

Changes

1. Adjust the class name. Replace the original GenerateUtil with factory, and replace the original generateObject method name with buildObject.
2. Add three abstract factory classes, and each has three concrete implementation classes. The corresponding relationship is as follows:
 - AbstractExplodableFactory
 - DNTFactory
 - TNTFactory
 - UnknownFactory
 - AbstractExplosionFactory
 - PETNFactory
 - RDXFactory
 - TetrylFactory
 - AbstractExplosionFactoryA
 - ANFactory
 - NGFactory
 - HMXFactory
3. Adjust the calling relationship of methods in the factory. Taking buildExplosions as an example, a factory is randomly selected from the factoryList, and the explosion is generated through the getObject of the factory. Specific factories decide how to produce , and the internal implementation still uses reflection to build instances .

```
1 //The call stack is as follows
2
3 Factory.getRandomFactory ();
4 Factory.buildExplosion ();
5 Factory.buildExplosions ();
```

How to run this project

The main method in the top-level Driver class integrates three demo methods of all packages, and can directly run and print results in console.

By the way, each model class has a static demo method.

Four packages

I divided all classes into four packages according to function and assignment requirements. The four packages are composed of explosionModel-related classes, explosionModelA-related classes, explodable-related classes and tool classes.

Explosion package

1. DNT, TNT and UnknownBomb are three classes that implement ExplodableAPI interface.
2. The ExplodableAPI declares the explode method
As an abstract base class, AbstractExplodableModel defines the methods that need to be rewritten in ExplodableModel class.
3. The demo method is provided in the ExplodableModel, and the add method is rewritten to add the class instance that implements explodable into the collection. Finally, in the explode method, all the elements in the collection will execute the explode method.

Model package

1. PETN, RDX and tetryl are three classes that inherit explosion
2. Explosion provides the default explode method
3. AbstractExplosionModel, as an abstract base class, defines the methods to be rewritten in ExplosionModel
4. The ExplosionModel provides the demo method to run, and rewrites the add method to add the class instance that implements the explosion to the collection. Finally, in the explode method, all elements in the collection run the explode method.

Modela package

1. An, HMX and ng are three classes that inherit AbstractExplosion
2. AbstractExplosion provides the default explode method (there is no abstract method, but all subclasses that inherit the abstract class override the method)
3. As in 3.2, the AbstractExplosionAPI provides default methods, so the ExplosionAPI class that implements the AbstractExplosionAPI interface can directly call its default implementation
4. ExplosionModelA provides the demo method to run, and rewrites the add method to add the class instance that implements the explosion to the collection. Finally, in the explode method, all elements in the collection run the explode method

Util package

1. Three kinds of random instance generation are provided, and an integer is generated randomly to limit the number of instances put into the container
2. The methods include
 - Generate random integer
 - Get the instance of a class randomly by reflection
 - Put the instance into the collection
 - Get the collection