模板方法模式 - 和平主义者

发表于 2022-09-21 | 更新于 2023-04-06 | 设计模式

| 字数总计: 2.3k | 阅读时长: 7 分钟 | 阅读量: 873 | 评论数: 0

配套视频课程已更新完毕，大家可通过以下两种方式观看视频讲解：

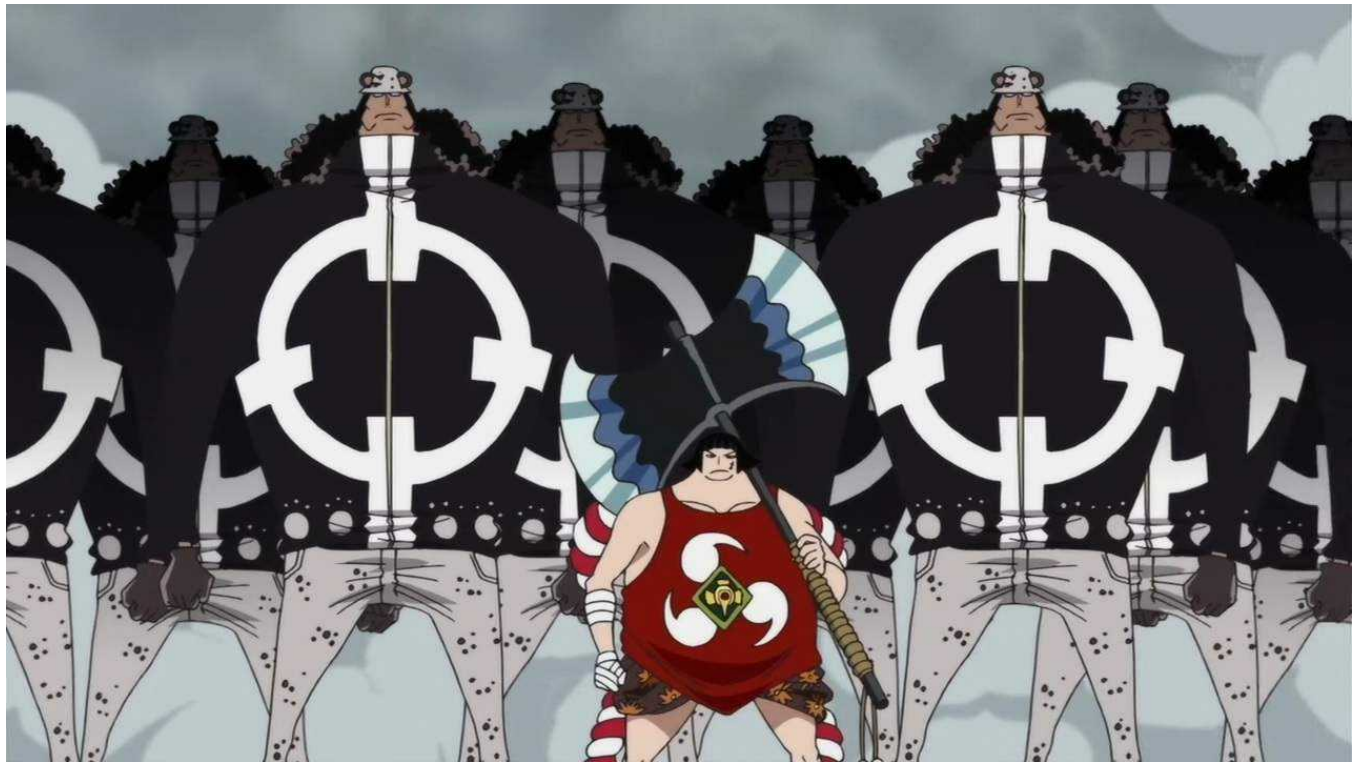
关注公众号： 爱编程的大丙 ，或者进入 大丙课堂 学习。

苏丙榅

合抱之木，生于毫末；九层之台，起于垒土；千里之行，始于足下。

1. 和平主义者

和平主义者是世界顶级科学家贝加庞克为世界政府研制的人形兵器，归海军直属，其相貌和王下七武海中的巴索罗缪·大熊一样，可以发射镭射光，杀伤力和战斗力非常强。海军利用这些和平主义者来消灭赏金上亿的海贼。



