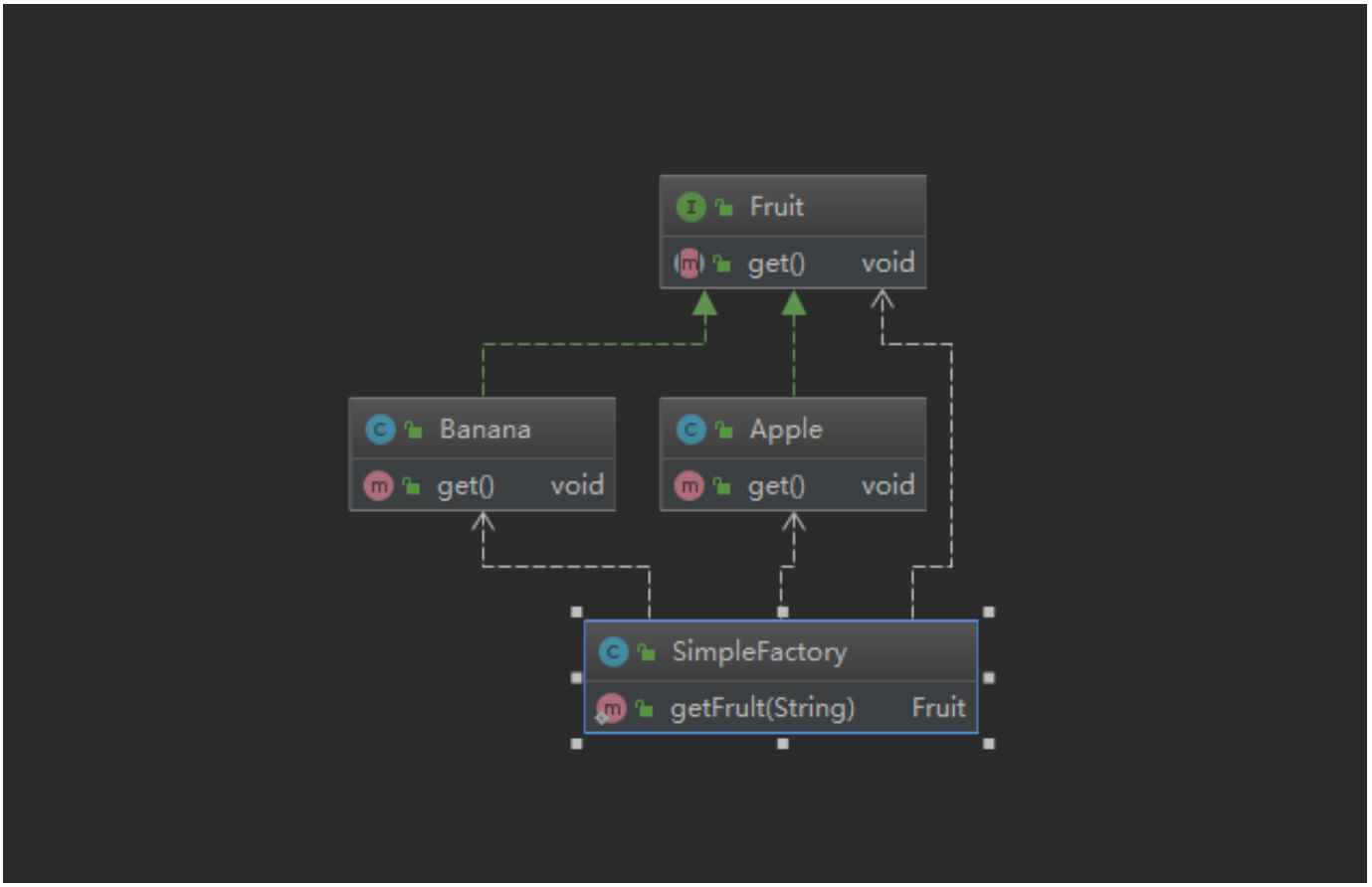
类图：



1.创建一个接口

```
public interface Fruit {
    // 采集水果
    void get();
}
```

2.创建实现类

```
public class Apple implements Fruit{
```

```
    @Override
    public void get() {
        System.out.println("采集苹果");
    }
}

```

```
public class Banana implements Fruit{
    @Override
    public void get() {
        System.out.println("采集香蕉");
    }
}
```

3.创建简单工厂

```
public class SimpleFactory {


  /*  public static Fruit getApple() {
        return new Apple();
    }

    public static Fruit getBanana() {
        return new Banana();
    }*/

    public static Fruit getFrult(String type) throws Exception {
        if (type.equals("apple")) {
            return Apple.class.newInstance();
        } else if (type.equals("banana")) {
            return Banana.class.newInstance();
        } else {
            return null;
        }
    }

```

```
   /* public static Fruit getFrult1(String type) throws Exception {
        Class<?> name = Class.forName(type);
        return (Fruit) name.newInstance();
    }*/
}
```

4.使用

```
public class Test {
    public static void main(String[] args) throws Exception{

      /*  Fruit apple = SimpleFactory.getApple();
        Fruit banana = SimpleFactory.getBanana();
        apple.get();
        banana.get();*/
     /*   Fruit apple = SimpleFactory.getFrult("apple");
        Fruit banana = SimpleFactory.getFrult("banana");
        apple.get();
        banana.get();*/
        Fruit apple = SimpleFactory.getFrult1("Apple");
        Fruit banana = SimpleFactory.getFrult1("Banana");
        apple.get();
        banana.get();
    }
}
```