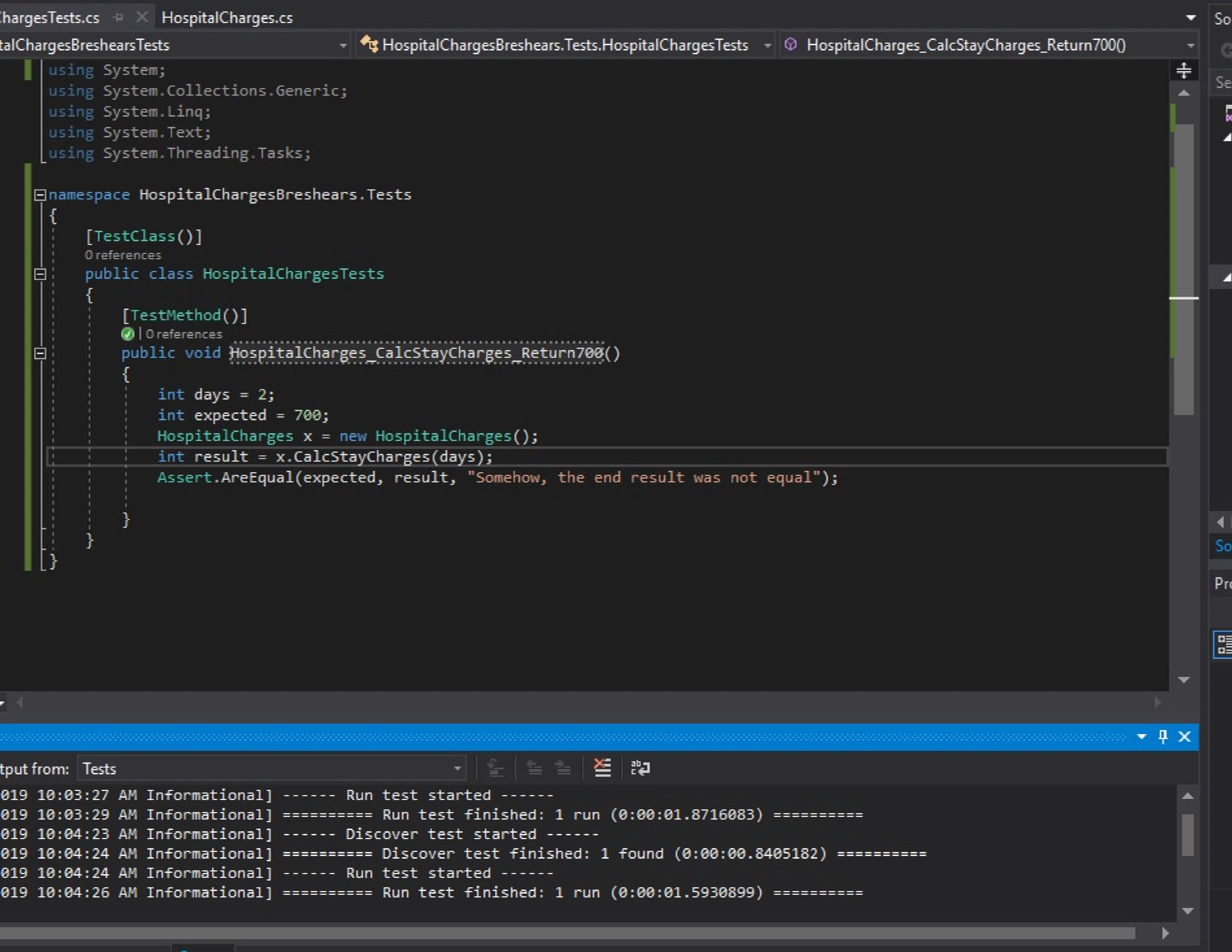
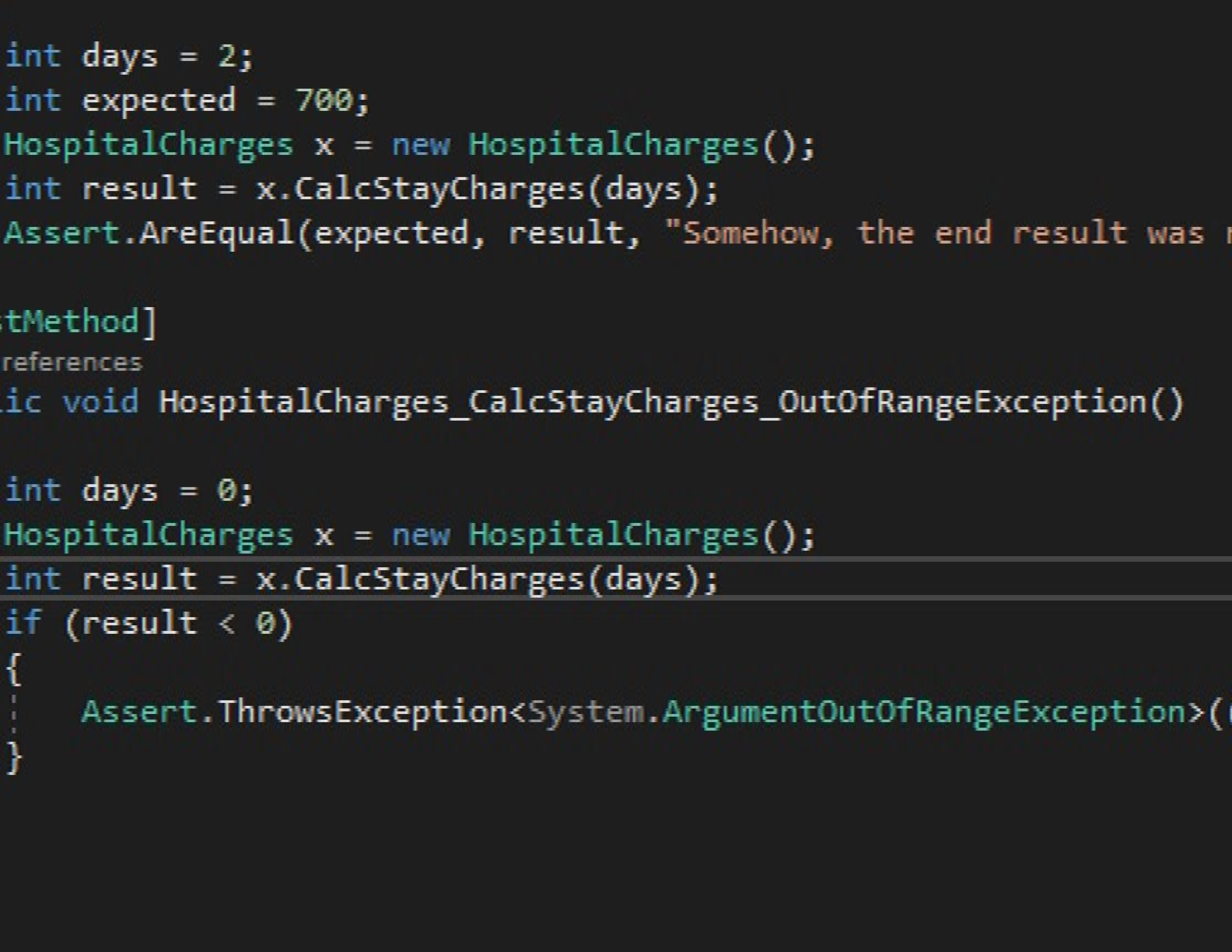
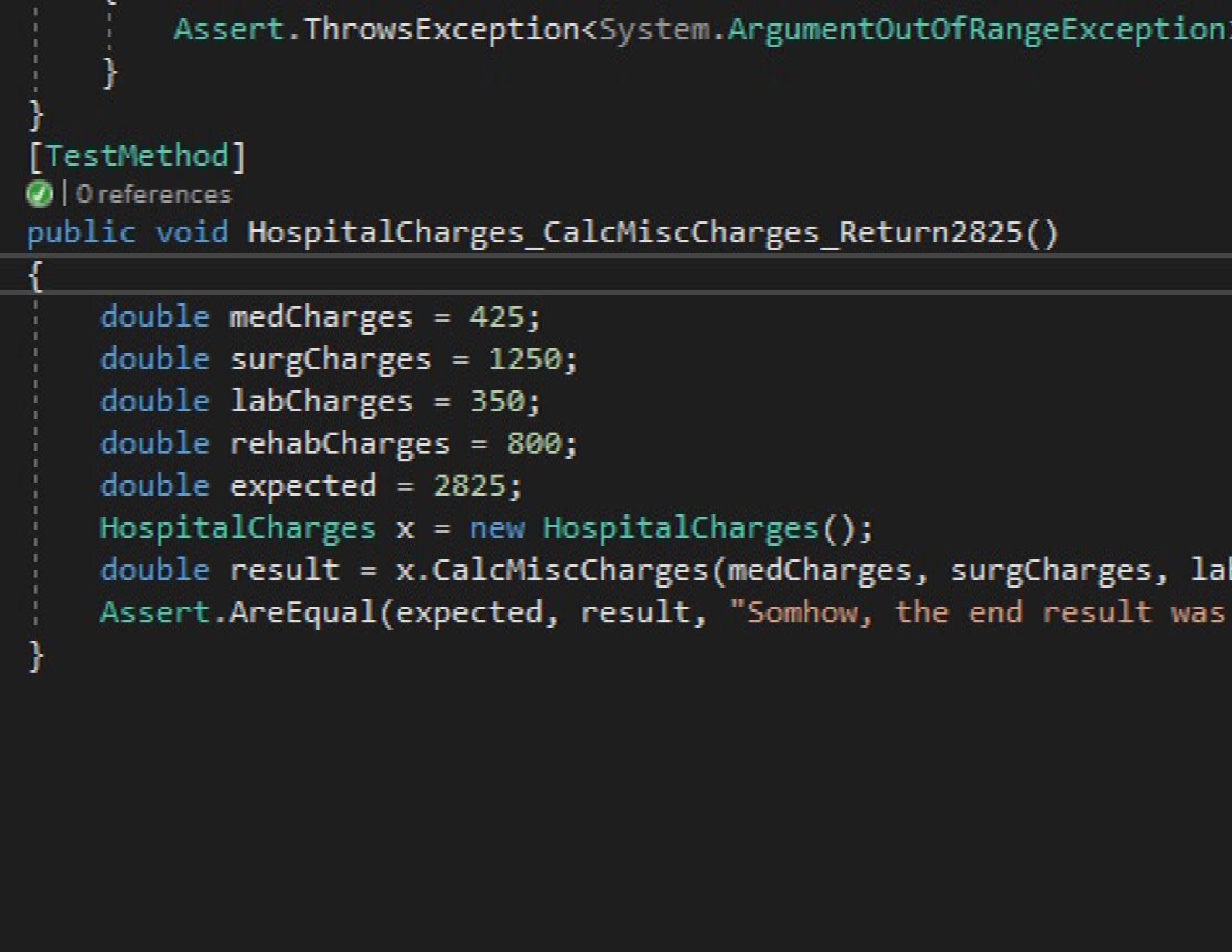

hargesTests.cs
HospitalCharges.cs
alChargesBreshearsTests
HospitalChargesBreshears.Tests.HospitalChargesTests
HospitalCharges_CalcStayCharges_Return700()
So
Se
So
Pro
tput from: Tests

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HospitalChargesBreshears.Tests
{
    [TestClass()]
    0 references
    public class HospitalChargesTests
    {
        [TestMethod()]
        | 0 references
        public void HospitalCharges_CalcStayCharges_Return700()
        {
            int days = 2;
