

---

---

```
/**
```

\* 未传入泛型实参时，与泛型类的定义相同，在声明类的时候，需将泛型的声明也一起加到类中

\* 即：class FruitGenerator<T> implements Generator<T>{

\* 如果不声明泛型，如：class FruitGenerator implements Generator<T>，编译器会报错：“Unknown class”

```
*/
```

```
class FruitGenerator<T> implements Generator<T>{
```

```
    public T next() {
```

```
        return null;
```

```
    }
```

```
}
```

---

---

```
public interface Generator<T> {
```

```
    public T next();
```

```
}
```

---

---

```
/**
```

\* 传入泛型实参时：

\* 定义一个生产者实现这个接口，虽然我们只创建了一个泛型接口Generator<T>

\* 但是我们可以为T传入无数个实参，形成无数种类型的Generator接口。

\* 在实现类实现泛型接口时，如已将泛型类型传入实参类型，则所有使用泛型的地方都要替换成传入的实参类型

\* 即：Generator<T>，public T next();中的T都要替换成传入的String类型。

```
*/
```

```
public class FruitGenerator implements Generator<String> {
```

```
    private String[] fruits = new String[]{"Apple", "Banana", "Pear"};
```

```
    @Override
```

```
    public String next() {
```

```
Random rand = new Random();  
return fruits[rand.nextInt(3)];  
}  
}
```

---

---