两年后，贝加庞克又推出了和平主义者 2.0 版本，名字叫做炽天使，身材比和平主义者小了很多，但是战斗力和自身携带的技能却上升了一个档次。

目录

对于贝加庞克来说这两款机器人是一脉相承的，也就是说 他们的架构是一样的，2.0版本的炽天使只是在原来架构基础上增强了某些功能，或者在原来预留的接口上实现了某些功能，使用这种方式无疑能够使研发效率最大化。

和领先人类科技 500 年的天才科学家贝加庞克设计机器人的思路类似，在编程的时候也有一种类似的设计模式叫做模板方法模式。模板方法模式就是在基类中定义一个算法的框架，允许子类在不修改结构的情况下重写算法的特定步骤。说的再直白一些就是先定义一个基类，在基类中把与需求相关的所有操作函数全部作为虚函数定义出来，然后在这个基类的各个子类中重写父类的虚函数，这样子类基于父类的架构使自己有了和其他兄弟类不一样的行为。模板方法这种设计模式是对多态的典型应用。

模板方法这种模式在显示生活中的应用场景也有很多，比如：

CMake 保姆级教程（下）
2023-03-15

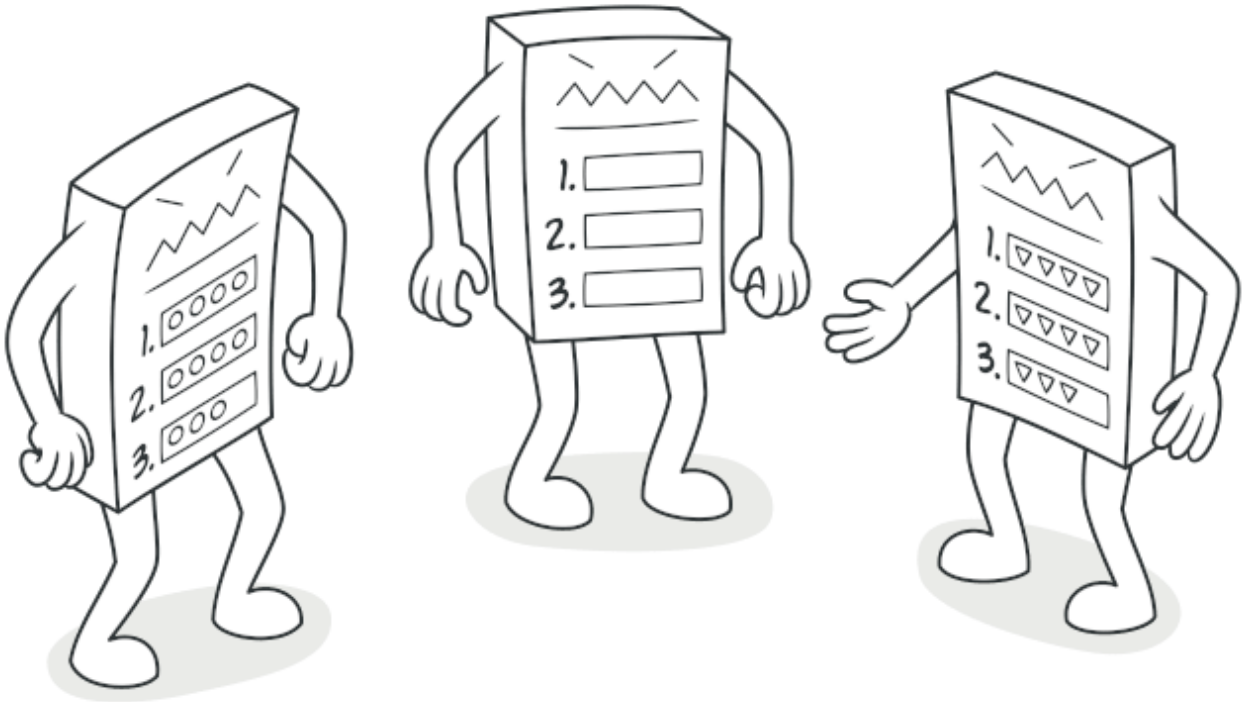
CMake 保姆级教程（上）
2023-03-06

访问者模式 - 再见，香波地群岛
2022-09-22

模板方法模式 - 和平主义者
2022-09-21

状态模式 - 文斯莫克・山治
2022-09-20

1. 盖房子：地基、建造架构相同，但是水电安装却可以不一样
2. 造车、造船：使用相同的车、船架构可以造出很多不同型号的车、船
3. 考试卷：试题相同，但是每个人书写的答案却不尽相同。