            int expected = 700;
            HospitalCharges x = new HospitalCharges();
            int result = x.CalcStayCharges(days);
            Assert.AreEqual(expected, result, "Somehow, the end result was not equal");

        }
    }
}
```

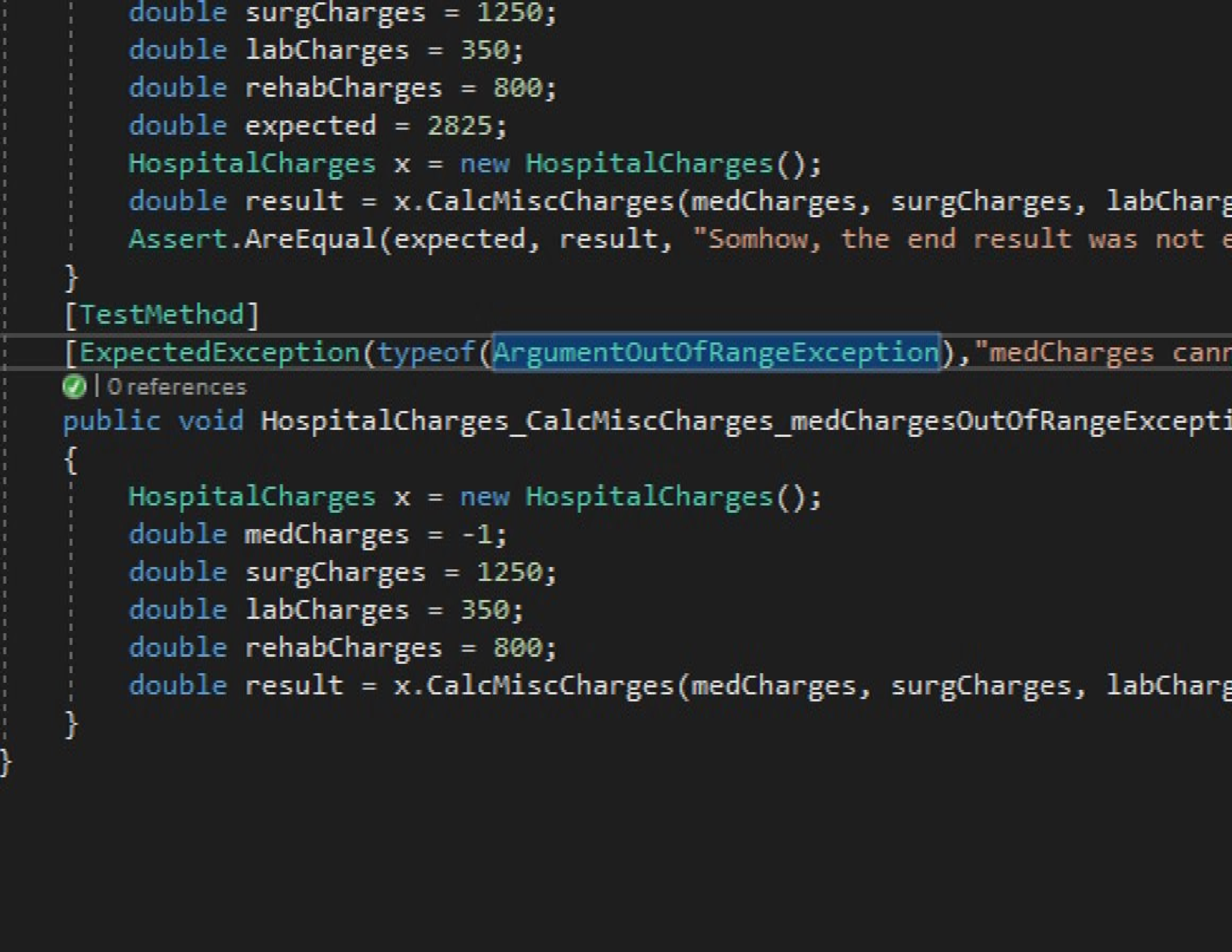
```
019 10:03:27 AM Informational] ------ Run test started ------
019 10:03:29 AM Informational] ========== Run test finished: 1 run (0:00:01.8716083) ==========
019 10:04:23 AM Informational] ------ Discover test started ------
019 10:04:24 AM Informational] ========== Discover test finished: 1 found (0:00:00.8405182) ==========
