Contents

[**1.** **Tạo project C# xác định ví trí tương đối giữa một điểm và đường tròn:** 2](#_Toc165547594)

[**1.1.** **Đặc tả các chức năng** 2](#_Toc165547595)

[**1.1.1.** **Chức năng tính khoảng cách giữa hai điểm** 2](#_Toc165547596)

[**1.1.2.** **Chức năng kiểm tra vị trí giữa một điểm và đường tròn** 2](#_Toc165547597)

[**1.2.** **Tạo project C#** 2](#_Toc165547598)

[**1.2.1.** **Tạo các lớp cần thiết** 2](#_Toc165547599)

[**1.2.2.** **Chức năng 1: Tính khoảng cách giữa hai điểm** 10](#_Toc165547600)

[**1.2.3.** **Chức năng 2: Kiểm tra vị trí tương đối giữa điểm và đường tròn** 10](#_Toc165547601)

[**1.2.4.** **Chạy project** 12](#_Toc165547602)

[**2.** **Chuẩn bị để kiểm thử** 13](#_Toc165547603)

[**2.1.** **Tạo project** 13](#_Toc165547604)

[**2.1.1.** **Kiểm thử chức năng tính khoảng cách giữa hai điểm(CN1)** 18](#_Toc165547605)

[**2.1.2.** **Kiểm thử chức năng kiểm tra vị trí giữa điểm và đường tròn(CN2)** 18](#_Toc165547606)

[**2.1.3.** **Kiểm thử ngoại lệ không phải đường tròn** 19](#_Toc165547607)

[**2.2.** **Bảng quyết định cho các chức năng** 19](#_Toc165547608)

[**2.2.1.** **Chức năng tính khoảng cách giữa hai điểm(CN1)** 19](#_Toc165547609)

[**2.2.2.** **Chức năng kiểm tra vị trí tương đối giữa một điểm và đường tròn(CN2)** 20](#_Toc165547610)

[**2.3.** **Chạy các tests:** 21](#_Toc165547611)

**CHƯƠNG 1: NUNIT**

1. **Tạo project C# xác định ví trí tương đối giữa một điểm và đường tròn:**
   1. **Đặc tả các chức năng**
      1. **Chức năng tính khoảng cách giữa hai điểm**

* Mô tả: Chức năng này cho phép tính toán khoảng cách giữa hai điểm trong không gian hai chiều dựa trên tọa độ của chúng
* Đầu vào (input): Tọa độ của hai điểm lần lượt là M1(x1, y1) và M2(x2, y2). Các giá trị x1, y1, x2, y2 là các số thực
* Đầu ra (output): Khoảng cách giữa hai điểm, là một số thực
* Thuật toán: *d =* 
  + 1. **Chức năng kiểm tra vị trí giữa một điểm và đường tròn**
* Mô tả: Chức năng này nhằm mục đích xác định vị trí tương đối của một điểm với đường tròn (xem điểm đó nằm bên trong, trên hay ngoài đường tròn)
* Đầu vào (input): Tọa độ của một điểm M(Mx, My) và một đường tròn (bao gồm tọa độ của một điểm là tâm đường tròn O(Ox, Oy) và bán kính R). Các giá trị Mx, My, Ox, Oy, R là các số thực
* Thuật toán:
* Đầu tiên, tính khoảng cách từ tâm đường tròn O(Ox, Oy) đến điểm M(Mx, My) bằng công thức *d =*
* So sánh khoảng cách *d* với bán kính R của đường tròn:

+ Nếu *d < R*: Điểm M(Mx, My) nằm **trong** đường tròn

+ Nếu *d > R*: Điểm M(Mx, My) nằm **ngoài** đường tròn

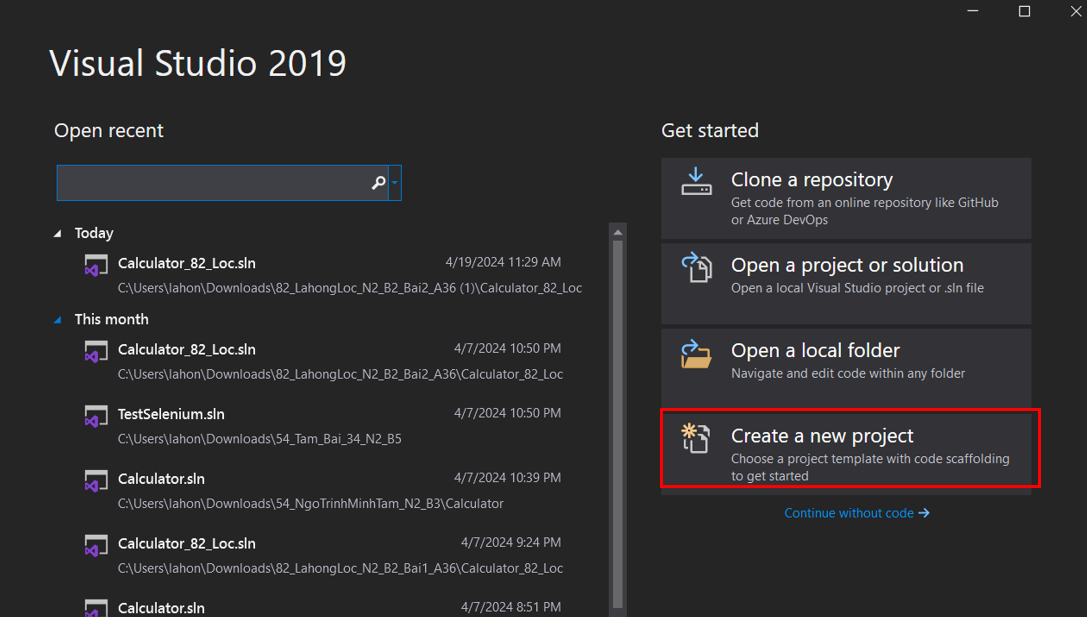
+ Nếu *d = R*: Điểm M(Mx, My) nằm **trên** đường tròn

* Đầu ra: Vị trí của điểm M(Mx, My) với đường tròn tâm O(Ox, Oy) bán kính R (trong, ngoài, trên)
* Ngoại lệ:

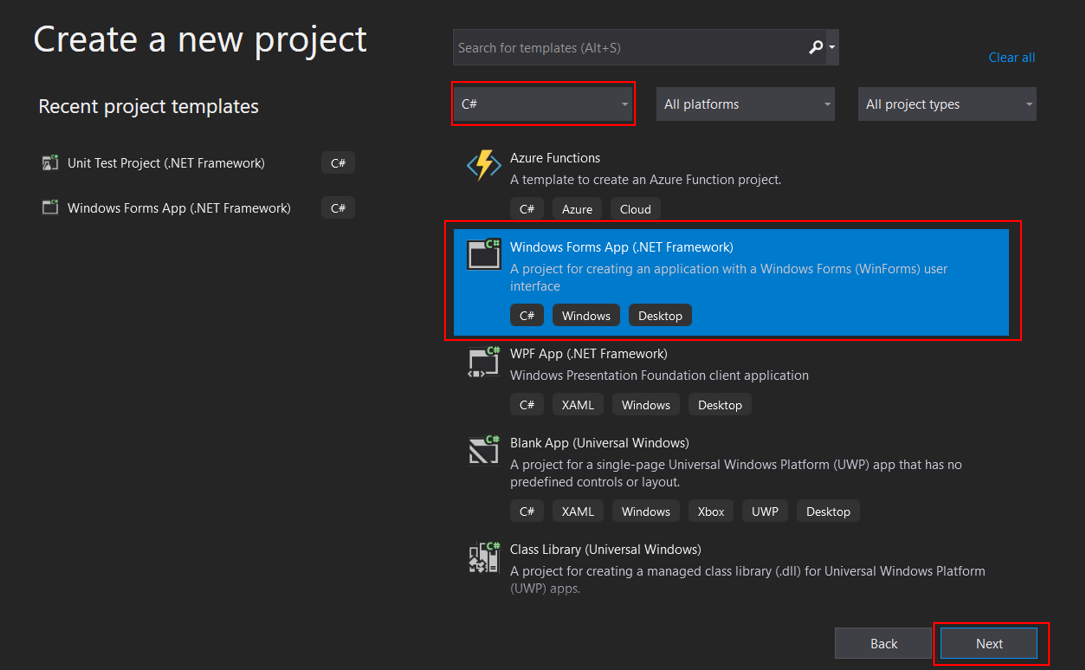
|  |  |
| --- | --- |
| **Tên ngoại lệ** | KhongPhaiDuongTron\_82\_LaHongLoc |
| **Điều kiện gọi** | Khi bán kính đường tròn âm (R < 0) |

