

CS 485

Advanced Object Oriented Design

Design Patterns

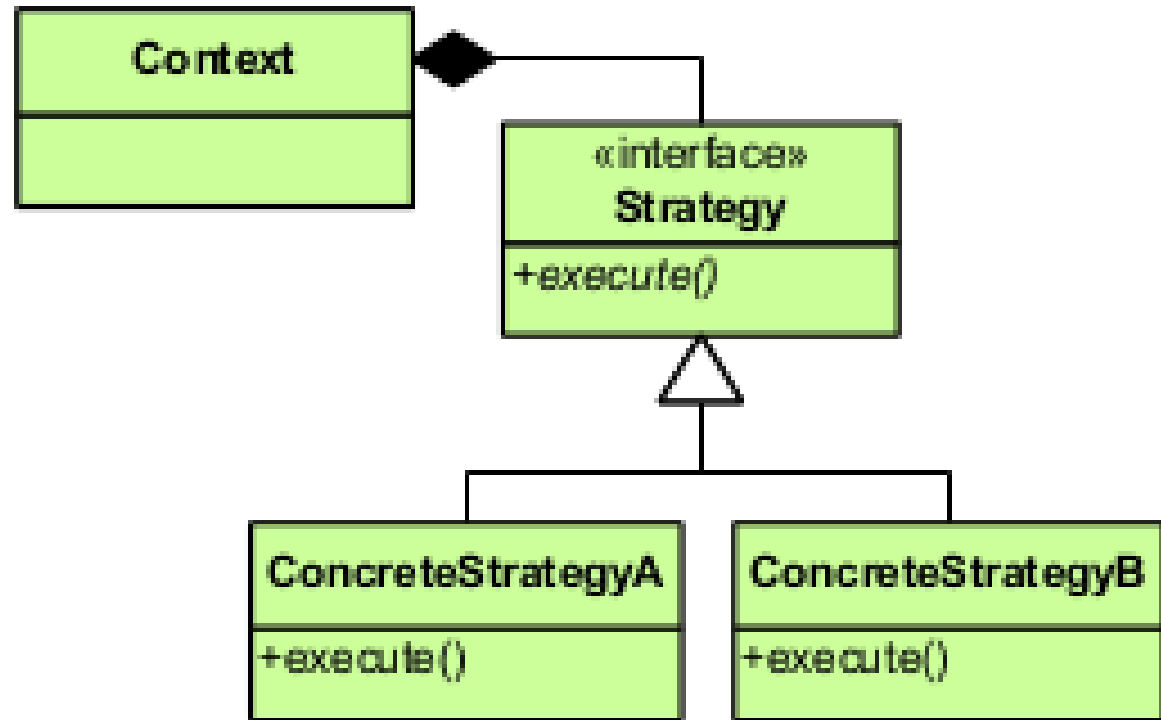
Strategy Pattern

Spring 2019

<http://www.netobjectives.com/PatternRepository/index.php?title=PatternsByAlphabet>

