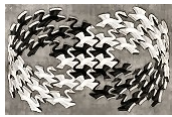


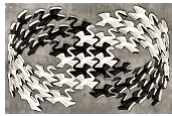
Template method



Motivační příklad – reálný svět

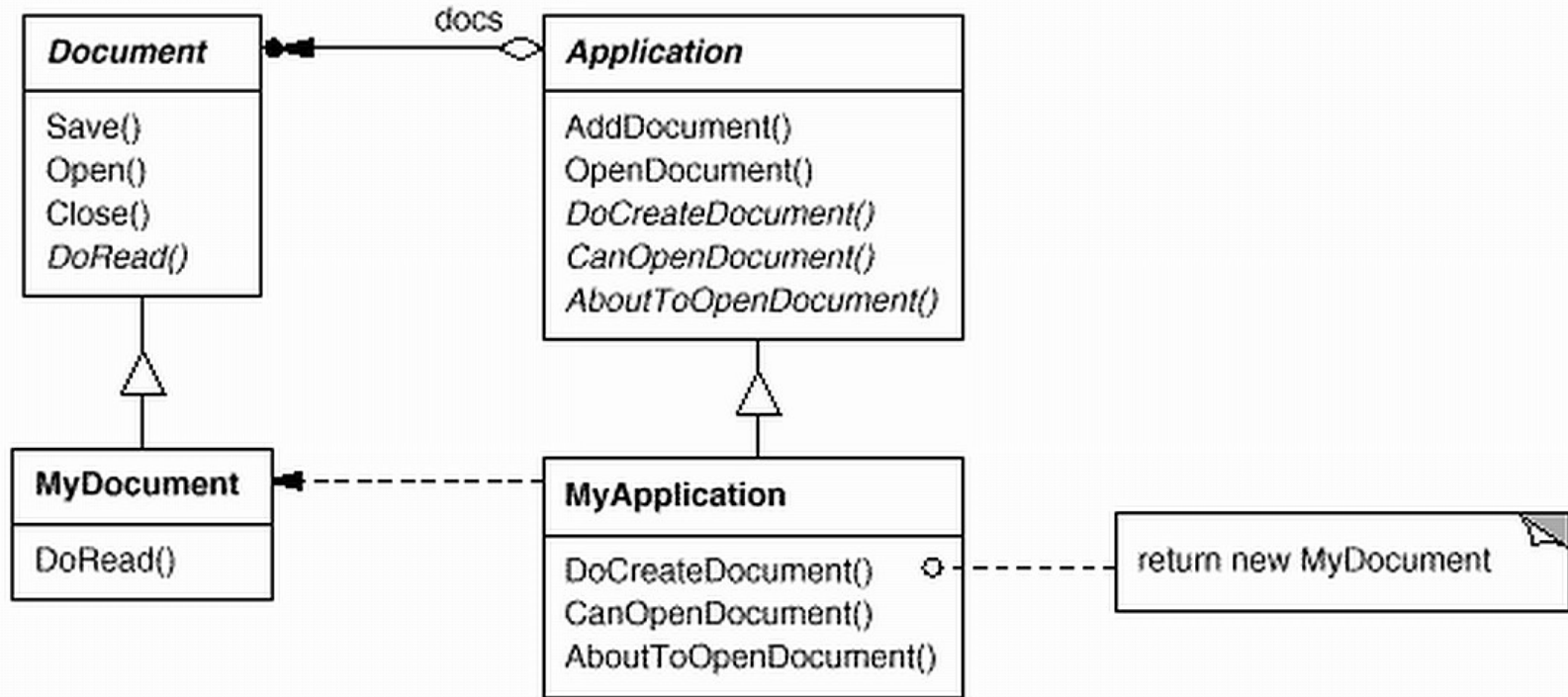
■ Pásová výroba (assembly line)

