Forge Class

The forge class in the main class of the DataForge library, this class contains several partial classes which are used to categorize functions. This will keep the package clean and straight forward.

Namespace:

```
using DataForge;
```

Person methods

RandomFirstname(Genders gender)

This method will generate a random firstname based on the gender provided. Gender is an enum from the DataForge package.

```
Console.WriteLine("Random firstname : " + Forge.Person.RandomFirstname());
```

RandomFirstname(string gender = "NEUTRAL")

This is an overload method of `RandomFirstname`, here you pass a string value instead of the enum.

```
Console.WriteLine("Random female firstname : " +
Forge.Person.RandomFirstname("FEMALE"));
```

RandomLastname()

This method returns a random lastname.

```
Console.WriteLine("Random lastname : " + Forge.Person.RandomLastname());
```

RandomGender()

This method will return a random gender from the forge gender enum.

```
Console.WriteLine("Random Gender : " + Forge.Person.RandomGender());
```

Communication methods

RandomEmailAddress(string firstname = "", string lastname = "")

The following method will return a random email address, first- and lastname are optional parameters. When parameters are empty a random first- and lastname will be generated. The output of the method will have the following format `{firstname}.{lastname}@{domain}`. All spaces and special characters will be replace with a `.`.

```
Console.WriteLine("Random Email Address without params : " +
Forge.Communication.RandomEmailAddress());
Console.WriteLine("Random Email Address with params : " +
Forge.Communication.RandomEmailAddress("Quinten", "De Clerck"));
Console.WriteLine("Random Email Address with single param : " +
Forge.Communication.RandomEmailAddress("Quinten"));
```

RandomPhoneNumber(string pattern = "+32 ## ########")

This method will generate a random phone number, the method will use a pattern with `#` as placeholders each `#` will be replaced with a random number. The pattern is an optional parameter.

```
Console.WriteLine("Random Phone number : " +
Forge.Communication.RandomPhoneNumber());
Console.WriteLine("Random Phone number : " +
Forge.Communication.RandomPhoneNumber("+334 ## ## ## ##"));
```

RandomPhoneNumber(bool plus, string countryCode = "32", string areaCode = "20")

This is an overload method of the RandomPhoneGenerator, this method accepts multiple parameters that are optional.

```
Console.WriteLine("random phone number : " +
Forge.Communication.RandomPhoneNumber(true));
Console.WriteLine("random phone number : " +
Forge.Communication.RandomPhoneNumber(true, "123", "45"));
```

Address methods

RandomStreet(bool includeNumber = false)

This method will generate a random street as string from a list, the include number is an optional parameter. This will add a random number to the street between 0 and 100.

```
Console.WriteLine("Random street : " + Forge.Address.RandomStreet());
Console.WriteLine("Random street : " + Forge.Address.RandomStreet(true));
```

RandomCity()

This method generates a random city as string from a predefined list.

```
Console.WriteLine("Random city : " + Forge.Address.RandomCity());
```

Text methods

GenerateRandomText(int numWords)

The following method will generate a random lorem ipsum text with a number of words.

```
Console.WriteLine("Random text : " + Forge.Text.GenerateRandomText(10));
```

DateTime methods

GenerateRandomDateTimeBetweenYears(int startYear, int endYear)

This method will generate a random DateTime between a range of years.

```
DateTime dt = Forge.DateTime.GenerateRandomDateTimeBetweenYears(1000, 2000);
Console.WriteLine("Random Birthday between the years 1000 & 2000 : " +
dt.ToString("dd/MMM/yyyy"));
```

GenerateRandomDateTimeBetweenAges(int minAge, int maxAge)

This method will generate a random DateTime to match with a birthday between the range of ages.

```
DateTime dt = Forge.DateTime.GenerateRandomDateTimeBetweenAges(10, 27);
Console.WriteLine("Random birthday between the ages 10 & 27 : " +
dt.ToString("dd/MMM/yyyy"));
```

Utils

GetRandomEnumValue()

This method will generate a random value from any enum you pass.

```
Console.WriteLine("Random enum value : " + Forge.Utils.GetRandomEnumValue<Genders>
());
```

RandomStringPattern(string pattern, ConversionTypes conversionTypes)

The following method generate a random string value with the pattern provided, the pattern must use # as placeholders, the ConversionTypes enum will generate the pattern with numbers, characters or both.

```
Console.WriteLine("Random string based on pattern : " +
Forge.Utils.RandomStringPattern("###-#-###", ConversionTypes.Numerical));
```

Forge Methods

CreateInstance

This method method will create a single instance of any kind of object.

```
Person newPerson = Forge.CreateInstance<Person>(p =>
{
    p.FirstName = Forge.Person.RandomFirstname();
    p.LastName = Forge.Person.RandomLastname();
});
```

CreateInstances

The following method will create a list of instances of the specified object.

```
List<Person> persons = Forge.CreateInstances<Person>(10, p =>
{
    p.FirstName = Forge.Person.RandomFirstname();
    p.LastName = Forge.Person.RandomLastname();
});
```