

- Factory Abstract Factory Singleton Builder ProtoType
- Bridge Adapter Decorator Proxy Composite Facade
- Strategy Interpreter Command Observer Iterator Template Method Visitor

FactoryMethod

Singleton

```
/**
 * @Description
 * @author zch
 * @time 2018 7 20 11:03:43
 */
class Demo2 {

    private static volatile Demo2 inst;

    private Demo2() {}

    public static Demo2 getInst() {
        if (inst == null) {
            synchronized (Demo2.class) {
                if (inst == null) {
                    inst = new Demo2();
                }
            }
        }
    }
}
```

```

        }
    }
    return inst;
}
}

```

```

class Resource {
}

/**
 * @Description    effective java
 * @author zch
 * @time 2018 7 20   11:14:16
 *
 */
enum Demo3 {

    INSTANCE;

    private Resource instance;

    Demo3() {
        instance = new Resource();
    }

    public Resource getInstance() {
        return instance;
    }
}

```

```

/**
 * @Description
 * @author zch
 * @time 2018 7 20   11:21:14
 *
 */
class Demo4 {
    private Demo4() {

```

```

    }

    public static Demo4 getInstance() {
        return Holder.inst;
    }

    private static class Holder {
        private static Demo4 inst = new Demo4();
    }
}

```

Prototype

```
A a = new A.Builder().seta(1).setb(2).setc(3).build();
```

Adapter

3*3=9

Adapter Pattern