* 1. **Tạo project C#**
     1. **Tạo các lớp cần thiết**

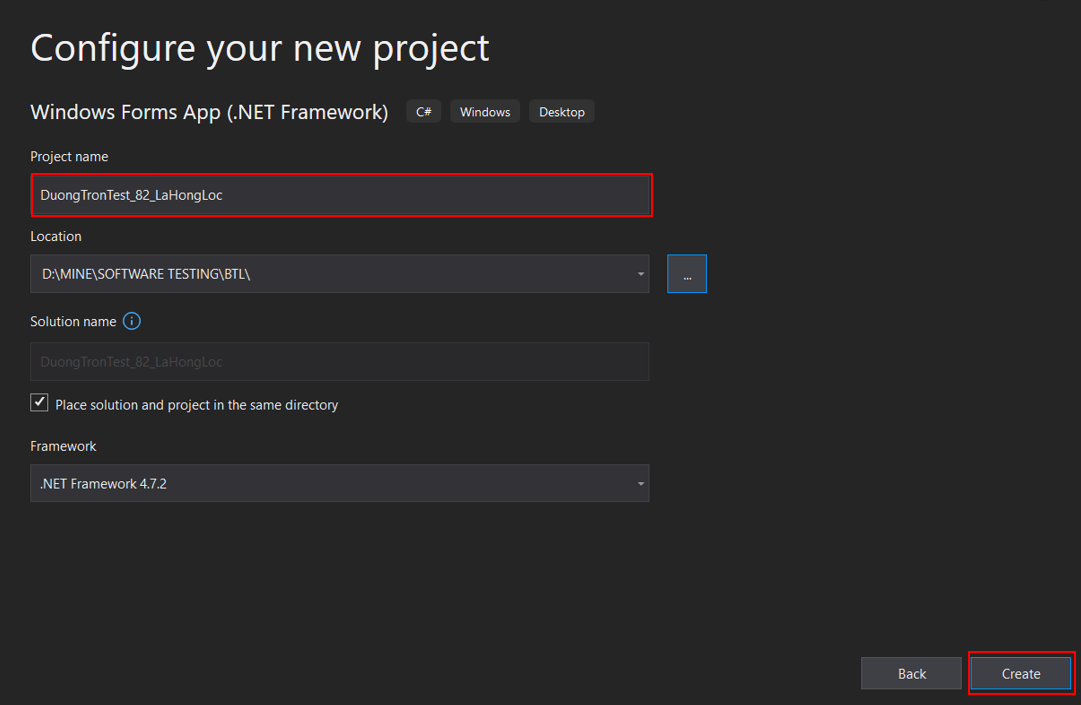
Bước 1: Chọn vào **“Create a new project”**



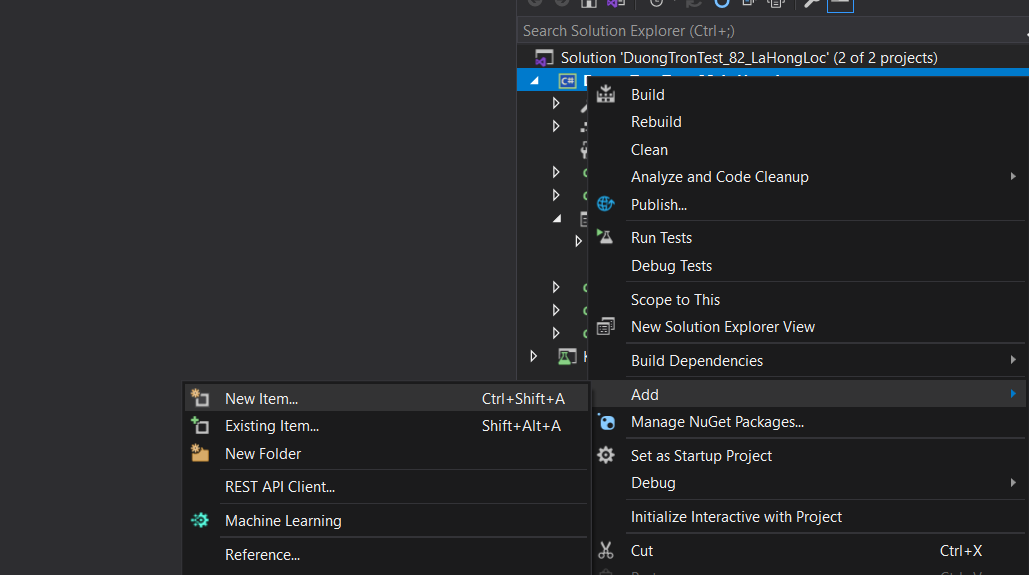
Bước 2: Chọn vào ngôn ngữ C#, chọn **“Windows forms App (.NET framework)”** và nhấn **“Next”**



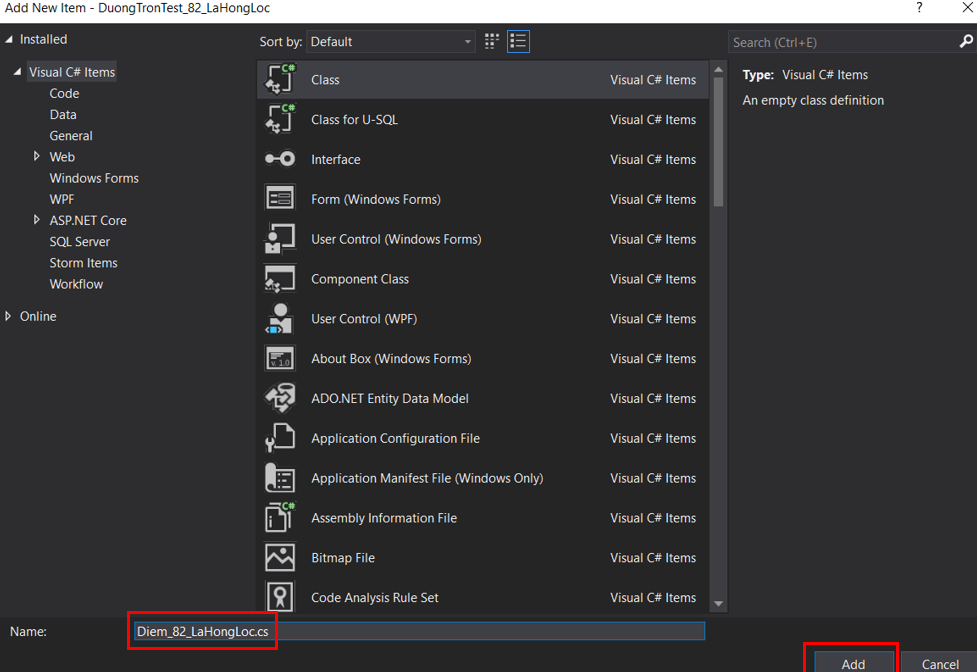
Bước 3: Đặt tên cho project, chọn vị trí thư mục muốn lưu và nhấn **“Create”** để tạo project



Bước 4: Chuột phải vào project, chọn **“Add”** và **“New Item”** để thêm các class cần thiết