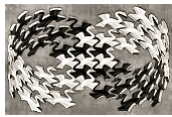




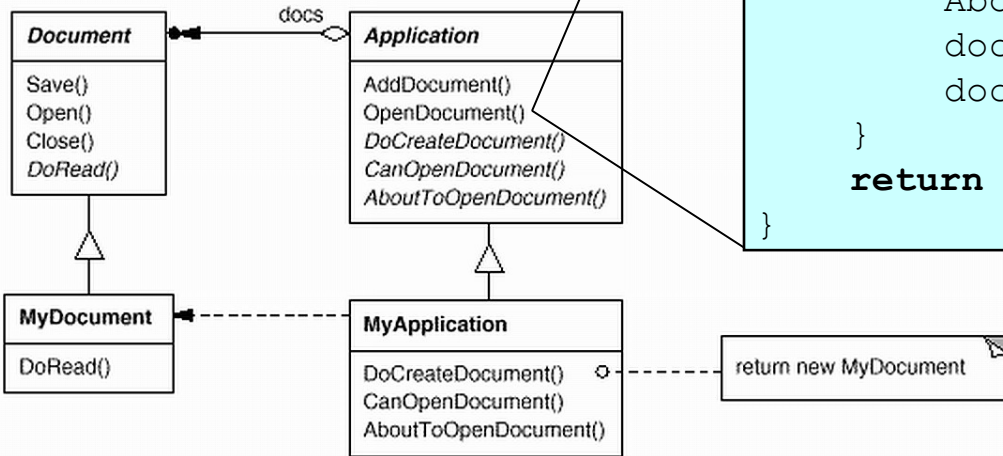
Motivační příklad – SW inženýrství

- Otevírání souborů
- Třídy `Application` a `Document` (+potomci)





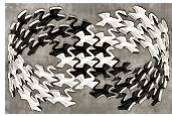
Motivační příklad – SW inženýrství



```
bool Application::OpenDocument() {
    if (!CanOpenDocument(name)) {
        return false;
    }
    Document *doc = DoCreateDocument();

    if (doc) {
        _docs->AddDocument(doc);
        AboutToOpenDocument(doc);
        doc->Open(name);
        doc->DoRead();
    }
    return true;
}
```

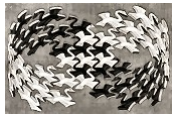
- OpenDocument() – Template method
- DoCreateDocument(), CanOpenDocument() – primitivní operace



