//r1

[TestClass]

public class UnitTestPoint

{

[TestMethod]

[ExpectedException(typeof(ArgumentOutOfRangeException))]

public void TestMethodCtor\_ThrowsArgumentOutOfRangeException()

{

Point p;

double a = -1000.0;

double b = -1000.0;

p = new Point(a, b);

}

}

public class Point

{

public Point(double lat, double longit)

{

}

}

//g1

using System;

namespace GeographicLibrary

{

public class Point

{

public Point(double lat, double longit)

{

throw new ArgumentOutOfRangeException();

}

}

}

//r2

[TestMethod]

public void TestMethodCtor\_CteatePoin()

{

Point p;

double a = 0.0;

double b = 0.0;

p = new Point(a, b);

Assert.IsNotNull(p);

}

//g2

double lat;

double longit;

public Point(double lat, double longit)

{

if (Math.Abs(lat) > 90 || Math.Abs(longit) > 180)

{

throw new ArgumentOutOfRangeException();

}

this.lat = lat;

this.longit = longit;

}

//ref2

using System;

namespace GeographicLibrary

{

public class Point

{

double latitude;

double longitude;

public double Longitude

{

get => longitude;

set

{

if (Math.Abs(value) > 180)

{

throw new ArgumentOutOfRangeException();

}

longitude = value;

}

}

public double Latitude

{

get => latitude;

set

{

if (Math.Abs(value) > 90)

{

throw new ArgumentOutOfRangeException();

}

latitude = value;

}

}

public Point(double latitude, double longitude)

{

if (Math.Abs(latitude) > 90 || Math.Abs(longitude) > 180)

{

throw new ArgumentOutOfRangeException();

}

this.Latitude = latitude;

this.Longitude = longitude;

}

}

}