



Project

First-Java

src

main

java

itstep.learning

db

fs

ioc

DbModule

iocDemo

RandDemo

RandModule

ServicesModule

oop

services

hash

randstring

NameFile

PermanentPassword

RandStringService

Salt

TemporaryPassword

App

Basics

resources

db.ini

test

target

.gitignore

pom.xml

text.txt

External Libraries

Scratches and Consoles

PermanentPassword.java

iocDemo.java

App.java

ShaHashService.java

Md5HashService.java

HashService.java

Salt.java

NameFile.java

RandModule.java

```
1 package itstep.learning.ioc;
2
3 import com.google.inject.AbstractModule;
4 import com.google.inject.name.Names;
5 import itstep.learning.services.hash.HashService;
6 import itstep.learning.services.hash.ShaHashService;
7 import itstep.learning.services.randstring.*;
8
9 public class RandModule extends AbstractModule {
10
11     @Override
12     protected void configure() {
13
14         bind(RandStringService.class).annotatedWith(Names.named("TemporaryPassword")).to(TemporaryPassword.class);
15         bind(RandStringService.class).annotatedWith(Names.named("NameFile")).to(NameFile.class);
16         bind(RandStringService.class).annotatedWith(Names.named("Salt")).to(Salt.class);
17         bind(RandStringService.class).annotatedWith(Names.named("PermanentPassword")).to(PermanentPassword.class);
18
19         bind(HashService.class).toInstance(new ShaHashService());
20
21         super.configure();
22     }
23 }
24
```

First-Java > src > main > java > itstep > learning > ioc > RandModule > configure

20:9 CRLF UTF-8 4 spaces

The screenshot shows an IDE with a project named "First-Java" and a file named "RandDemo.java" open. The code defines a class `RandDemo` with four private final fields: `temporaryPassword`, `nameFile`, `salt`, and `permanentPassword`, each of type `RandStringService`. The class has an `@Inject` annotation and a constructor that takes four arguments of the same type. The `run()` method prints the values of these fields.

```
6 public class RandDemo 1 usage new *
7 {
8     private final RandStringService temporaryPassword; 2 usages
9     private final RandStringService nameFile; 2 usages
10    private final RandStringService salt; 2 usages
11    private final RandStringService permanentPassword; 2 usages
12
13    @Inject no usages new *
14    public RandDemo(@Named("TemporaryPassword") RandStringService temporaryPassword,
15                    @Named("NameFile") RandStringService nameFile,
16                    @Named("Salt") RandStringService salt,
17                    @Named("PermanentPassword") RandStringService permanentPassword) {
18        this.temporaryPassword = temporaryPassword;
19        this.nameFile = nameFile;
20        this.salt = salt;
21        this.permanentPassword = permanentPassword;
22    }
23
24    public void run() 1 usage new *
25    {
26        System.out.println("Name File: " + nameFile.getRandString());
27        System.out.println("Salt: " + salt.getRandString());
28        System.out.println("Temporary Password: " + temporaryPassword.getRandString());
29        System.out.println("Permanent Password: " + permanentPassword.getRandString());
30    }
```

The Run window shows the output of the application:

```
Run App x
C:\Program Files\Java\jdk-1.8\bin\java.exe ...
Name File: byhebmckoknlnbovslewrzwxitrkjsbyxzn
Salt: 08351f0d1e1cc6a9aeaf
Temporary Password: 734807
Permanent Password: thD11Al-qMVcxNxsV}NR
```

The status bar at the bottom indicates the file is "First-Java > src > main > java > itstep > learning > ioc > RandDemo" with a line length of 6:14, CRLF line endings, UTF-8 encoding, and 4 spaces for indentation.