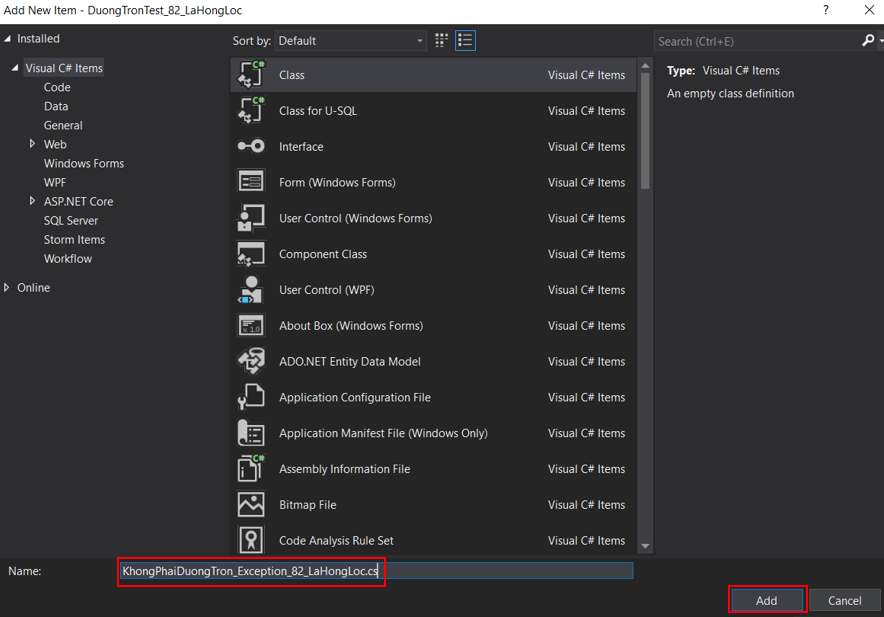
Bước 5: Thêm class **“Diem\_82\_LaHongLoc.cs”** vào project



* Tiếp theo, thêm các thuộc tính và phương thức khởi tạo cho class

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thuộc kiểu** | **Ý nghĩa** |
| x\_82\_LaHongLoc | double | Hoành độ của điểm |
| y\_82\_LaHongLoc | double | Tung độ của điểm |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DuongTronTest\_82\_LaHongLoc  {  public class Diem\_82\_LaHongLoc  {  public double x\_82\_LaHongLoc { get; set; }  public double y\_82\_LaHongLoc { get; set; }  public Diem\_82\_LaHongLoc(double x, double y)  {  this.x\_82\_LaHongLoc = x;  this.y\_82\_LaHongLoc = y;  }  }  } | | |

Bước 6: Tương tự, thêm class **“KhongPhaiDuongTron\_Exception\_82\_LaHongLoc.cs”**



* Class này là một **Exception** cảnh báo khi **đầu vào** của một đường tròn **không hợp lệ**. Nên nó sẽ kế thừa lớp **“Exception”**
* Sau đó, tiến hành định nghĩa các phương thức khởi tạo

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DuongTronTest\_82\_LaHongLoc

{

public class KhongPhaiDuongTron\_Exception\_82\_LaHongLoc : Exception

{

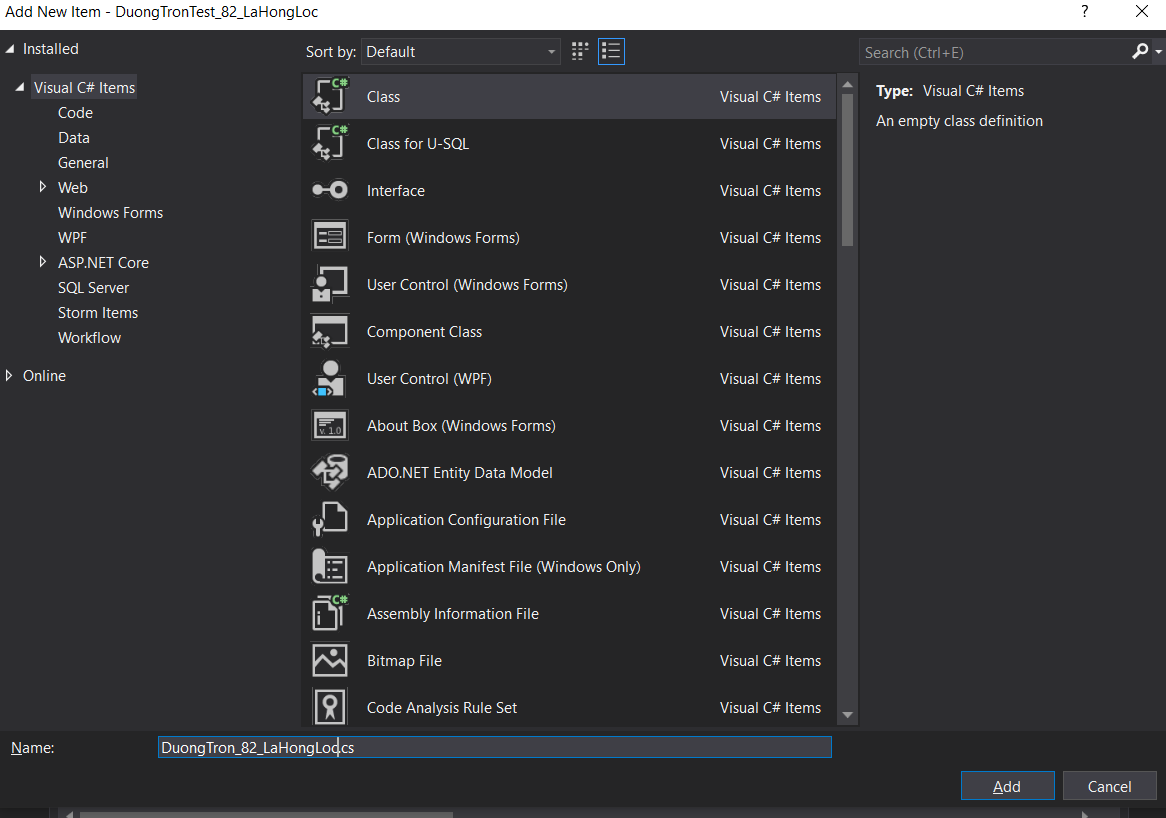
public KhongPhaiDuongTron\_Exception\_82\_LaHongLoc() : base("Đây không phải đường tròn(R < 0)!") { }

public KhongPhaiDuongTron\_Exception\_82\_LaHongLoc(string message) : base(message) { }

}

}

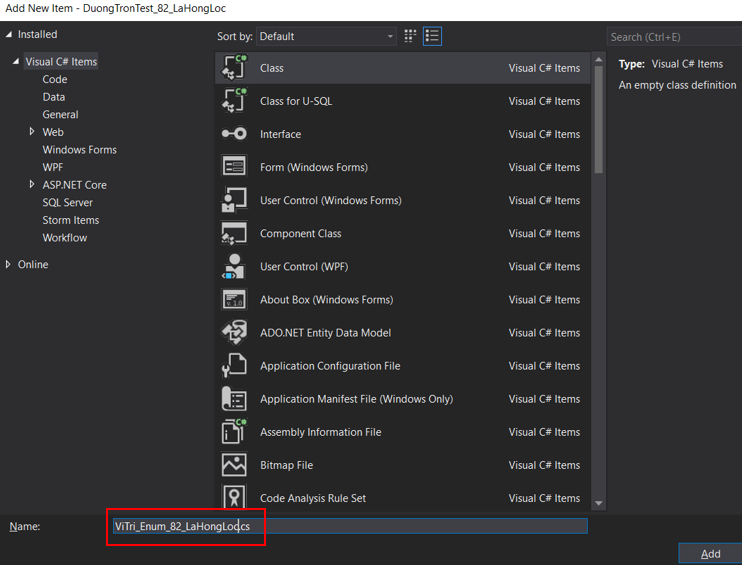
Bước 7: Thêm class **“DuongTron\_82\_LaHongLoc.cs”**



* Class này dùng để định nghĩa một đường tròn(gồm 1 điểm là tâm, và bán kính)
* Sau đó, thêm các thuộc tính, và phương thức khởi tạo cho class (ném ngoại lệ **“KhongPhaiDuongTron\_Exception\_82\_LaHongLoc”** khi đầu vào của các tham số trong phương thức khởi tạo không hợp lệ (bán kính âm))

