

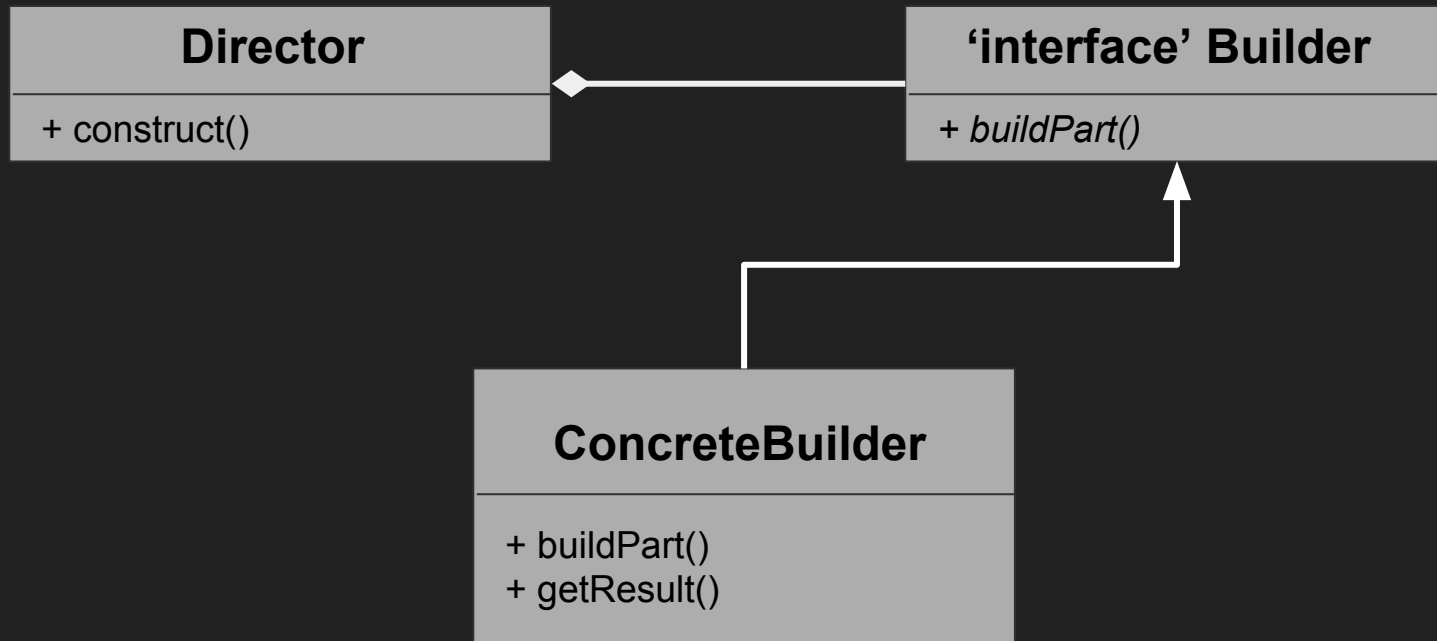
Паттерны проектирования

Порождающие паттерны. Builder, Prototype

Еще немного об оценивании...

Оценка	Базовая часть	Модели интерфейсов	Творческие усилия	Конечный продукт
3 - 5				
6 - 7				
8				
9 -10				

Builder



Builder

```
class CSpaceship {  
    private:  
        unsigned int m_lives;  
        unsigned int m_armor;  
        EWeaponType m_weapon;  
    public:  
        CSpaceship() {}  
        ~CSpaceship() {}  
        void set_lives(unsigned int lives) { m_lives = lives; }  
        void set_armor(unsigned int armor) { m_armor = armor; }  
        void set_lives(EWeaponType weapon) { m_weapon = weapon; }  
};
```

