth270(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |
| [Test]  [TestCase(0, 120, 0, 50, "270")] //Для точек на экваторе 90 или 270, смотря куда ближе  public void TestGetAzimuth270(int latitude, int longitude, int latitude\_2, int longitude\_2, string expected)  {  Point point\_1 = new Point(latitude, longitude);  Point point\_2 = new Point(latitude\_2, longitude\_2);  string actual = point\_1.Azimuth(point\_2);  Assert.AreEqual(expected, actual);  } |