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thuộc kiểu** | **Ý nghĩa** |
| TrungDiem\_82\_LaHongLoc | Diem\_82\_LaHongLoc | Trung điểm đường tròn |
| BanKinh\_82\_LaHongLoc | double | Bán kính đường tròn |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DuongTronTest\_82\_LaHongLoc  {  public class DuongTron\_82\_LaHongLoc  {  public Diem\_82\_LaHongLoc TrungDiem\_82\_LaHongLoc { get; set; }  public double BanKinh\_82\_LaHongLoc { get; set; }  public DuongTron\_82\_LaHongLoc(Diem\_82\_LaHongLoc O, double R)  {  if (R >= 0)  {  this.TrungDiem\_82\_LaHongLoc = O;  this.BanKinh\_82\_LaHongLoc = R;  }  else throw new KhongPhaiDuongTron\_Exception\_82\_LaHongLoc();  }  }  } | | |

Bước 8: Thêm Enum **“ViTri\_Enum\_82\_LaHongLoc.cs”.** Enum này bao gồm các vị trí tương đối giữa một điểm và đường tròn(trong, ngoài, trên)



* Thêm các giá trị cho Enum

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DuongTronTest\_82\_LaHongLoc

{

public enum ViTri\_Enum\_82\_LaHongLoc

{

Trong\_82\_LaHongLoc,

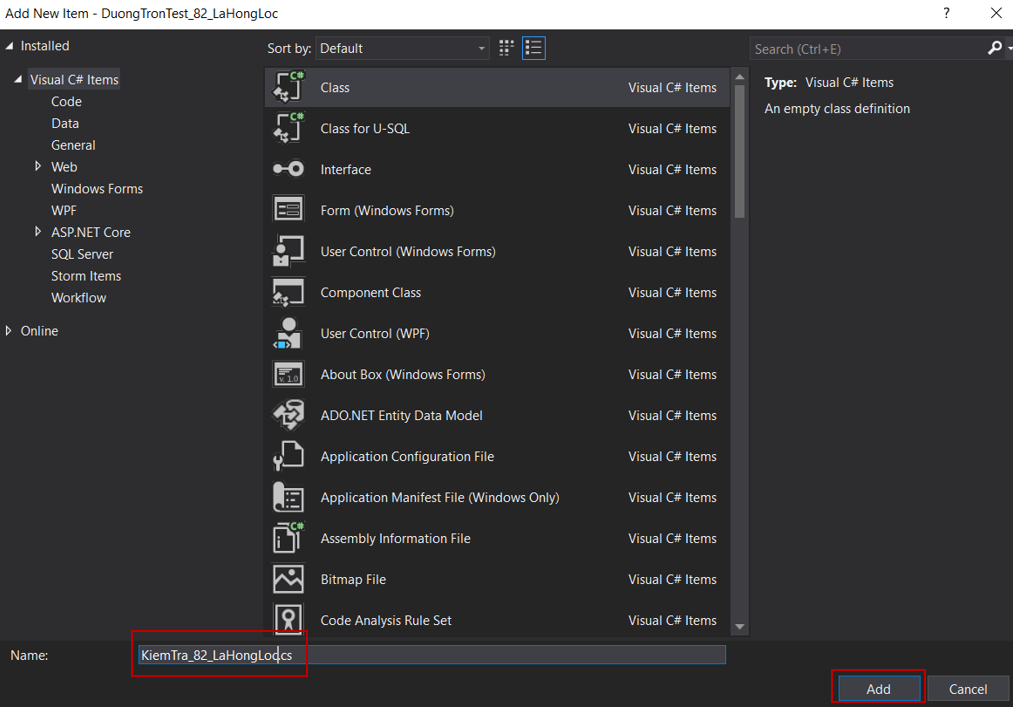
Tren\_82\_LaHongLoc,

Ngoai\_82\_LaHongLoc

}

}

Bước 9: Thêm class **“KiemTra\_82\_LaHongLoc.cs”**. Class này để kiểm tra vị trí tương đối của một điểm và đường tròn



* Thêm các thuộc tính và phương thức khởi tạo

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thuộc kiểu** | **Ý nghĩa** |
| DuongTron\_82\_LaHongLoc | DuongTron\_82\_LaHongLoc | Là 1 đường tròn |
| Diem\_82\_LaHongLoc | Diem\_82\_LaHongLoc | Là 1 điểm |
| public class KiemTra\_82\_LaHongLoc  {  public DuongTron\_82\_LaHongLoc DuongTron\_82\_LaHongLoc;  public Diem\_82\_LaHongLoc Diem\_82\_LaHongLoc;  public KiemTra\_82\_LaHongLoc() { }  public KiemTra\_82\_LaHongLoc(double Ox, double Oy, double R, double Mx, double My)  {  this.DuongTron\_82\_LaHongLoc = new DuongTron\_82\_LaHongLoc(new Diem\_82\_LaHongLoc(Ox, Oy), R);  this.Diem\_82\_LaHongLoc = new Diem\_82\_LaHongLoc(Mx, My);  } | | |

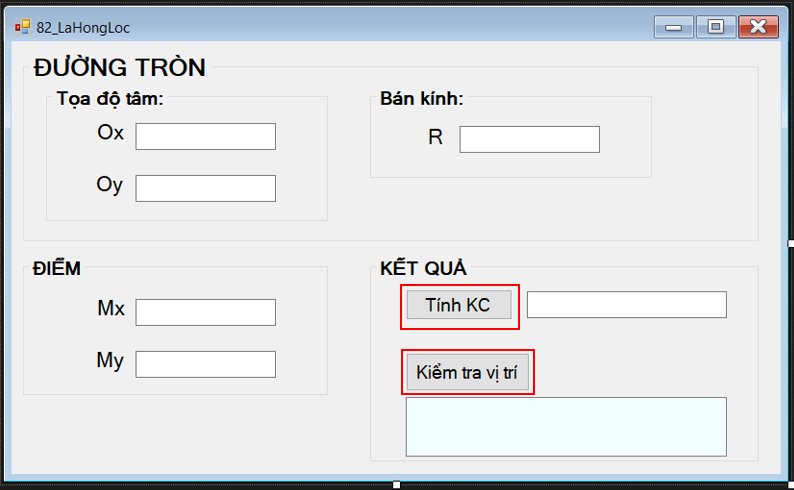
* Sau khi có những lớp cần thiết, ta tiến hành vào các chức năng chính của project
  + 1. **Chức năng 1: Tính khoảng cách giữa hai điểm**
* Định nghĩa phương thức **“TinhKhoangCach\_82\_LaHongLoc.cs”**. Phương thức này sẽ tính khoảng cách giữa 2 điểm dựa trên tọa độ của 2 điểm đó

|  |  |  |  |
| --- | --- | --- | --- |
| **Phương thức** | **Tham số đầu vào** | **Kiểu dữ liệu trả về** | **Ý nghĩa** |
| TinhKhoangCach\_82\_LaHongLoc | d1\_82\_LaHongLoc, d2\_82\_LaHongLoc | double | Tính khoảng cách giữa 2 điểm |
| **(thuộc kiểu Diem\_82\_LaHongLoc)** |
| public static double TinhKhoangCach\_82\_LaHongLoc(Diem\_82\_LaHongLoc d1\_82\_LaHongLoc, Diem\_82\_LaHongLoc d2\_82\_LaHongLoc)  {  double KhoangCach\_82\_LaHongLoc = Math.Round(  Math.Sqrt(Math.Pow(d1\_82\_LaHongLoc.x\_82\_LaHongLoc - d2\_82\_LaHongLoc.x\_82\_LaHongLoc, 2) +  Math.Pow(d1\_82\_LaHongLoc.y\_82\_LaHongLoc - d2\_82\_LaHongLoc.y\_82\_LaHongLoc, 2)), 2);  return KhoangCach\_82\_LaHongLoc;  } | | | |