// product

```
class ISpaceshipBuilder {  
    protected:  
        std::shared_ptr<CSpaceship> m_spaceship;  
    public:  
        ISpaceshipBuilder() {}  
        virtual ~ISpaceshipBuilder() {}  
        std::shared_ptr<CSpaceship> get_spaceship() { return m_spaceship; }  
        void create_product() { m_spaceship.reset(new CSpaceship); }  
        virtual void build_lives() = 0;  
        virtual void build_armor() = 0;  
        virtual void build_weapon() = 0;  
};
```

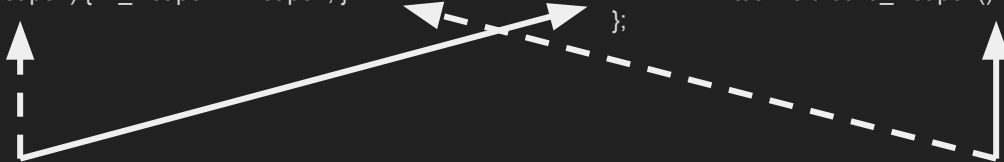
// abstract builder

```
class CEarthSpaceshipBuilder : public ISpaceshipBuilder {  
    public:  
        CEarthSpaceshipBuilder() : ISpaceshipBuilder() {}  
        ~CEarthSpaceshipBuilder() {}  
        void build_lives() { m_spaceship->set_lives(100); }  
        void build_armor() { m_spaceship->set_armor(50); }  
        void build_weapon() { m_spaceship->set_weapon(EWeaponType::Laser); }  
};
```

// concrete builder

```
class CAlienSpaceshipBuilder : public ISpaceshipBuilder {  
    public:  
        CAlienSpaceshipBuilder() : ISpaceshipBuilder() {}  
        ~CAlienSpaceshipBuilder() {}  
        void build_lives() { m_spaceship->set_lives(50); }  
        void build_armor() { m_spaceship->set_armor(100); }  
        void build_weapon() { m_spaceship->set_weapon(EWeaponType::EM); }  
};
```

// concrete builder



Builder

```
class ISpaceshipBuilder { // abstract builder
protected:
    std::shared_ptr<CSpaceship> m_spaceship;
public:
    ISpaceshipBuilder() { }
    virtual ~ISpaceshipBuilder() { }
    std::shared_ptr<CSpaceship> get_spaceship() { return m_spaceship; }
    void create_product() { m_spaceship.reset(new CSpaceship); }
    virtual void build_lives() = 0;
    virtual void build_armor() = 0;
    virtual void build_weapon() = 0;
};
```

