Francisco Crespo Martín

Gatitos alemanes buscan proteger sus miaufins

Clase Gatito:

Clase TanqueGatuno:

```
public class TanqueGatuno
{
    private int gatitosAbordo;
    private int maxGatitos;
    private int peso;
    private int maxPeso;
    private int miaufins;
setValue} (Alt+2)
    7 referencias | 7/7 pasando
    public TanqueGatuno(int peso, int gatitos,int miaufins)
    {
        this.maxPeso = peso;
        this.maxGatitos = gatitos;
        this.miaufins = miaufins;
    }

    5 referencias | 3/3 pasando
    public int getPeso() {
        return this.peso;
    }

    4 referencias | 2/2 pasando
    public int getGatitos() {
        return this.gatitosAbordo;
    }
```

```
2 referencias | 2/2 pasando
public int getMaxGatitos()
{
    return this.maxGatitos;
}

3 referencias | 1/1 pasando
public int getMaxPeso()
{
    return this.maxPeso;
}

4 referencias | 3/3 pasando
public int getMiaufins()
{
    return this.miaufins;
}
```

8 referencias | 4/4 pasando
```java
public void meterGatito(Gatito g)
{
    if (comprobarPesoMaximoAlcanzado() || getGatitos() >= getMaxPeso()) {
        throw new Exception("No caben más gatitos o el peso máximo ha sido alcanzado");
    } else
    {
        this.peso += g.getPeso();
        this.gatitosAbordo += 1;
    }
}
```

1 referencia | 1/1 pasando
```java
public void sacarGatito(Gatito g)
{
    if (getGatitos() < 1 || getPeso() < g.getPeso()) {
        throw new Exception("No se puede eliminar al gato que quieres");
    } else
    {
        this.peso -= g.getPeso();
        this.gatitosAbordo -= 1;
    }
}
```

3 referencias | 3/3 pasando
```java
public void comerMiaufin() {
    if (comprobarDisponibilidadMiaufins()) {
        this.miaufins--;
    } else
    {
        throw new Exception("No quedan miaufins");
    }
}
```

2 referencias | 1/1 pasando
```java
public Boolean comprobarPesoMaximoAlcanzado()
{
    if (getPeso() >= getMaxPeso()) {
        return true;
    } else
    {
        return false;
    }
}
```

3 referencias | 1/1 pasando
```java
public Boolean comprobarDisponibilidadMiaufins()
{
    if (getMiaufins() > 0) {
        return true;
    } else
    {
        return false;
    }
}
```

Tests:

```
[TestMethod]
0 referencias
public void TestCrearGatitos() {
    Gatito garfield = new Gatito(14);
    Assert.AreEqual(garfield.getPeso(), 14);
}//Fin del Test1

[TestMethod]
0 referencias
public void TestCrearTanque()
{
    TanqueGatuno antiPolacos = new TanqueGatuno(30, 4, 8);

    Assert.AreEqual(antiPolacos.getMaxGatitos(), 4);
    Assert.AreEqual(antiPolacos.getMaxPeso(), 30);
    Assert.AreEqual(antiPolacos.getMiaufins(), 8);
}//Fin del Test2
```

```
[TestMethod]
0 referencias
public void TestAñadirGatitos()
{
    Gatito dave = new Gatito(7);
    Gatito garfield = new Gatito(14);
    TanqueGatuno antiPolacos = new TanqueGatuno(30,4,8);

    antiPolacos.meterGatito(dave);
    antiPolacos.meterGatito(garfield);

    Assert.AreEqual(antiPolacos.getGatitos(), 2);
    Assert.AreEqual(antiPolacos.getPeso(), 21);
}//Fin del Test3
```

```
[TestMethod]
0 referencias
public void TestEliminarGatitos()
{
    Gatito dave = new Gatito(7);
    Gatito garfield = new Gatito(14);
    TanqueGatuno antiPolacos = new TanqueGatuno(30, 4, 8);

    antiPolacos.meterGatito(dave);
    antiPolacos.meterGatito(garfield);
    antiPolacos.sacarGatito(dave);

    Assert.AreEqual(antiPolacos.getGatitos(), 1);
    Assert.AreEqual(antiPolacos.getPeso(), 14);
}//Fin del Test4

[TestMethod]
0 referencias
public void TestComerMiaufins() {
    TanqueGatuno antiPolacos = new TanqueGatuno(30, 4, 8);

    antiPolacos.comerMiaufin();

    Assert.AreEqual(antiPolacos.getMiaufins(), 7);
}//Fin del Test5
```

```
[TestMethod]
0 referencias
public void TestComprobarPesoMax() {
    Gatito dave = new Gatito(10);
    Gatito garfield = new Gatito(14);
    TanqueGatuno antiPolacos = new TanqueGatuno(24, 4, 8);

    antiPolacos.meterGatito(dave);
    antiPolacos.meterGatito(garfield);

    Assert.IsTrue(antiPolacos.comprobarPesoMaximoAlcanzado());
}//Fin del Test6

[TestMethod]
0 referencias
public void TestComprobarDisponibilidadMiaufins()
{
    TanqueGatuno antiPolacos = new TanqueGatuno(24, 4, 1);
    Assert.IsTrue(antiPolacos.comprobarDisponibilidadMiaufins());

    antiPolacos.comerMiaufin();
    Assert.IsFalse(antiPolacos.comprobarDisponibilidadMiaufins());
}//Fin del Test7
```

```
[TestMethod]
| 0 referencias
public void TestGetters()
{
    Gatito dave = new Gatito(7);
    Gatito garfield = new Gatito(14);
    TanqueGatuno antiPolacos = new TanqueGatuno(24, 4, 8);

    antiPolacos.meterGatito(dave);
    antiPolacos.meterGatito(garfield);
    antiPolacos.comerMiaufin();

    Assert.AreEqual(dave.getPeso(), 7);
    Assert.AreEqual(antiPolacos.getPeso(), 21);
    Assert.AreEqual(antiPolacos.getMaxGatitos(), 4);
    Assert.AreEqual(antiPolacos.getMiaufins(), 7);
}//Fin del test8
```