* + 1. **Chức năng 2: Kiểm tra vị trí tương đối giữa điểm và đường tròn**
* Tiếp theo, định nghĩa phương thức **“KiemTraViTri\_82\_LaHongLoc”.** Kết quả trả về là vị trí tương đối giữa điểm và đường tròn đó (thuộc kiểu **ViTri\_Enum\_82\_LaHongLoc)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phương thức** | **Tham số đầu vào** | **Kiểu dữ liệu trả về** | **Ý nghĩa** |
| KiemTraViTri\_82\_LaHongLoc | - | ViTri\_Enum\_82\_LaHongLoc | Kiểm tra vị trí tương đối giữa điểm và đường tròn |
| public ViTri\_Enum\_82\_LaHongLoc KiemTraViTri\_82\_LaHongLoc()  {  double KhoangCach\_82\_LaHongLoc =  KiemTra\_82\_LaHongLoc.TinhKhoangCach\_82\_LaHongLoc(this.Diem\_82\_LaHongLoc,  this.DuongTron\_82\_LaHongLoc.TrungDiem\_82\_LaHongLoc);  if (KhoangCach\_82\_LaHongLoc < this.DuongTron\_82\_LaHongLoc.BanKinh\_82\_LaHongLoc)  return ViTri\_Enum\_82\_LaHongLoc.Trong\_82\_LaHongLoc;  else if (KhoangCach\_82\_LaHongLoc > this.DuongTron\_82\_LaHongLoc.BanKinh\_82\_LaHongLoc)  return ViTri\_Enum\_82\_LaHongLoc.Ngoai\_82\_LaHongLoc;  else return ViTri\_Enum\_82\_LaHongLoc.Tren\_82\_LaHongLoc;  } | | | |

* Thiết kế giao diện cho windows forms như sau. Sau đó xử lý sự kiện khi click vào button **“btn\_tinhKC\_82\_LaHongLoc”** và button **“btn\_kiemTra\_82\_LaHongLoc”**



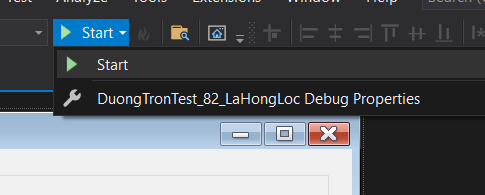
* Code xử lý sự kiện khi click bào button **“btn\_tinhKC\_82\_LaHongLoc”**

|  |
| --- |
| private void btn\_tinhKC\_82\_LaHongLoc\_Click(object sender, EventArgs e)  {  try  {  double Ox\_82\_LaHongLoc = double.Parse(txt\_Ox\_82\_LaHongLoc.Text);  double Oy\_82\_LaHongLoc = double.Parse(txt\_Oy\_82\_LaHongLoc.Text);  Diem\_82\_LaHongLoc O\_82\_LaHongLoc = new Diem\_82\_LaHongLoc(Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc);  double Mx\_82\_LaHongLoc = double.Parse(txt\_Mx\_82\_LaHongLoc.Text);  double My\_82\_LaHongLoc = double.Parse(txt\_My\_82\_LaHongLoc.Text);  Diem\_82\_LaHongLoc M\_82\_LaHongLoc = new Diem\_82\_LaHongLoc(Mx\_82\_LaHongLoc, My\_82\_LaHongLoc);  txt\_PT\_82\_LaHongLoc.Text =  KiemTra\_82\_LaHongLoc.TinhKhoangCach\_82\_LaHongLoc(M\_82\_LaHongLoc, O\_82\_LaHongLoc).ToString();  }  catch (Exception ex)  {  txt\_PT\_82\_LaHongLoc.Text = ex.Message.ToString();  }  } |

* Code xử lý sự kiện khi click vào button **“btn\_kiemTra\_82\_LaHongLoc”**

|  |
| --- |
| private void btn\_kiemTra\_82\_LaHongLoc\_Click(object sender, EventArgs e)  {  try  {  double Ox\_82\_LaHongLoc = double.Parse(txt\_Ox\_82\_LaHongLoc.Text);  double Oy\_82\_LaHongLoc = double.Parse(txt\_Oy\_82\_LaHongLoc.Text);  double R\_82\_LaHongLoc = double.Parse(txt\_R\_82\_LaHongLoc.Text);  double Mx\_82\_LaHongLoc = double.Parse(txt\_Mx\_82\_LaHongLoc.Text);  double My\_82\_LaHongLoc = double.Parse(txt\_My\_82\_LaHongLoc.Text);  KiemTra\_82\_LaHongLoc KiemTra =  new KiemTra\_82\_LaHongLoc(Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc,  R\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc);  txt\_Kq\_82\_LaHongLoc.Text = KiemTra.KiemTraViTri\_82\_LaHongLoc().ToString();  }  catch (Exception ex)  {  txt\_Kq\_82\_LaHongLoc.Text = ex.Message.ToString();  }  } |

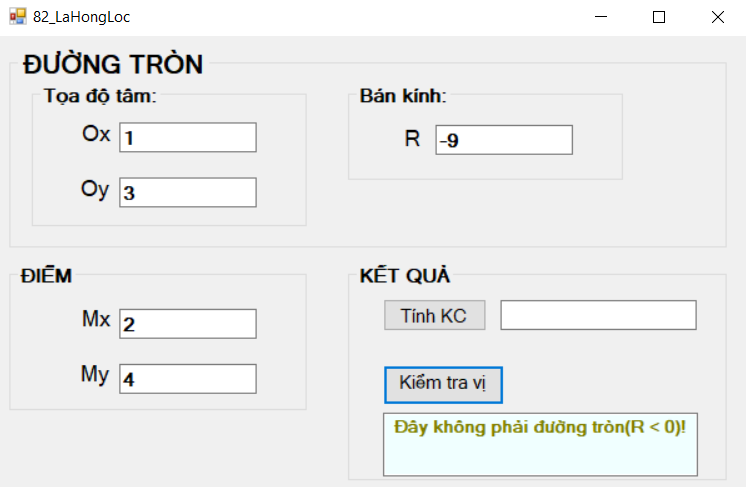
* + 1. **Chạy project**
* Bấm vào **start** để chạy winforms, nhập các input và xem các kết quả thu được



* Các kết quả thu được

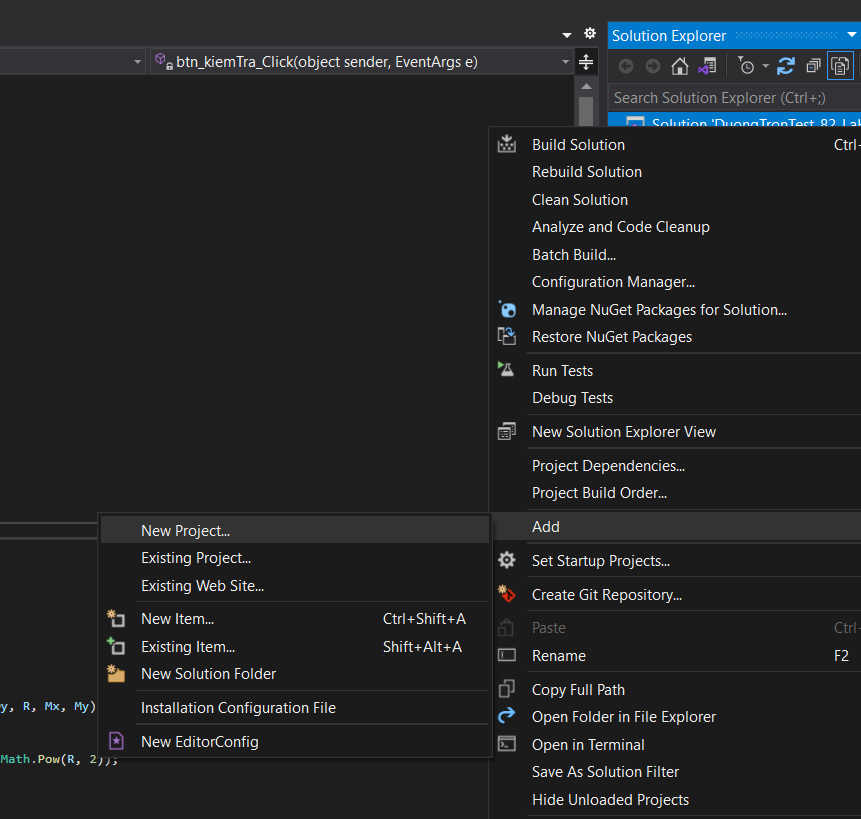
|  |  |  |
| --- | --- | --- |
|  |  |  |

* Các kết quả ngoại lệ nếu bán kính âm (R<0)