019 10:04:24 AM Informational] ------ Run test started ------
019 10:04:26 AM Informational] ========== Run test finished: 1 run (0:00:01.5930899) ==========
```



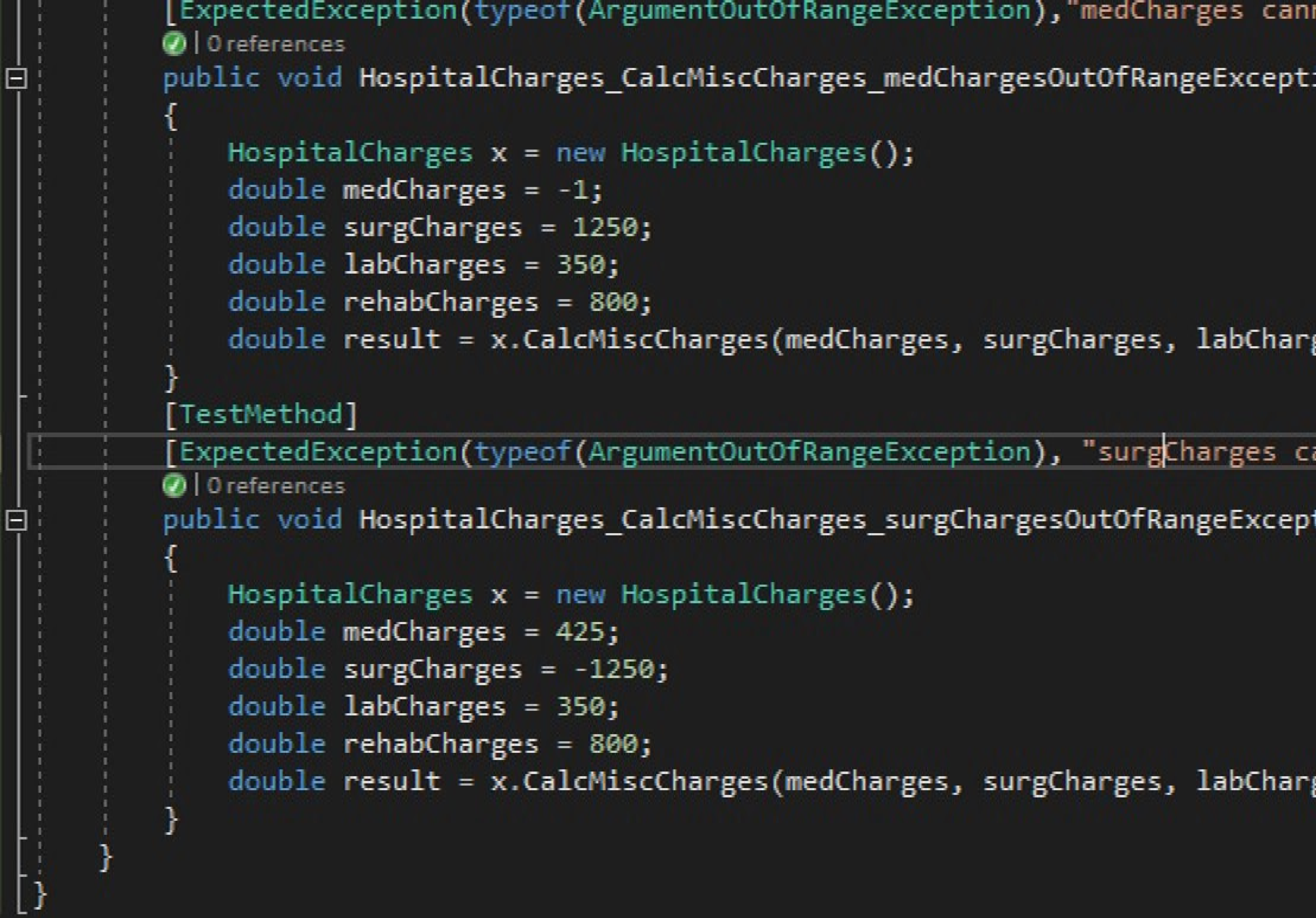
```
int days = 2;
int expected = 700;
HospitalCharges x = new HospitalCharges();
int result = x.CalcStayCharges(days);
Assert.AreEqual(expected, result, "Somehow, the end result was 

tMethod]
references
ic void HospitalCharges_CalcStayCharges_OutOfRangeException()

int days = 0;
HospitalCharges x = new HospitalCharges();
int result = x.CalcStayCharges(days);
if (result < 0)
{
