|  |
| --- |
| [Type the company name] |
| P1 Reference |
| [Type the document subtitle] |

|  |
| --- |
| [Type the author name]  [Pick the date] |

Table of Contents

[Press ALT+A and F9 to update the TOC and other fields]

P1 Reference

Namespaces

[P1\_UnitTest](#topic_0000000000000000)

P1\_UnitTest Namespace

Classes

[UnitTest1](#topic_0000000000000001)

UnitTest1 Class

[System.Object](https://docs.microsoft.com/en-us/dotnet/api/system.object)

**P1\_UnitTest.UnitTest1**

|  |  |
| --- | --- |
| C# |  |
| [TestClass()] public class UnitTest1 | |

Requirements

**Namespace:**[P1\_UnitTest](#topic_0000000000000000)

**Assembly:** P1-UnitTest (in P1-UnitTest.dll)

Methods

[CheckGetPokemonByName](#topic_0000000000000008), [CheckGetPokemonByNumber](#topic_0000000000000006), [CheckImport](#topic_0000000000000002), [CheckIsEqual](#topic_0000000000000004), [CheckPokemonCreation](#topic_0000000000000003), [CheckPokemonToString](#topic_0000000000000005), [CheckWeaknessesAndResistances](#topic_0000000000000007)

UnitTest1.CheckGetPokemonByName Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokedex")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckGetPokemonByName() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokedex")]  public void CheckGetPokemonByName()  {  P1.Pokedex kanto = new P1.Pokedex(@"D:\Users\arobi\Desktop\pokemondatabase.txt");  P1.Pokemon rhydon = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);    Assert.IsTrue(kanto.getPokemonByName("Rhydon") == rhydon);  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckGetPokemonByNumber Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokedex")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckGetPokemonByNumber() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokedex")]  public void CheckGetPokemonByNumber()  {  P1.Pokedex kanto = new P1.Pokedex(@"D:\Users\arobi\Desktop\pokemondatabase.txt");  P1.Pokemon rhydon = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  Assert.IsTrue(rhydon == kanto.getPokemonByNumber(112));  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckImport Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("FileReader")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckImport() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("FileReader")]  public void CheckImport()  {  List<P1.Pokemon> pokemonList = new List<P1.Pokemon>();  pokemonList.AddRange(P1.FileReader.getPokemonFromFile(@"D:\Users\arobi\Desktop\pokemondatabase.txt"));  Assert.AreEqual(pokemonList.Count, 151, "Pokemon list not of expected size");  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckIsEqual Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokemon")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckIsEqual() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokemon")]  public void CheckIsEqual()  {  P1.Pokemon pokemon1 = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  P1.Pokemon pokemon2 = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  Assert.IsTrue(pokemon1 == pokemon2);  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckPokemonCreation Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokemon")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckPokemonCreation() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokemon")]  public void CheckPokemonCreation()  {  P1.Pokemon pokemon = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  Assert.IsTrue(pokemon.Name.Equals("Rhydon") && pokemon.PrimaryType.Equals(P1.Pokemon.Type.Ground) && pokemon.SecondaryType.Equals(P1.Pokemon.Type.Rock), "Test Pokemon Creation Successful");  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckPokemonToString Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokemon")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckPokemonToString() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokemon")]  public void CheckPokemonToString()  {  P1.Pokemon rhydon = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  string testString = "Pokemon: Rhydon\nTypes: Ground-Rock\n";  Assert.IsTrue(testString.Equals(rhydon.ToString()));  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

UnitTest1.CheckWeaknessesAndResistances Method

|  |  |
| --- | --- |
| C# |  |
| [TestMethod()] [TestCategory("Pokedex")] [TestProperty("Test", "1")] public [void](https://docs.microsoft.com/en-us/dotnet/api/system.void) CheckWeaknessesAndResistances() | |

Source code

|  |
| --- |
| [TestMethod, TestCategory("Pokedex"), TestProperty("Test", "1")]  public void CheckWeaknessesAndResistances()  {  P1.Pokemon rhydon = new P1.Pokemon("Rhydon", P1.Pokemon.Type.Ground, P1.Pokemon.Type.Rock);  Assert.AreEqual(4.0, P1.Pokedex.getWeaknessVal(rhydon, P1.Pokemon.Type.Grass), "Incorrect Value: Rhydon is 4x times weak to grass");  } |

See Also

Applies to: [UnitTest1](#topic_0000000000000001)

# Index

[CheckGetPokemonByName Method](#topic_0000000000000008)

[CheckGetPokemonByNumber Method](#topic_0000000000000006)

[CheckImport Method](#topic_0000000000000002)

[CheckIsEqual Method](#topic_0000000000000004)

[CheckPokemonCreation Method](#topic_0000000000000003)

[CheckPokemonToString Method](#topic_0000000000000005)

[CheckWeaknessesAndResistances Method](#topic_0000000000000007)

[P1 Reference](#topic_0000000000000009)

[P1\_UnitTest Namespace](#topic_0000000000000000)

[UnitTest1 Class](#topic_0000000000000001)