2. 人形兵器

2.1 理想

贝加庞克在设计机器人的时候肯定是花了很多心思，尽量让他趋于完美。所以这款机器人的架构中肯定是留有很多接口的，以方便科学技术成熟之后对他们进行拓展和升级。假设贝加庞克的机器人对应的是一个类，那么在这个类中肯定需要定义很多的虚函数，以方便在子类中进行实现或者是功能的改进，我们认为在这个机器人类中提供了以下这些功能：

- 机器人的武器系统
- 机器人的外观
- 机器人的战斗力
- 机器人的名字
- 机器人的自愈能力
- 机器人是否可以飞行
- 机器人是否能够自主控制
- 得到机器人的所有属性

根据以上描述，我们就可以把机器人的抽象类定义出来了：

C++

```cpp
// 抽象机器人类
class AbstractRobot
{
public:
    // 武器
    virtual void weapon() = 0;
    // 外观
    virtual void appearance() = 0;
```

```
    // 战斗能力
    virtual void fightAbility() = 0;
    // 名字
    virtual string getName() = 0;
    // 自愈能力
    virtual void selfHealing() {};
    // 是否能飞
    virtual bool canFlying()
    {
        return false;
    }
    // 是否是自动控制
    virtual bool isAuto()
    {
        return true;
    }
    // 得到机器人属性
    virtual void getProperty()
    {
        cout << "贝加庞克制造的" << getName() << "有以下属性: " << endl;
        if (canFlying())
        {
            cout << "有飞行能力!" << endl;
        }
        else
        {
            cout << "没有飞行能力!" << endl;
        }
        if (isAuto())
        {
            cout << "可以自动控制, 完全体机器人!" << endl;
```

```
40          }
41          else
```

在上面的抽象类中提供了一些纯虚函数，这些纯虚函数在子类中是必须要进行重写的，否则子类也是抽象类，这样子类就无法实例化了。另外，还有一些虚函数，这些虚函数可以根据实际需求可以在子类中重写，也可以不重写。有了这个抽象类，机器人的骨架我们就已经搭建好了，接下来就需要为梦想而奋斗，让梦想照进现实了。

2.2 现实

和平主义者

理想很丰满，现实很骨感，对于贝加庞克这个天才科学家来说在科研的道路上饭也还是得一口一口吃，所以他造出了他理想中的第一代机器人：和平主义者：

C++

```cpp
// 和平主义者
class Pacifist : public AbstractRobot
{
public:
    // 武器
    void weapon() override
    {
        cout << "可以发射镭射光..." << endl;
    }
    // 外观
```

```
    void appearance() override
    {
        cout << "外部和巴索罗米·熊一样，体型庞大，拥有呈半圆形的耳朵，内部似乎金属。"
    }
    // 能力
    void fightAbility() override
    {
        cout << "结实抗揍，可以通过手部或者嘴部发射镭射激光，可以融化钢铁!!!" << en
    }
    string getName() override
    {
        return "和平主义者";
    }
};
```

作为第一代机器人，和平主义者是不完美的，它只是实现了一些必备功能，比如它没有自愈能力，也不能飞行（没有重写父类的这些虚函数，也就是意味着这些功能还没有实现）。

炽天使

又过了两年，技术更加成熟了，所以就有了第二个版本的炽天使：

C++

```
// 炽天使
class Seraphim : public AbstractRobot
{
public:
    // 武器
    void weapon() override
```

```
    {
        cout << "可以发射镭射激光，鹰眼外形的炽天使携带者一把巨剑，可以斩断一切!!!"
    }
    // 外观
    void appearance() override
    {
        cout << "外观和七武海小时候的外形一样，并且拥有一对和烬一样的翅膀!!!" << end
    }
    // 能力
    void fightAbility() override
    {
        cout << "不仅可以发射镭射激光，还拥有七武海的能力，牛逼plus，无敌了!!!!" <<
    }
    // 自愈能力
    void selfHealing() override
    {
        cout << "非常厚实抗揍，并且拥有非常强的自愈能力，开挂了!!!" << endl;
    }
    // 是否能飞
    bool canFlying() override
    {
        return true;
    }
    string getName() override
    {
        return "炽天使";
    }
};
```