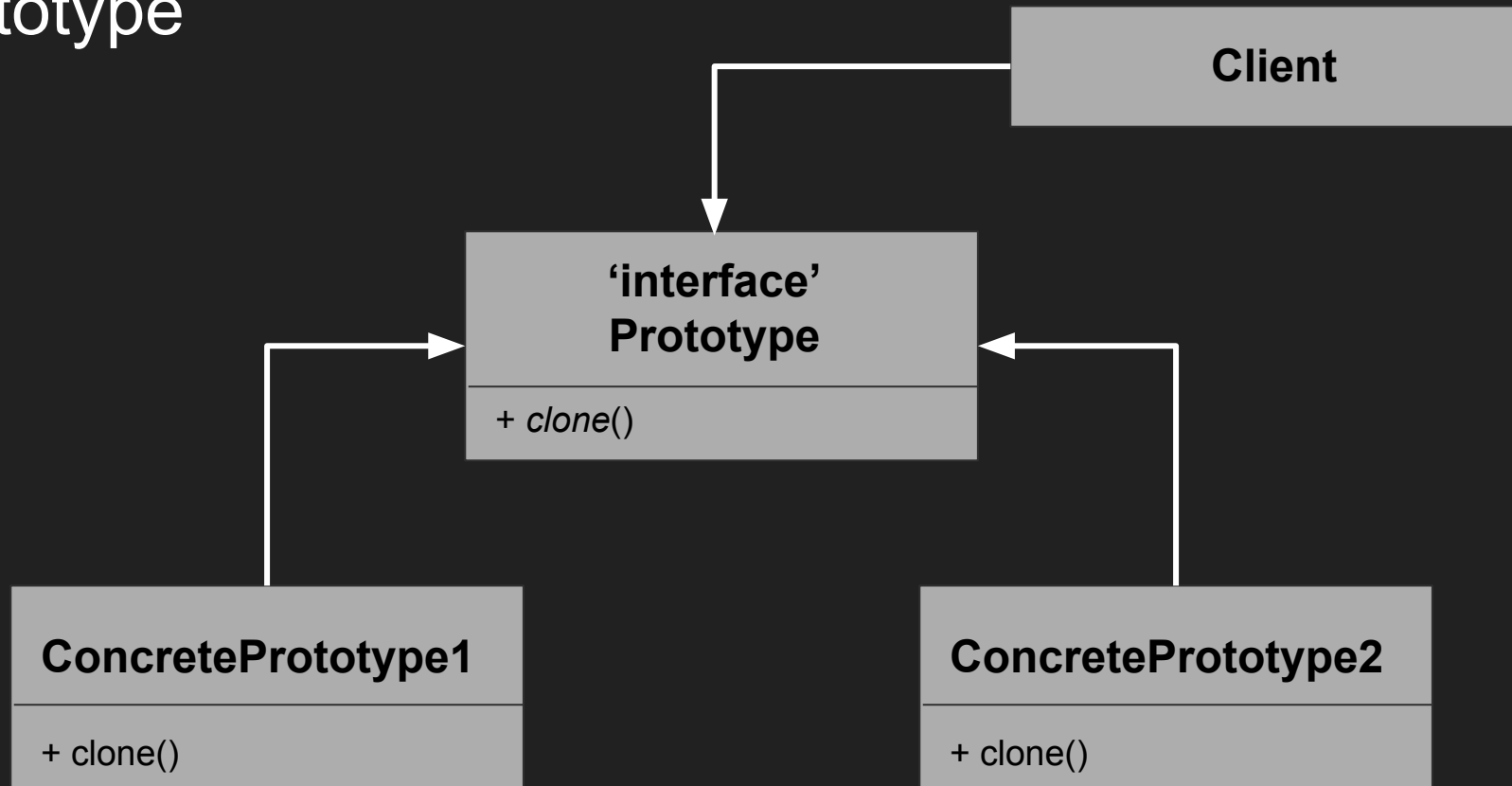


```
class CShipyards { // director
private:
    ISpaceshipBuilder m_spaceship_builder;
public:
    CShipyards() : m_spaceship_builder(NULL) { }
    ~CShipyards() { }
    void set_ship_builder(ISpaceshipBuilder* builder) { m_spaceship_builder = builder; }
    std::shared_ptr<CSpaceship> get_ship() { return m_spaceship_builder->get_spaceship(); }
}

void construct_pizza() {
    m_spaceship_builder->create_product();
    m_spaceship_builder->build_lives();
    m_spaceship_builder->build_armor();
    m_spaceship_builder->build_weapon();
}

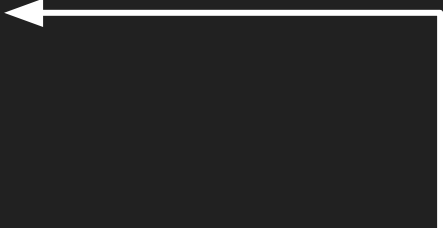
};
```

Prototype



Prototype

```
class ISpaceshipPrototype {  
public:  
    virtual ~CSpaceship();  
    virtual void do_something();  
    virtual CSpaceship* clone() const = 0;  
};
```



```
class CFighter : public ISpaceshipPrototype {  
public:  
    CFighter(const CFighter&);  
    void do_something();  
    CFighter* clone() const { return new CFighter(*this); }  
};
```