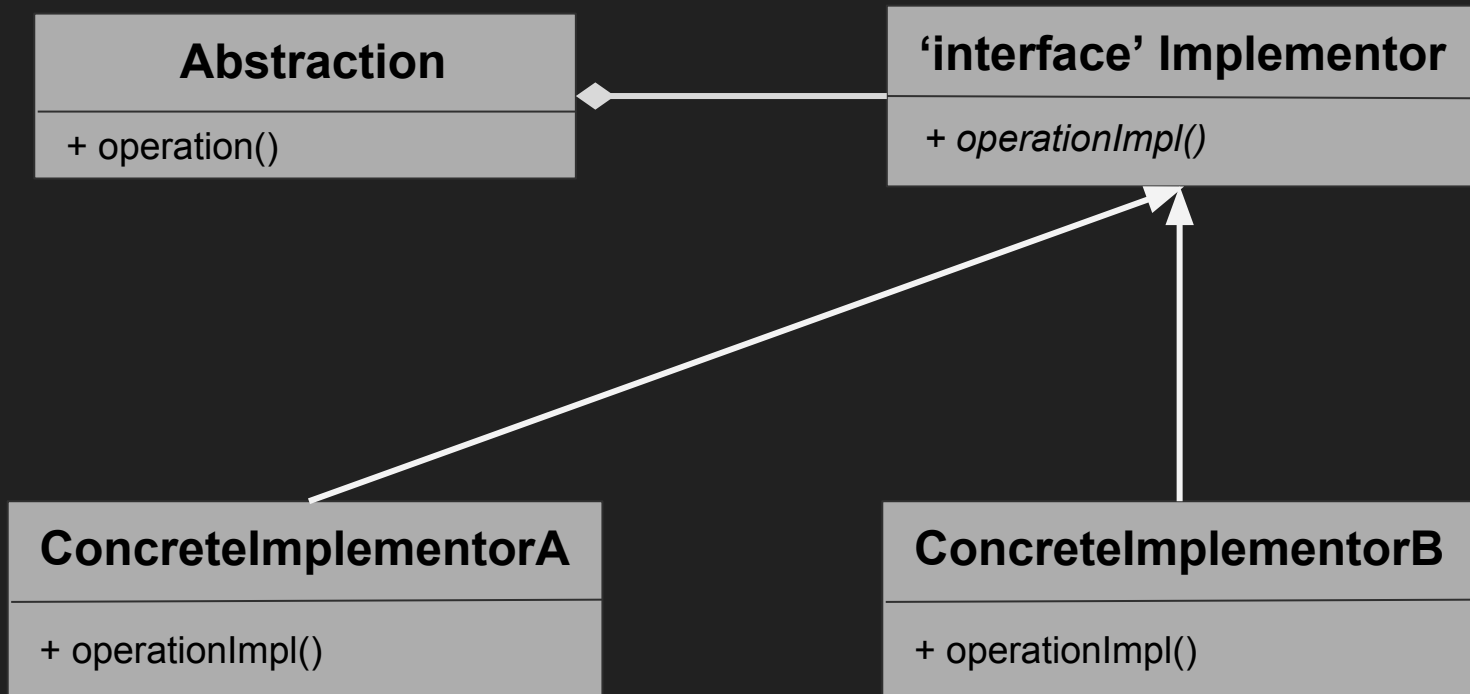


# Технологии программирования

Структурные паттерны. Bridge, Composite.  
Кросс-компиляция

# Bridge



# Bridge

```
class ISpaceshipRace {  
protected:  
    ISpaceshipClass* m_spaceship_impl;  
public:  
    virtual ~ISpaceshipRace () {}  
    virtual void use_special_ability() = 0;  
};
```

// abstraction

```
class CEarthSpaceship : public ISpaceshipRace {  
public:  
    /* implementation */  
};
```

// concrete abstraction

```
class CAlienSpaceship : public ISpaceshipRace {  
public:  
    /* implementation */  
};
```

// concrete abstraction

```
class ISpaceshipClass {  
public:  
    virtual ~ISpaceshipClass() {}  
    virtual void shoot() = 0;  
    virtual void move() = 0;
```