    Assert.ThrowsException<System.ArgumentOutOfRangeException>(
}
```



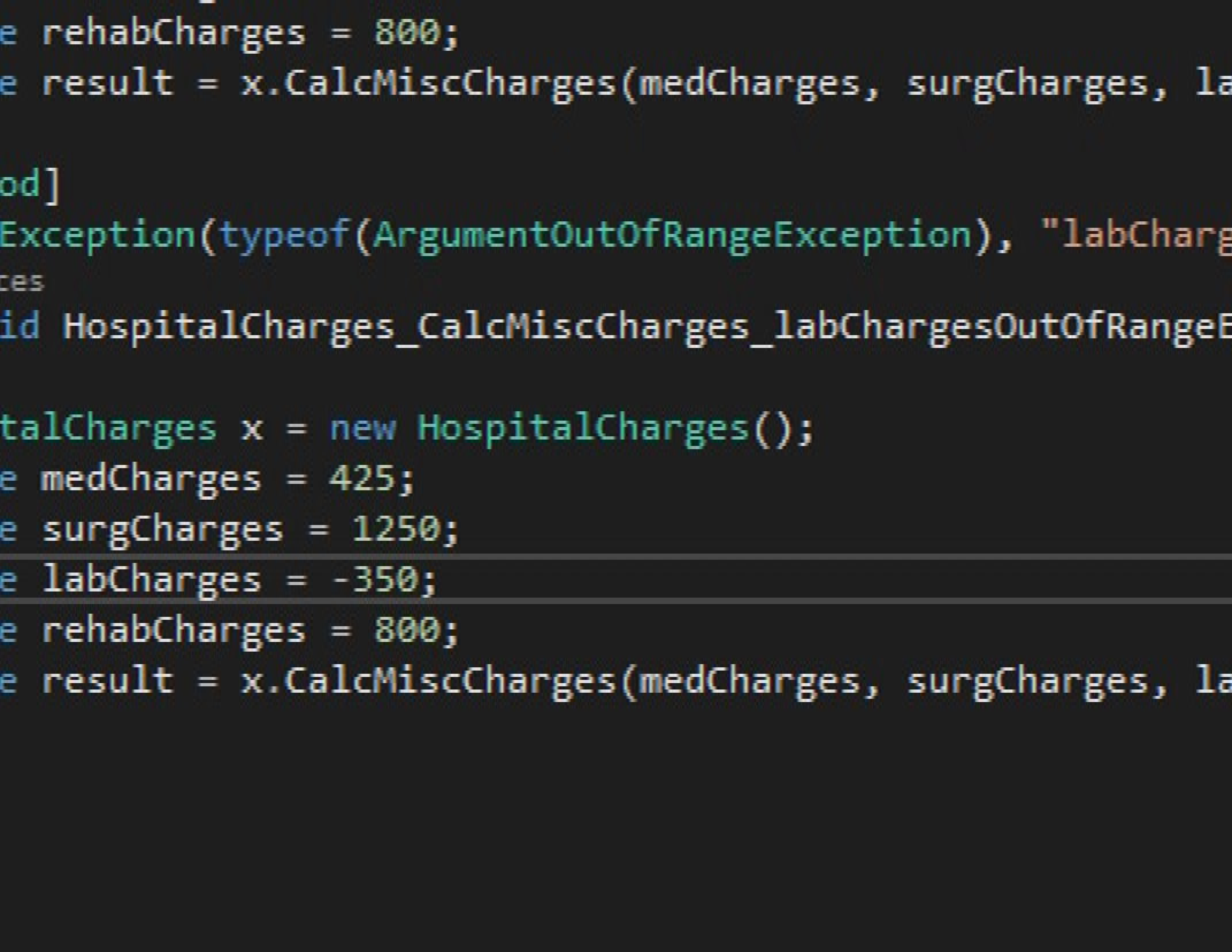
```
        Assert.ThrowsException<System.ArgumentOutOfRangeException
    }
}
[TestMethod]
0 references
public void HospitalCharges_CalcMiscCharges_Return2825()
{
    double medCharges = 425;
    double surgCharges = 1250;
    double labCharges = 350;
    double rehabCharges = 800;
    double expected = 2825;
    HospitalCharges x = new HospitalCharges();
    double result = x.CalcMiscCharges(medCharges, surgCharges, la
    Assert.AreEqual(expected, result, "Somhow, the end result was
}
```



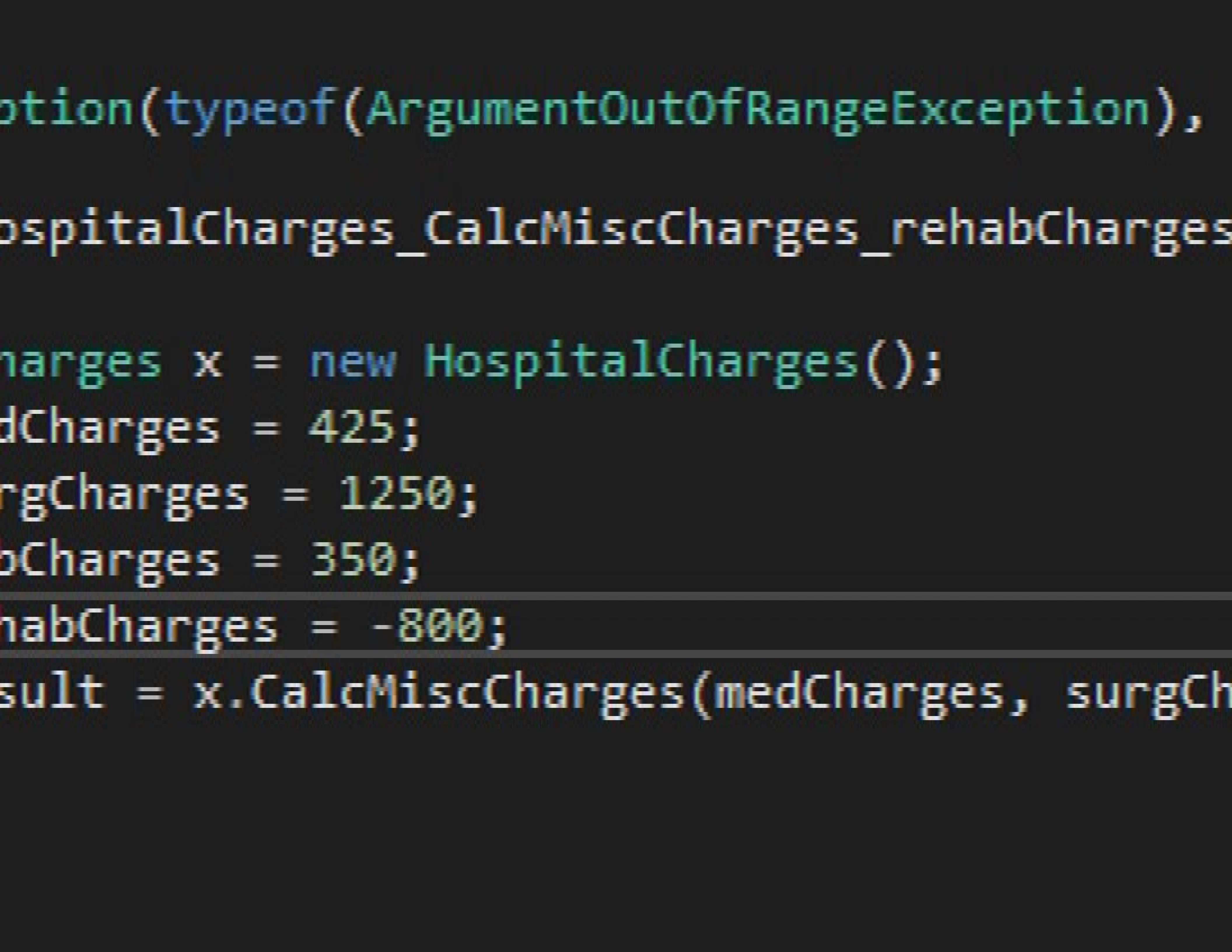