<http://www.netobjectives.com/files/books/dpe/design-patterns-matrix.pdf>

Chapter 9

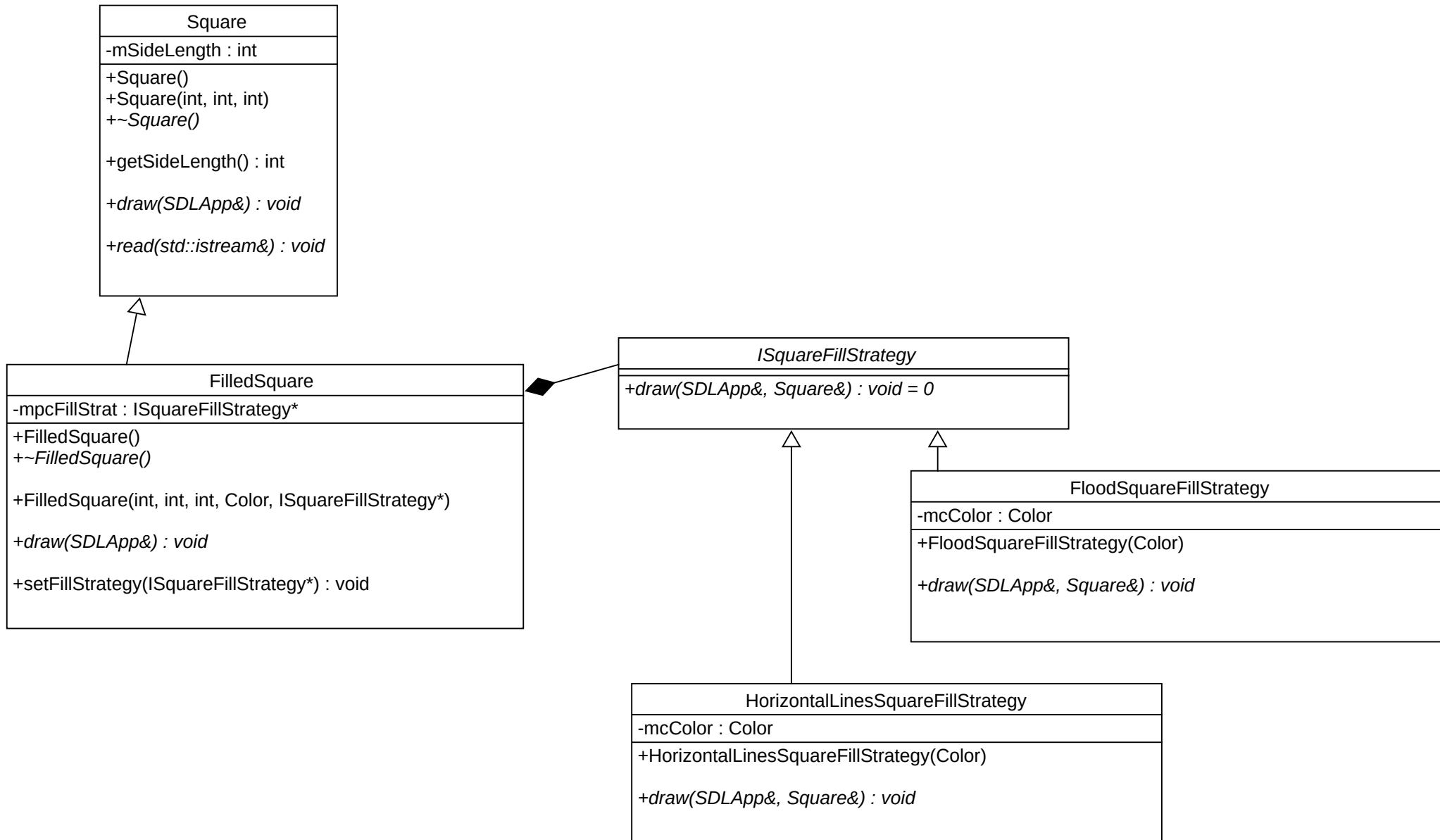


https://en.wikipedia.org/wiki/File:Strategy_Pattern_in_UML.png

Workflow

- Instantiate Context
- Instantiate Concrete Strategy
- Send Concrete Strategy to Context

```
Context cContextObj;  
ConcreteStrategy cConcreteStrat;  
  
cContextObj.setStrategy(cConcreteStrat);  
  
cContextObj.doWork(); // relies on cConcreteStrat's behavior
```



Code

```
void FilledSquare::draw (SDLApp& rcTheDisplay)
{
    mpcFillStrategy->draw (rcTheDisplay, *this);
    Square::draw (rcTheDisplay);
}
```

Strategy

```
class ISquareFillStrategy
{
public:
    virtual void draw (SDLApp& rcTheDisplay, Square &rcSquare) = 0;

private:
};

class HorizontalLinesSquareFillStrategy : public ISquareFillStrategy
{
public:
    HorizontalLinesSquareFillStrategy (Color cColor);

    virtual void draw (SDLApp& rcTheDisplay, Square &rcSquare);

private:
    Color mcColor;
};
```

Code

```
void HorizontalLinesSquareFillStrategy::draw (SDLApp& rcTheDisplay,
Square &rcSquare)
{
    int x, y, sideLen = rcSquare.getSideLength ();

    rcSquare.getPosition (x, y);

    rcTheDisplay.setColor (mcColor);
    for (int newy = y; newy < y + sideLen; newy += 6)
    {
        rcTheDisplay.drawLine (x, newy, x + sideLen, newy);
    }
}
```

Dynamic Strategies

```
void FilledSquare::rightClick (int x, int y)
{
    delete setFillStrategy (new VerticalLinesSquareFillStrategy
    ( cFillColor ));
}
```

```
mcTheCollection.addShape (new FilledSquare (25, x, y, cSquareColor,
    new HorizontalLinesSquareFillStrategy (cFillColor)));
```