可以看到第二个版本的机器人 – 炽天使现在有了自愈和飞行的能力（重写了父类的这些虚函数，实现了对应的功能）。

通过上面的代码可以看到不管是第一代和 平主义者 还是第二代的 炽天使 他们是发生了改变，但这些变化背后对应的却是不变，那就是 父类提供的架构没有改变。 假设以后贝加庞克要制造第三代的机器人 和平天使大丙 ， 只需在新的子类中重新实现父类提供的虚函数就可以了。

2.3 性能

最后我们来对比一下贝加庞克的这两款机器人的属性，看一下炽天使是否可以秒杀和平主义者：

C++

```cpp
int main()
{
    AbstractRobot* robot = nullptr;
    robot = new Pacifist;
    robot->getProperty();
    delete robot;
    cout << "====================================" << endl;
    robot = new Seraphim;
    robot->getProperty();
    delete robot;
    return 0;
}
```

得到的结果如下:

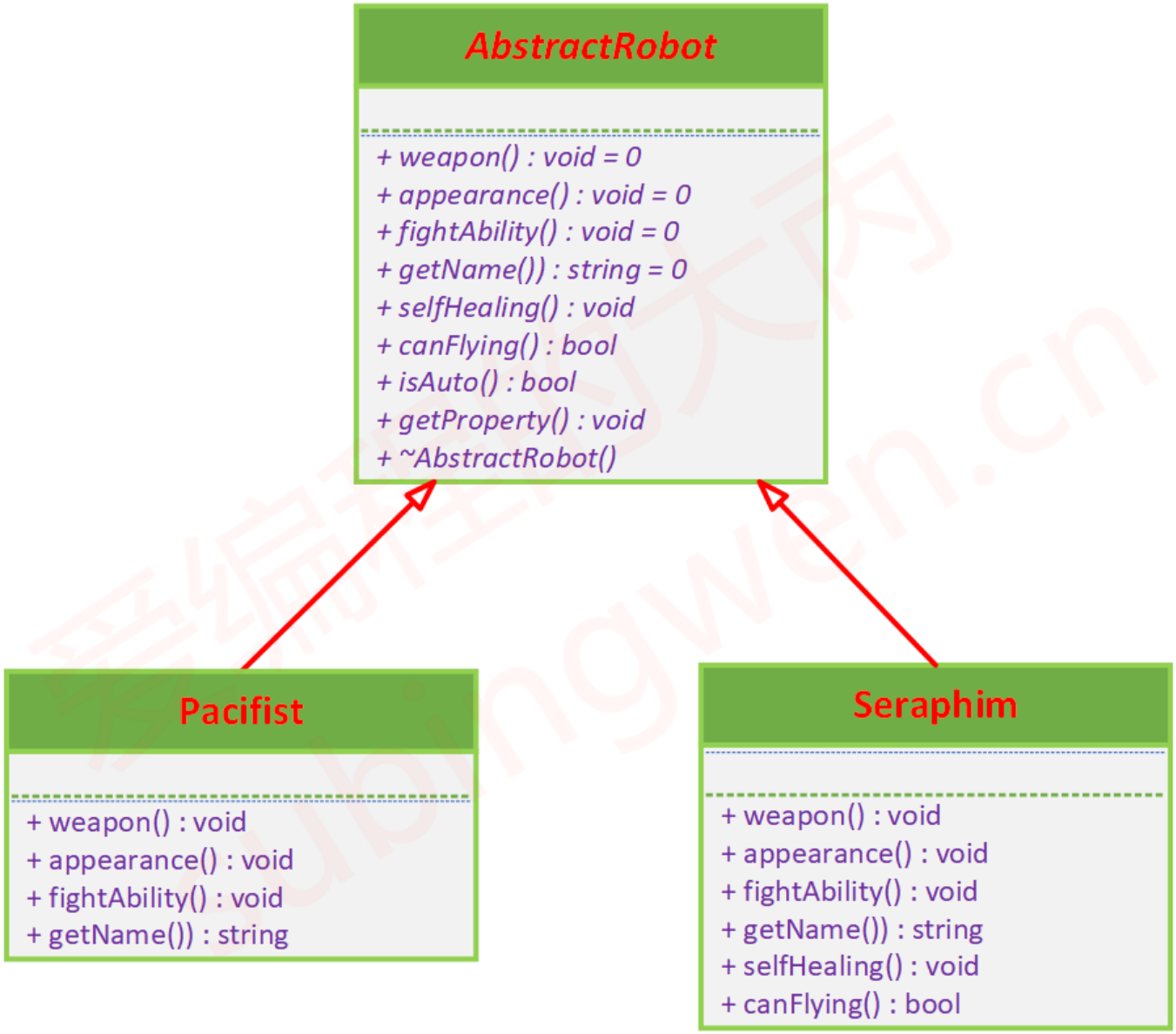
C++

```
贝加庞克制造的和平主义者有以下属性：
没有飞行能力！
可以自动控制，完全体机器人！
可以发射镭射光...
外部和巴索罗米·熊一样，体型庞大，拥有呈半圆形的耳朵，内部似乎金属。
结实抗揍，可以通过手部或者嘴部发射镭射激光，可以融化钢铁！！！
==================================
贝加庞克制造的炽天使有以下属性：
有飞行能力！
可以自动控制，完全体机器人！
可以发射镭射激光，鹰眼外形的炽天使携带者一把巨剑，可以斩断一切！！！
外观和七武海小时候的外形一样，并且拥有一对和烬一样的翅膀！！！
不仅可以发射镭射激光，还拥有七武海的能力，牛逼plus，无敌了！！！！
非常厚实抗揍，并且拥有非常强的自愈能力，开挂了！！！
```