```
        double surgCharges = 1250;
        double labCharges = 350;
        double rehabCharges = 800;
        double expected = 2825;
        HospitalCharges x = new HospitalCharges();
        double result = x.CalcMiscCharges(medCharges, surgCharges, labCharg
        Assert.AreEqual(expected, result, "Somhow, the end result was not e
    }
    [TestMethod]
    [ExpectedException(typeof(ArgumentOutOfRangeException),"medCharges cann
    0 references
    public void HospitalCharges_CalcMiscCharges_medChargesOutOfRangeExcepti
    {
        HospitalCharges x = new HospitalCharges();
        double medCharges = -1;
        double surgCharges = 1250;
        double labCharges = 350;
        double rehabCharges = 800;
        double result = x.CalcMiscCharges(medCharges, surgCharges, labCharg
    }
}
```



```
        [ExpectedException(typeof(ArgumentOutOfRangeException),"medCharges can
        0 references
        public void HospitalCharges_CalcMiscCharges_medChargesOutOfRangeExcept
        {
            HospitalCharges x = new HospitalCharges();
            double medCharges = -1;
            double surgCharges = 1250;
            double labCharges = 350;
            double rehabCharges = 800;
            double result = x.CalcMiscCharges(medCharges, surgCharges, labChar
        }
        [TestMethod]
        [ExpectedException(typeof(ArgumentOutOfRangeException), "surgCharges ca
        0 references
        public void HospitalCharges_CalcMiscCharges_surgChargesOutOfRangeExcept
        {
            HospitalCharges x = new HospitalCharges();
            double medCharges = 425;
            double surgCharges = -1250;
            double labCharges = 350;
            double rehabCharges = 800;
            double result = x.CalcMiscCharges(medCharges, surgCharges, labChar
        }
    }
}
```



```
e rehabCharges = 800;
e result = x.CalcMiscCharges(medCharges, surgCharges, la

od]
Exception(typeof(ArgumentOutOfRangeException), "labCharg
ces
id HospitalCharges_CalcMiscCharges_labChargesOutOfRangeE

talCharges x = new HospitalCharges();
e medCharges = 425;
e surgCharges = 1250;
e labCharges = -350;
e rehabCharges = 800;
e result = x.CalcMiscCharges(medCharges, surgCharges, la
```



```
otion(typeof(ArgumentOutOfRangeException),

ospitalCharges_CalcMiscCharges_rehabCharges

harges x = new HospitalCharges();
dCharges = 425;
rgCharges = 1250;
bCharges = 350;
habCharges = -800;
sult = x.CalcMiscCharges(medCharges, surgCh
```