

模式定义

BuilderPattern: 将一个复杂对象的构建和他的表示分离，使得同样的构建过程可以创建不同的表示

建造者模式是一步一步创建一个复杂的对象，它允许用户只通过指定复杂对象的类型和内容就可以构建它们，用户不需要知道内部的具体构建细节。建造者模式属性对象创建型模式。

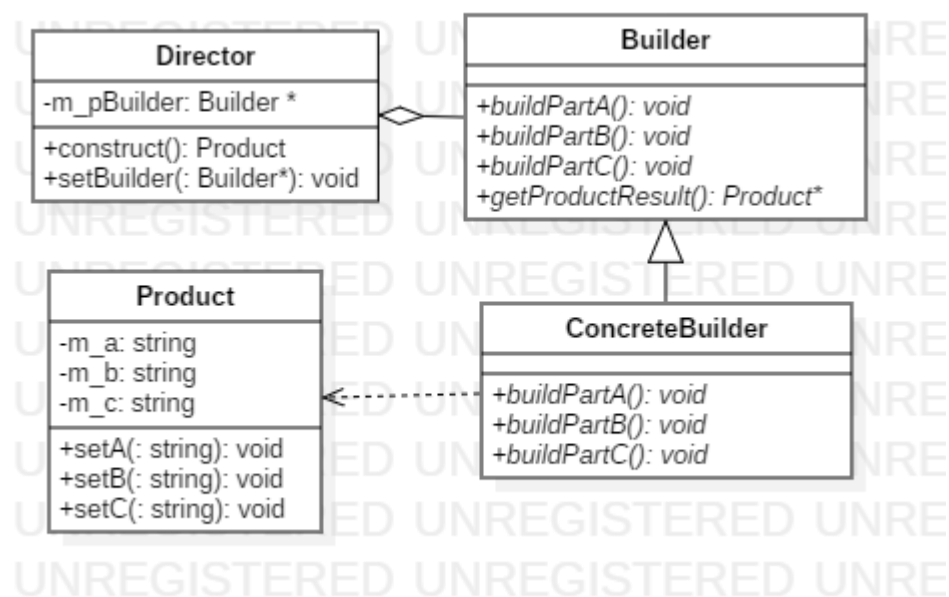
结构模式

Builder: 抽象建造者

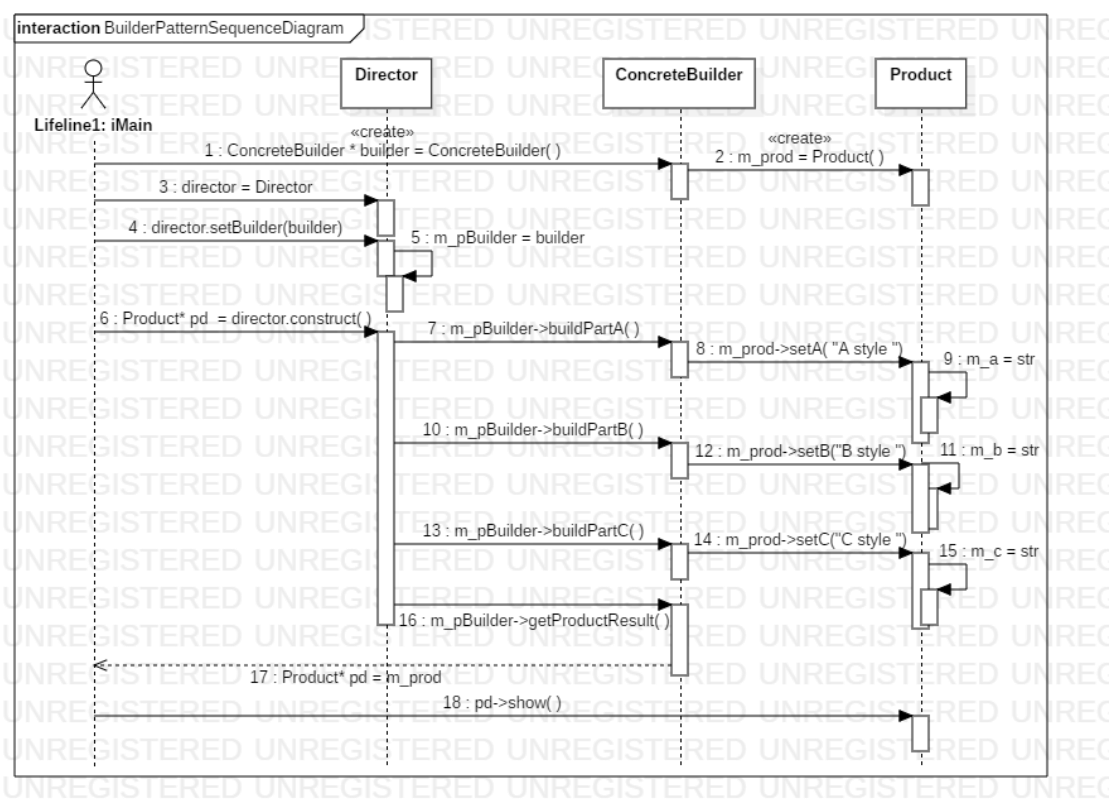
ConcreteBuilder: 具体建造者

Director: 指挥者

Product: 产品角色



时序图



关键代码

```
int main()
{
    ConcreteBuilder * builder = new ConcreteBuilder();
    Director director;

    director.setBuilder(builder);
    Product* pd = director.construct();

    pd->show();

    delete builder;
    delete pd;
}

void Director::setBuilder(Builder * builder)
{
    m_pBuilder = builder;
}

Product * Director::construct()
{
    m_pBuilder->buildPartA();
    m_pBuilder->buildPartB();
    m_pBuilder->buildPartC();

    return m_pBuilder->getProductResult();
}
```

```

void ConcreteBuilder::buildPartA()
{
    m_prod->setA("A style ");
}

void Product::setA(string str)
{
    m_a = str;
}

Product* Builder::getProductResult()
{
    return m_prod;
}

void Product::show()
{
    cout << "product has: " << endl << m_a << endl << m_b << endl << m_c << endl;
}

```

测试结果

```

using namespace std;
int main()
{
    ConcreteBuilder * builder = new ConcreteBuilder();
    Director director;

    director.setBuilder(builder);
    Product* pd = director.construct();

    pd->show();

    delete builder;
    delete pd;

    system("pause");
    return 0;
}

```



```

D:\Design Pattern1\DesignPatterDemo\Cre
product has:
A style
B style
C style
请按任意键继续. . .

```