果然是科技改变世界，怪不得世界政府废除了七武海！

3. 结构图

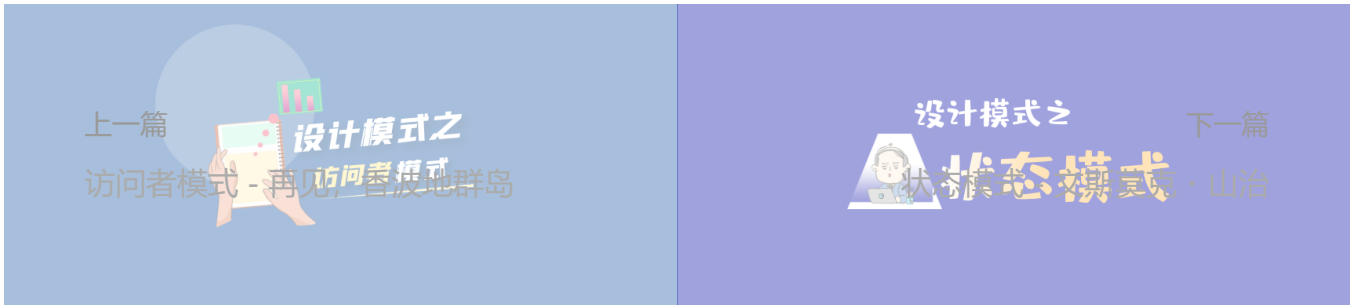
最后将上面的例子对应的 UML 类图画一下（学会了模板方法模式之后，需要先画 UML 类图，再写程序。）



我们在实现子类的时候，如果发现不变的行为和可变的行为混合在了一起，导致不变的行为在多个子类中重复出现，此时就可以使用模板方法模式把不变的行为搬到基类中，去除子类里边的重复代码，来体现它的优势，模板方法模式就是提供了一个很好的代码复用平台。

设计模式

打赏



相关推荐

评论

昵称 邮箱 网址(http://)

来都来了, 说点什么吧...

提交

来发评论吧 ~

Powered By Valine
v1.5.1