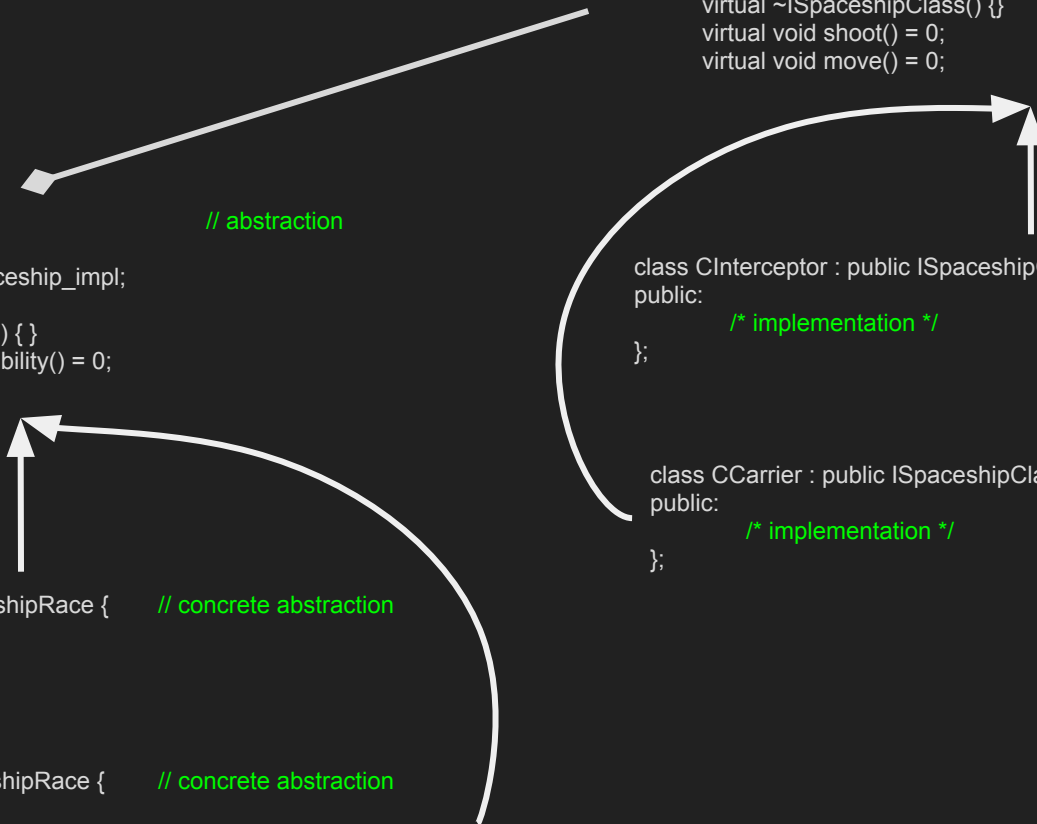
// implementation

```
class CInterceptor : public ISpaceshipClass {  
public:  
    /* implementation */  
};
```