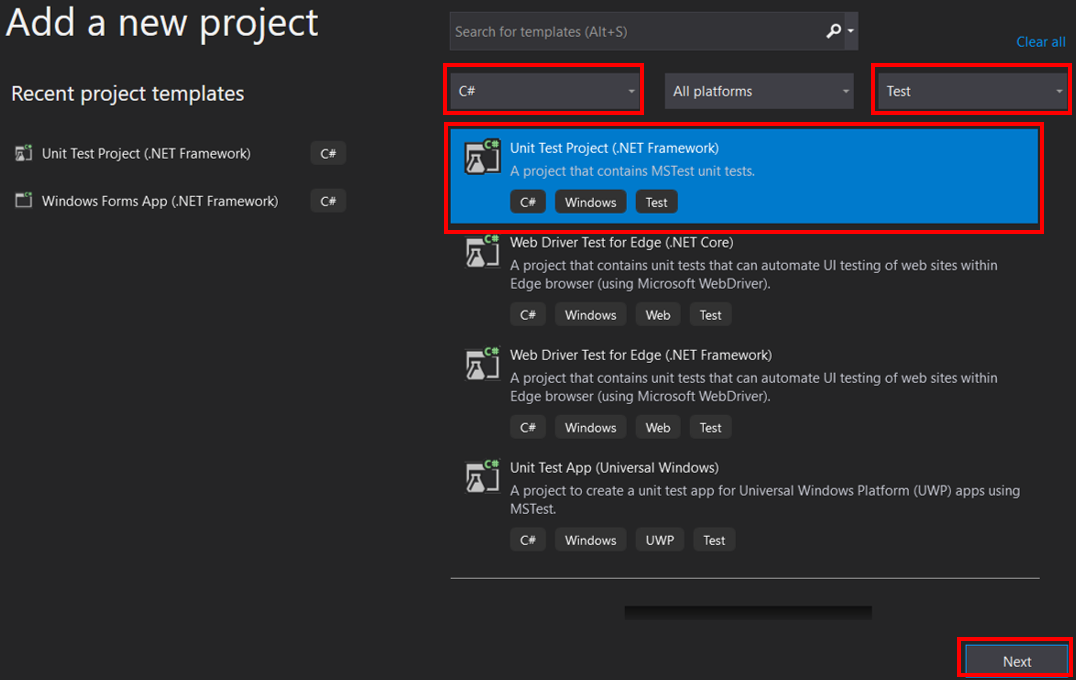


1. **Chuẩn bị để kiểm thử**
   1. **Tạo project**

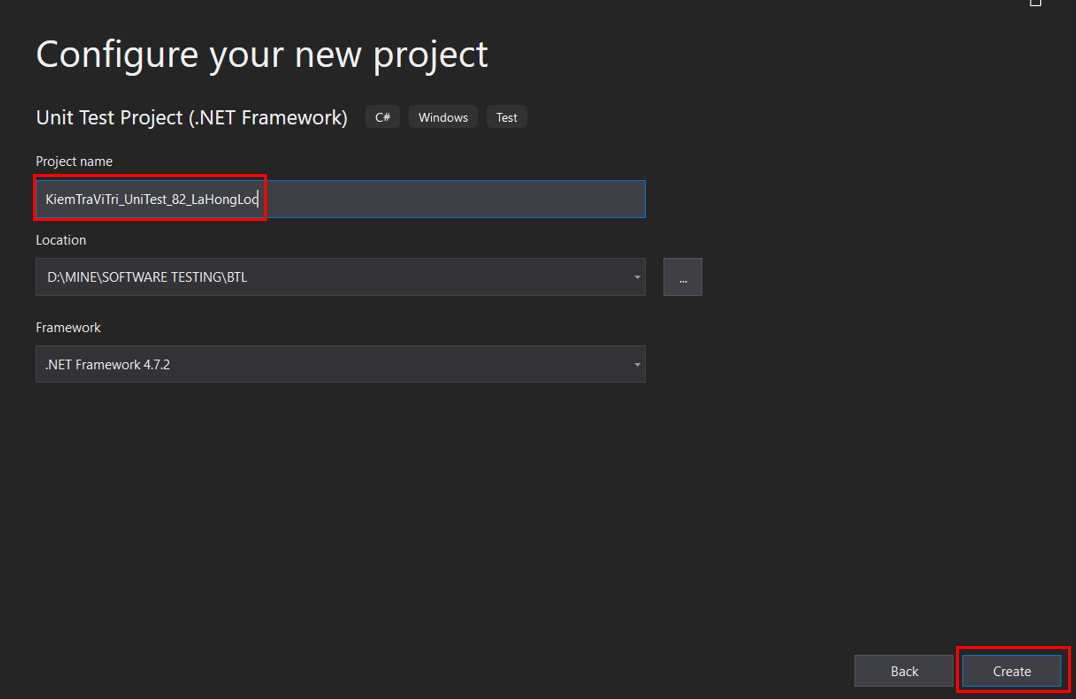
Bước 1: Chuột phải vào Solution **“DuongTronTest\_82\_LaHongLoc”**, chọn **“Add”** và **“New Project…”**



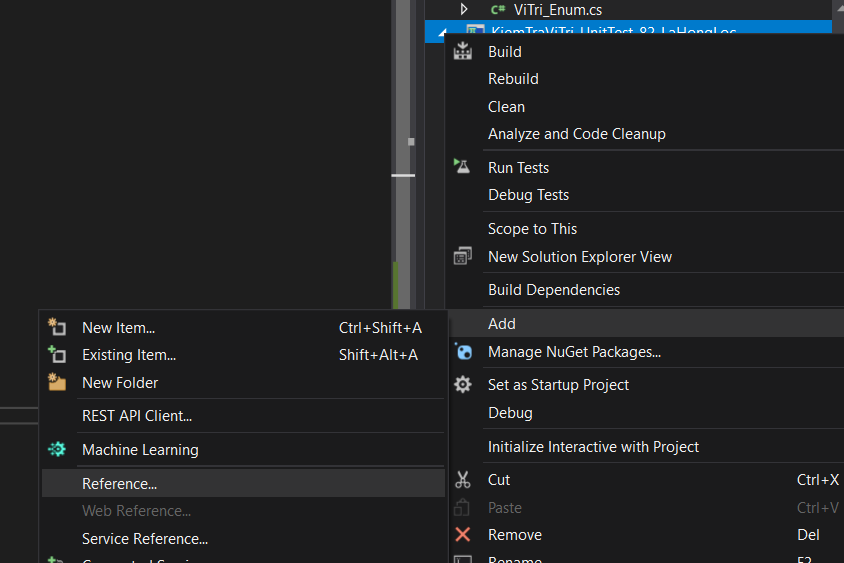
* Tiếp theo, chọn ngôn ngữ C#, là loại Test. Sau đó chọn **“Unit Test Project (.NET Framework)”**, bấm **“Next”** để tiếp tục



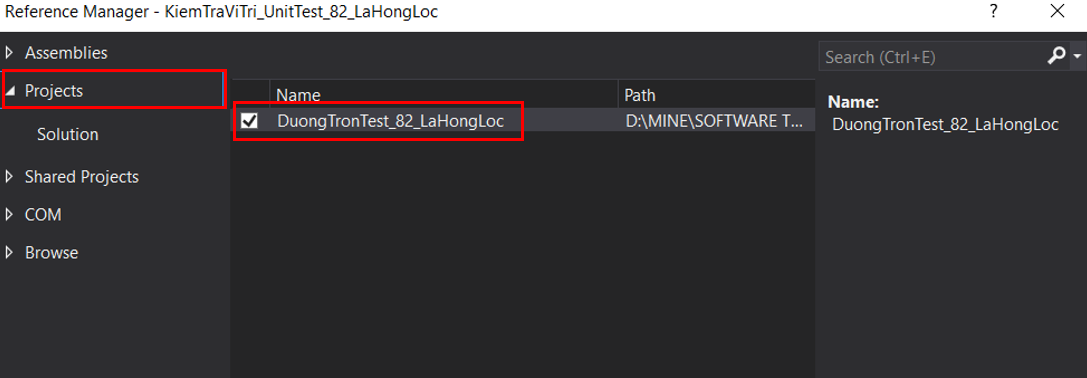
* Sau đó, đặt tên cho project **“KiemTraViTri\_UnitTest\_82\_LaHongLoc”**, và chọn vị trí thư mục muốn lưu và nhấn vào **“Create”** để tạo project



