using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using laba1;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

//сравнение чисел текущего(Cur) элемента кольца и другого элемента того же класса Element при равных значениях;

public void EaseEquals\_With\_Same\_Number\_Are\_Numbers\_Equals()

{

//Arrange

double a = 1;

double b = 1;

Ring myring = new Ring();

Ring.Element second = new Ring.Element();

bool expected = true;

//Act

myring.Cur.number = a;

second.number = b;

bool actual = myring.EaseEquals(second);

//Assert

Assert.AreEqual(expected, actual,"Numbers are equals");

}

[TestMethod]

//сравнение чисел текущего(Cur) элемента кольца и другого элемента того же класса Element при неравных значениях;

public void EaseEquals\_With\_Different\_Number\_Are\_Not\_Numbers\_Equals()

{

//Arrange

double a = 2;

double b = 7;

Ring myring = new Ring();

Ring.Element second = new Ring.Element();

bool expected = false;

//Act

myring.Cur.number = a;

second.number = b;

bool actual = myring.EaseEquals(second);

//Assert

Assert.AreEqual(expected, actual, "Numbers are not equals");

}

[TestMethod]

//проверка на правильность записи элемента в кольцо

public void Add\_To\_Ring\_Are\_added()

{

//Arrange

double a = 3;

Ring myring = new Ring();

//Act

myring.Add(a);

//Assert

Assert.AreEqual(a, myring.Cur.number, "Adding works right");

}

[TestMethod]

//проверка на правильность перемещения указателя по часовой стрелке

public void Right\_Current\_becomes\_next\_Is\_next()

{

//Arrange

Ring.Element first = new Ring.Element();

Ring.Element second = new Ring.Element();

Ring.Element third = new Ring.Element();

Ring myring = new Ring();

myring.Cur = third;

myring.Cur.next = first;

first.next = second;

second.next = myring.Cur;

//Act

myring.Right();

//Assert

Assert.AreEqual(first, myring.Cur, "Current pointer has moved to the right");

}

[TestMethod]

//проверка на правильность перемещения указателя против часовой стрелки

public void Left\_Current\_becomes\_previous\_Is\_previous()

{

//Arrange

Ring.Element first = new Ring.Element();

Ring.Element second = new Ring.Element();

Ring.Element third = new Ring.Element();

Ring myring = new Ring();

myring.Cur = third;

myring.Cur.previous = second;

second.previous = first;

first.previous = third;

//Act

myring.Left();

//Assert

Assert.AreEqual(second, myring.Cur, "Current pointer has moved to the left");

}

}

}