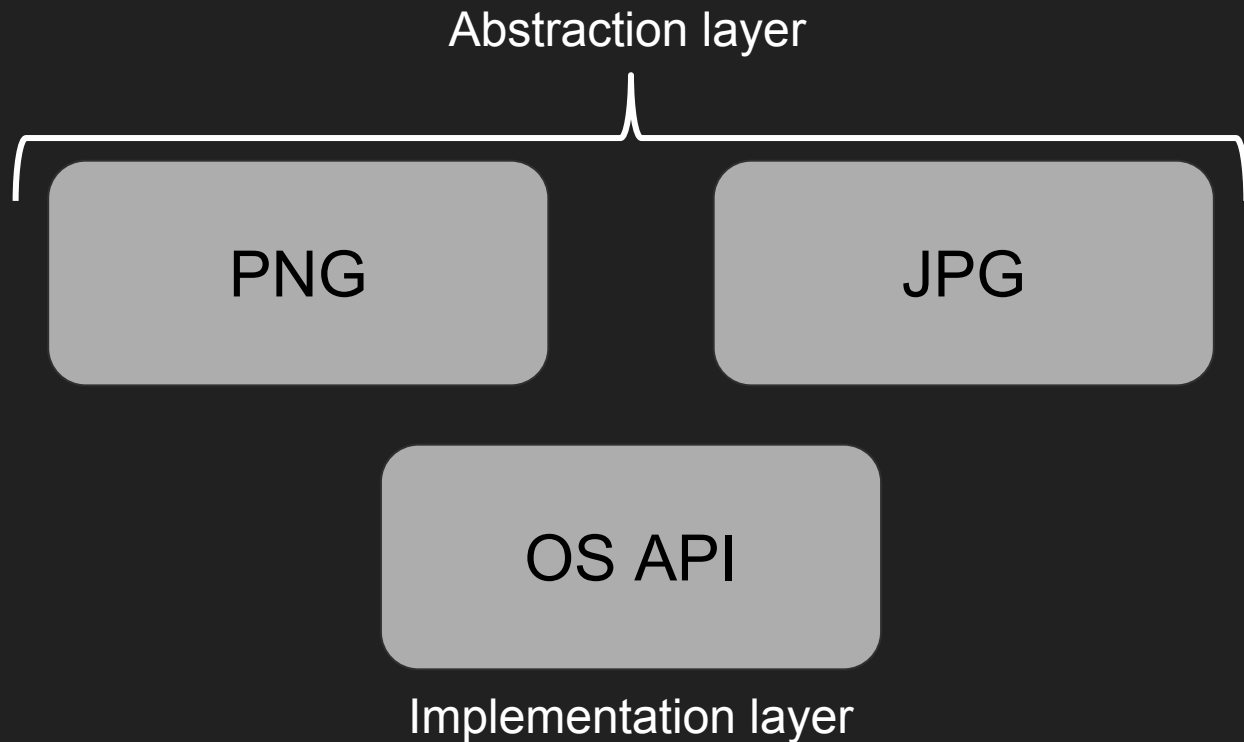
// concrete implementor

```
class CCarrier : public ISpaceshipClass {  
public:  
    /* implementation */  
};
```