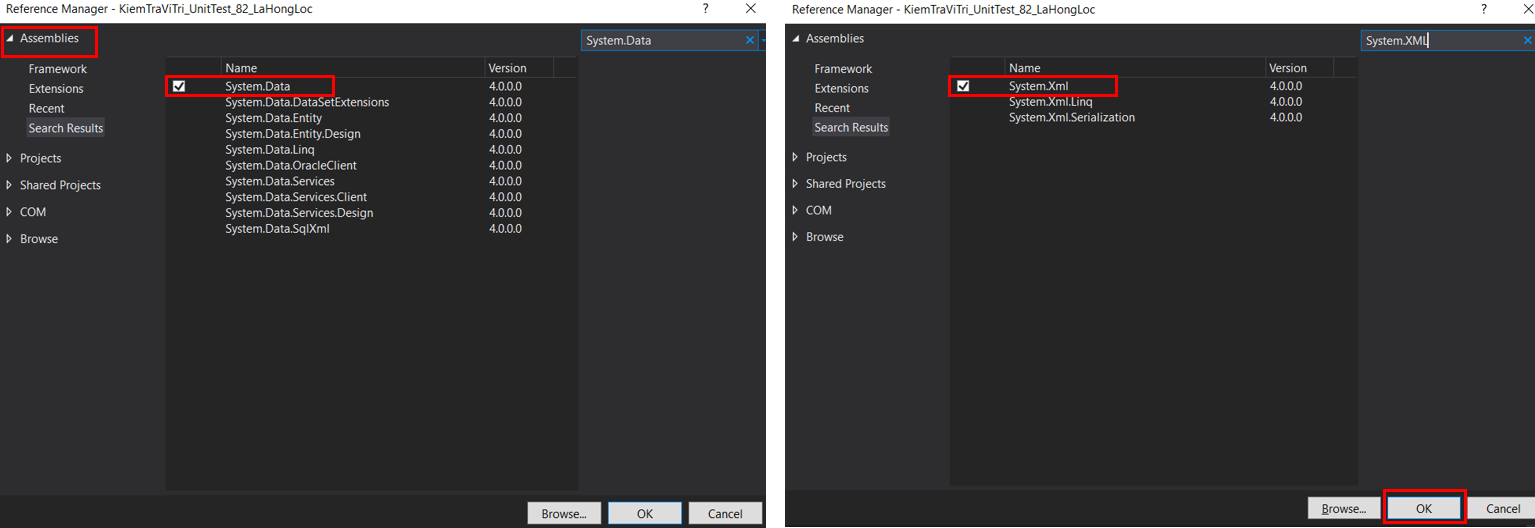
Bước 2: Chuột phải vào Unit test project vừa tạo, chọn **“Add”**, và chọn **“Reference…”** để thêm reference cho project



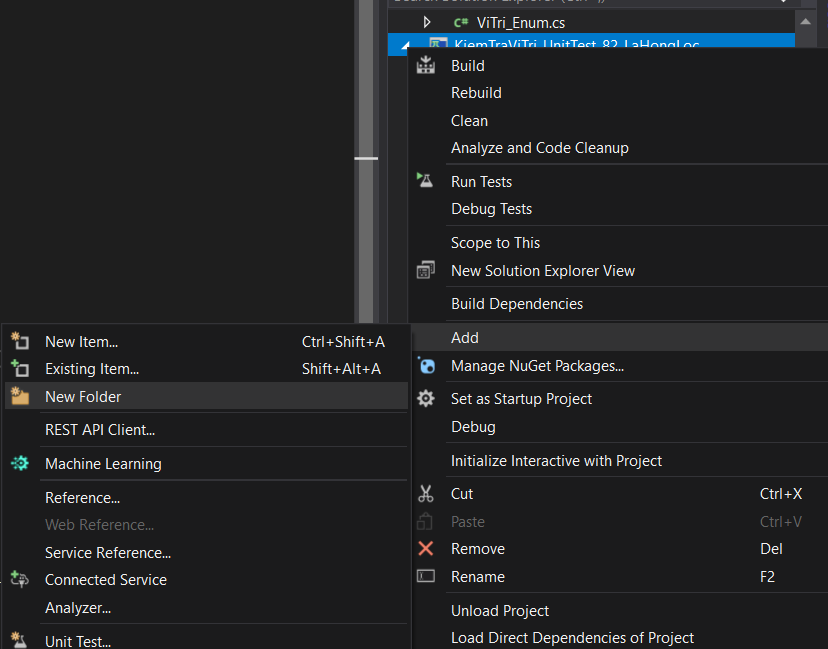
* Ở cửa sổ **“Reference Manager”**, chọn vào mục **“Projects”** và tick chọn project muốn test: **DuongTronTest\_82\_LaHongLoc**



* Tiếp theo, bấm vào mục **“Assembles”**, trên thanh Search tiến hành tìm kiếm **“System.Data”, “System.Xml**”*(hỗ trợ kết nối với DataSource để đọc file \*.csv),* tick chọn và bấm **“OK”** để thêm các reference cần thiết



Bước 3: Thêm folder **“Data”**



* Tạo các file **“TinhKhoangCachData.cvs, KiemTraVTData.csv, KhongPhaiDuongTronDât.csv”** trong thư mục **“Data”** có nội dung như bên dưới
* Ở **“File Properties”** của file **“\*.csv”**, ở mục **“Copy to Output Directory”** chọn thành **“Copy always”**

|  |  |
| --- | --- |
|  |  |

Bước 4: Định nghĩa các thuộc tính cần thiết và phương thức SetUp()

|  |
| --- |
| public class UnitTest1  {  private KiemTra\_82\_LaHongLoc KiemTra\_82\_LaHongLoc;  public TestContext TestContext { get; set; }  [TestInitialize]  public void SetUp()  {  KiemTra\_82\_LaHongLoc = new KiemTra\_82\_LaHongLoc();  }  } |

* + 1. **Kiểm thử chức năng tính khoảng cách giữa hai điểm(CN1)**
* Viết hàm kiểm thử chức năng tính khoảng cách giữa hai điểm **(TinhKhoangCach\_82\_LaHongLoc)** bằng cách đọc các testcase từ file **“Data.csv”**

|  |
| --- |
| [TestMethod]  [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",  @".\Data\TinhKhoangCachData.csv", "TinhKhoangCachData#csv", DataAccessMethod.Sequential)]  public void TinhKhoangCach\_82\_LaHongLoc\_Test()  {  double Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc;  double expected, actual;  Ox\_82\_LaHongLoc = double.Parse(TestContext.DataRow[0].ToString());  Oy\_82\_LaHongLoc = double.Parse(TestContext.DataRow[1].ToString());  Diem\_82\_LaHongLoc O\_82\_LaHongLoc = new Diem\_82\_LaHongLoc(Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc);  Mx\_82\_LaHongLoc = double.Parse(TestContext.DataRow[2].ToString());  My\_82\_LaHongLoc = double.Parse(TestContext.DataRow[3].ToString());  Diem\_82\_LaHongLoc M\_82\_LaHongLoc = new Diem\_82\_LaHongLoc(Mx\_82\_LaHongLoc, My\_82\_LaHongLoc);  expected = double.Parse(TestContext.DataRow[4].ToString());  actual = KiemTra\_82\_LaHongLoc.TinhKhoangCach\_82\_LaHongLoc(M\_82\_LaHongLoc, O\_82\_LaHongLoc);  Assert.AreEqual(expected, actual);  } |