// concrete implementor



# Bridge in real life: image formats



# Когда применять?

монолитный класс,  
содержащий несколько  
реализаций

класс нужно расширять  
в двух независимых  
плоскостях

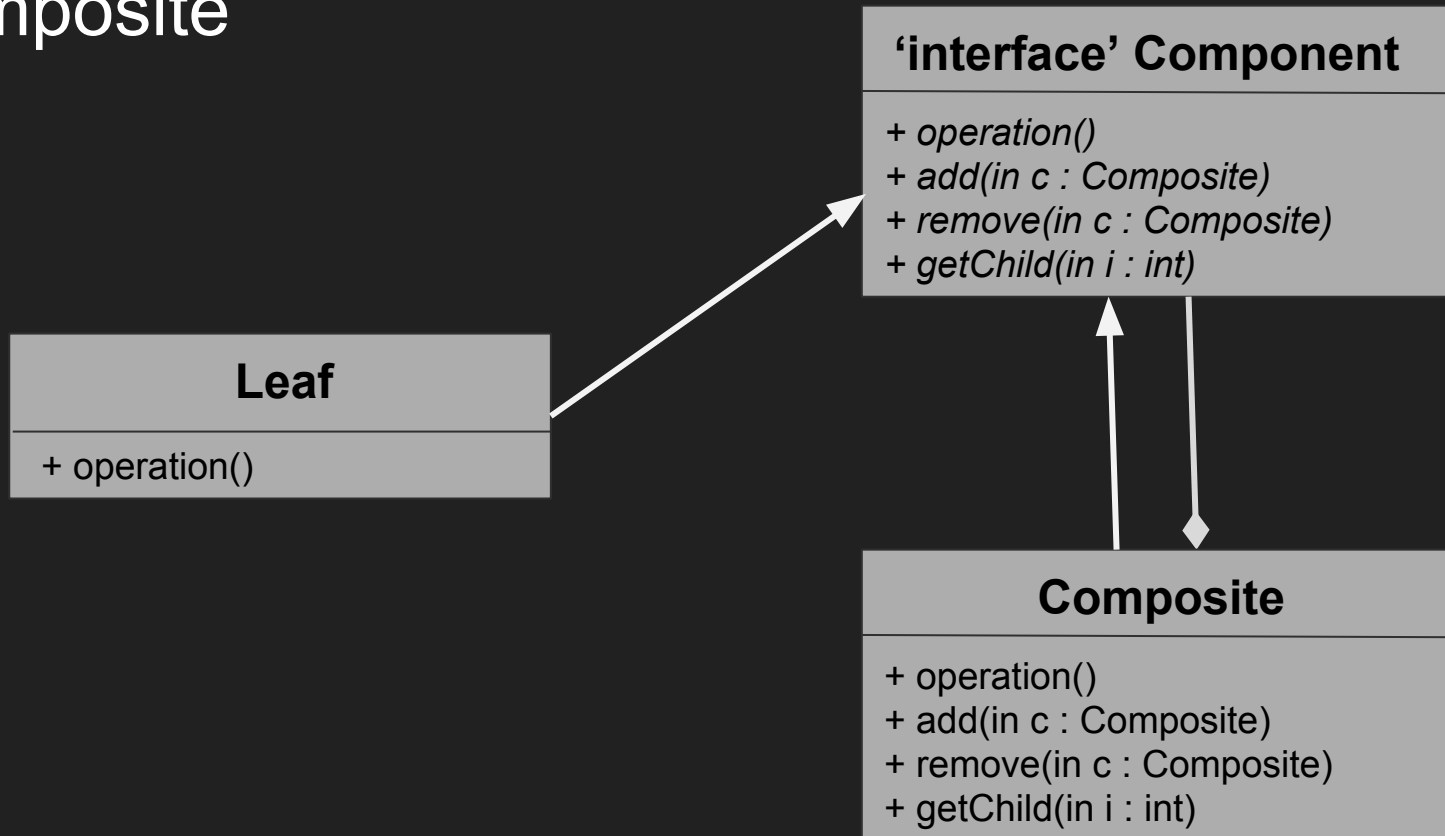
необходимо менять  
реализацию в runtime

# Relations

Abstract factory & Bridge

Builder & Bridge

# Composite



# Когда применять?

нужно представить  
древовидную структуру  
объектов

клиенты должны  
единообразно  
трактовать простые и  
составные объекты



# Composite use cases

GUI

армии

парсер  
математических  
выражений

# Composite in real life: Qt QWidget

```
void QWidget::render(  
    QPaintDevice *target,  
    const QPoint &targetOffset = QPoint(),  
    const QRegion &sourceRegion = QRegion(),  
    RenderFlags renderFlags = RenderFlags( DrawWindowBackground | DrawChildren )  
)
```

# Relations

Iterator & Composite

Builder & Composite

Prototype & Composite

# Кросс-компиляция

Зачем?



```
graph TD; A[Зачем?] --> B[платформы нет в наличии]; A --> C[компиляция на целевой платформе невозможна]
```

платформы нет  
в наличии

компиляция на  
целевой платформе  
невозможна

# Кросс-компиляция



# Кросс-компиляция: типичная подготовка тулчейна



# Кросс-компиляция: собираем тулчейн для ARM

```
$ mkdir ~/arm-toolchain  
$ cd ~/arm-toolchain  
$ tar xaf binutils.tar.gz  
$ tar xaf gcc.tar.bz2
```

# Кросс-компиляция: собираем тулчейн для ARM

```
$ mkdir build-binutils  
$ cd build-binutils  
$ ../binutils/configure \  
    --target=arm-elf \  
    --enable-interwork \  
    --enable-multilib \  
    --with-gnu-as \  
    --with-gnu-ld \  
    --disable-nls  
$ make && make install
```



# Кросс-компиляция: собираем тулчейн для ARM

```
$ mkdir build-gcc
$ cd build-gcc
$ ../gcc/configure
    --target=arm-elf \
    --enable-interwork \
    --enable-multilib \
    --enable-languages="c,c++" \
    --with-newlib \
    --disable-shared \
    --with-gnu-ld \
    --with-gnu-as \
    --with-float=soft \
    --disable-nls
$ make && make install
```