* + 1. **Kiểm thử chức năng kiểm tra vị trí giữa điểm và đường tròn(CN2)**
* Tiếp theo, viết hàm kiểm thử chức năng kiểm tra vị trí giữa điểm và đường tròn **(KiemTraViTri\_82\_LaHongLoc)** bằng cách đọc các testcases từ file **“Data.csv”**

|  |
| --- |
| [TestMethod]  [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",  @".\Data\KiemTraVTData.csv", "KiemTraVTData#csv", DataAccessMethod.Sequential)]  public void KiemTraViTri\_Test\_82\_LaHongLoc()  {  double Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc, R\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc;  ViTri\_Enum\_82\_LaHongLoc expected, actual;  Ox\_82\_LaHongLoc = double.Parse(TestContext.DataRow[0].ToString());  Oy\_82\_LaHongLoc = double.Parse(TestContext.DataRow[1].ToString());  R\_82\_LaHongLoc = double.Parse(TestContext.DataRow[2].ToString());  Mx\_82\_LaHongLoc = double.Parse(TestContext.DataRow[3].ToString());  My\_82\_LaHongLoc = double.Parse(TestContext.DataRow[4].ToString());  expected = (ViTri\_Enum\_82\_LaHongLoc)Enum.Parse(typeof(ViTri\_Enum\_82\_LaHongLoc), TestContext.DataRow[5].ToString());  KiemTra\_82\_LaHongLoc =  new KiemTra\_82\_LaHongLoc(Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc, R\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc);  actual = KiemTra\_82\_LaHongLoc.KiemTraViTri\_82\_LaHongLoc();  Assert.AreEqual(expected, actual);  } |

* + 1. **Kiểm thử ngoại lệ không phải đường tròn**
* Sau đó, tiếp tục viết hàm kiểm thử cho ngoại lệ **“KhongPhaiDuongTron\_Exception\_82\_LaHongLoc”** của chức năng “**KiemTraViTri\_82\_LaHongLoc”**

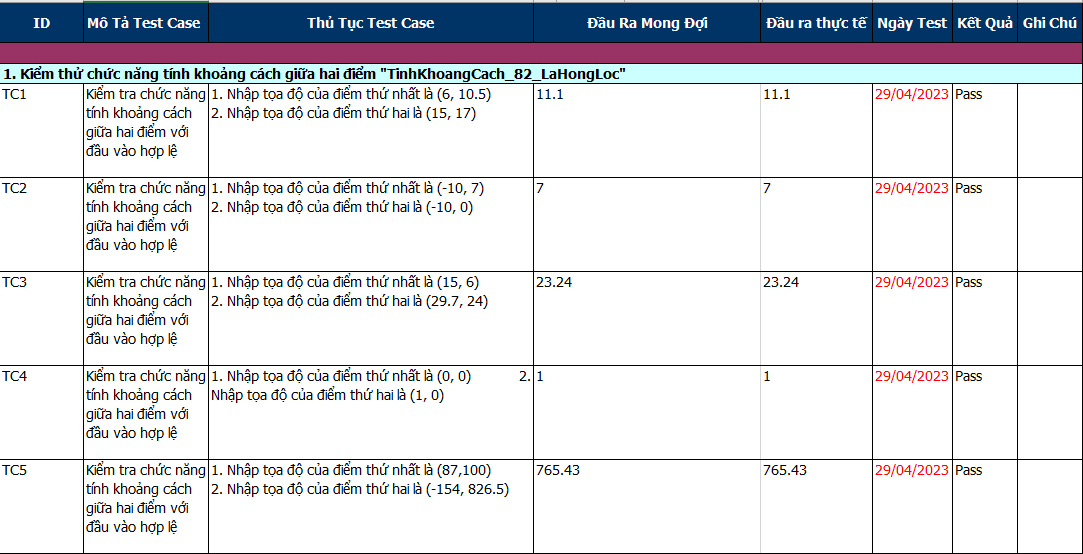
|  |
| --- |
| [TestMethod]  [ExpectedException(typeof(KhongPhaiDuongTron\_Exception\_82\_LaHongLoc))]  [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",  @".\Data\KhongPhaiDuongTronExData.csv", "KhongPhaiDuongTronExData#csv", DataAccessMethod.Sequential)]  public void KiemTraViTriException\_Test\_82\_LaHongLoc()  {  double Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc, R\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc;  Ox\_82\_LaHongLoc = double.Parse(TestContext.DataRow[0].ToString());  Oy\_82\_LaHongLoc = double.Parse(TestContext.DataRow[1].ToString());  R\_82\_LaHongLoc = double.Parse(TestContext.DataRow[2].ToString());  Mx\_82\_LaHongLoc = double.Parse(TestContext.DataRow[3].ToString());  My\_82\_LaHongLoc = double.Parse(TestContext.DataRow[4].ToString());  KiemTra\_82\_LaHongLoc =  new KiemTra\_82\_LaHongLoc(Ox\_82\_LaHongLoc, Oy\_82\_LaHongLoc, R\_82\_LaHongLoc, Mx\_82\_LaHongLoc, My\_82\_LaHongLoc);  ViTri\_Enum\_82\_LaHongLoc VT\_82\_LaHongLoc = KiemTra\_82\_LaHongLoc.KiemTraViTri\_82\_LaHongLoc();  } |

* 1. **Bảng quyết định cho các chức năng**
     1. **Chức năng tính khoảng cách giữa hai điểm(CN1)**
        1. **Bảng quyết định**

|  |  |  |
| --- | --- | --- |
| **CÁC ĐIỀU KIỆN** | x1 | - |
| y1 | - |
| x2 | - |
| y2 | - |
| **CÁC HÀNH ĐỘNG** | Hành động | D |
| -        D: Trả về d (khoảng cách giữa 2 điểm) | | |

* + - 1. **Thiết kế các testcases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Hệ Thống** | Sample Project | | |
| **Module Code：** | CR100 - Export to excel | | |
| **Yêu Cầu Kiểm Thử** | CR1 - | | |
| Pass | 5 | Pending | 0 |
| Fail | 0 | Number of test cases: | 5 |

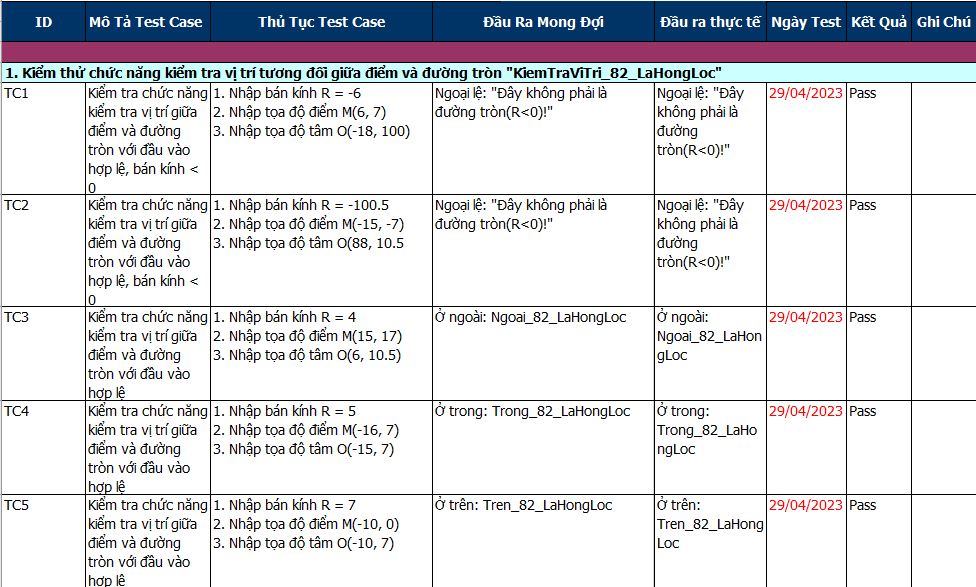
****

* + 1. **Chức năng kiểm tra vị trí tương đối giữa một điểm và đường tròn(CN2)**
       1. **Bảng quyết định**

|  |  |  |  |
| --- | --- | --- | --- |
| **CÁC ĐIỀU KIỆN** | R | <0 | >=0 |
| x1 | - | - |
| y1 | - | - |
| x2 | - | - |
| y2 | - | - |
| **CÁC HÀNH ĐỘNG** | Thông điệp lỗi | E1 | P |
| * E1(KhongPhaiDuongTron\_Exception\_82\_LaHongLoc): Đây không phải là đường tròn(R<0)! | | | |
| * P: Trả về vị trí giữa điểm và đường tròn(ViTri\_Enum\_82\_LaHongLoc) | | | |

* + - 1. **Thiết kế các testcases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Hệ Thống** | Sample Project | | |
| **Module Code：** | CR100 - Export to excel | | |
| **Yêu Cầu Kiểm Thử** | CR1 - | | |
| Pass | 5 | Pending | 0 |
| Fail | 0 | Number of test cases: | 5 |



* 1. **Chạy các tests:**