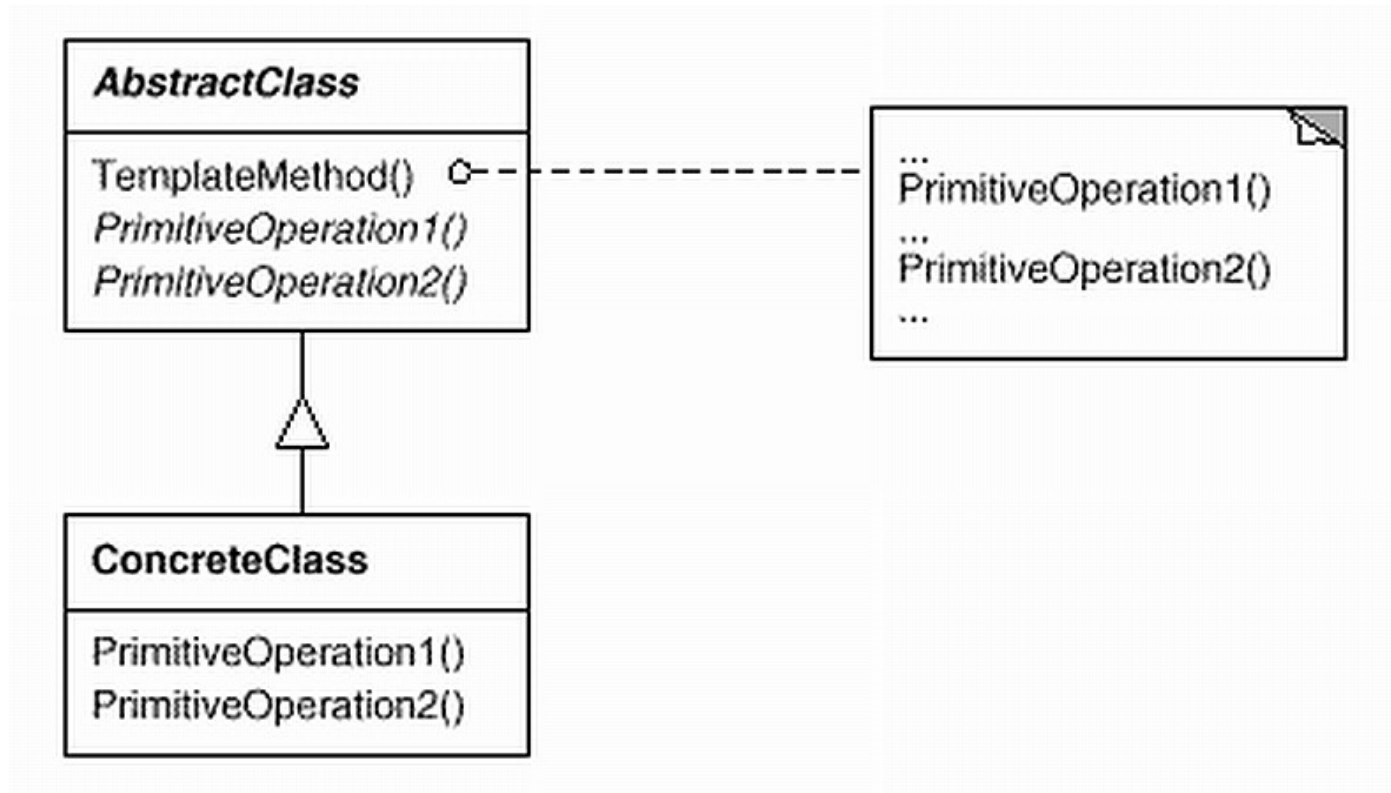
Účel

- **Definice skeletu algoritmu**
 - Pořadí operací
 - Invarianty
- **Agregace totožného kódu množiny tříd**
 - Template method u rodiče
 - Potomkové definují lišící se části
 - „Refactoring to generalize“
- **Řízená dědičnost**
 - Hook operations v Template method

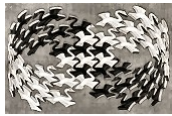




Struktura



- Inverzní princip (Hollywood principle)
 - „Don't call us, we'll call you“



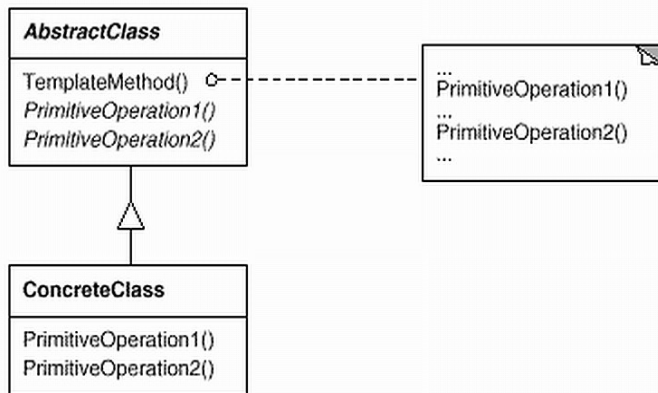
Účastníci

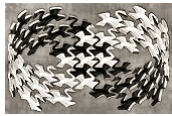
■ AbstractClass

- Definuje abstraktní primitivní operace k dědění
- Implementuje template metodu a definuje skeleton algoritmu
- Template metoda volá primitivní operace ale i další definované v Abstr. classs

■ ConcreteClass

- Implementuje primitivní operace pro změnu algoritmu





Template Metoda volá:

- **Konkrétní operace v ConcreteClass**
- **Konkrétní operace v AbstractClass**

- Invarianty
- `AddDocument()`

- **Primitivní (abstraktní) operace**

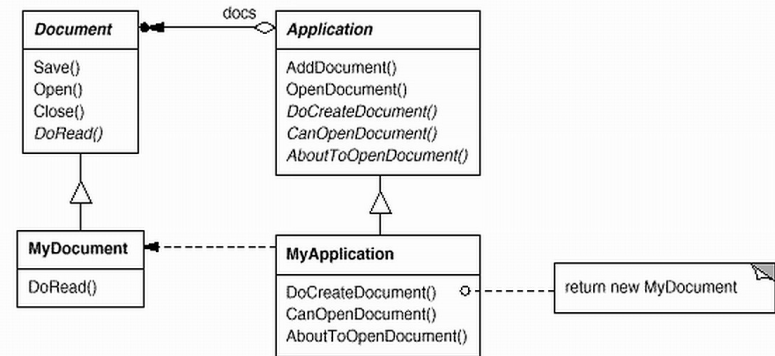
- `CanOpenDocument()`, `DoRead()`
- Implementovány pak v konkrétní třídě

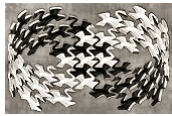
- **Factory method-y**

- `DoCreateDocument()`

- **Hook operations**

- Nepovinné – řízená extenze třídy
- `AboutToOpenDocument()`

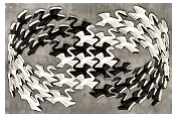




Hook operations

- **Specifická funkcionality `DerivedClass`**
- **Nemusí být implementovány**

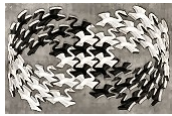
```
void ParentClass::Operation() {  
    // ParentClass behaviour  
    HookOperation();  
}  
  
void ParentClass::HookOperation() {  
}  
  
...  
  
void DerivedClass::HookOperation() {  
    // DerivedClass extension  
}
```



Implementace

- **Přístupová práva a modifikátory OO jazyků**
 - Primitivní operace – `protected`, čistě virtuální
 - Template method – `public`, nevirtuální
- **Minimalizace počtu primitivních operací**
 - Protože
 - to
 - komplikuje
 - implementaci
- **Pojmenovávací konvence**
 - Primitivní operace - prefix „Do–“ (MacApp framework)
 - Hook operations – prefix „Hook–“

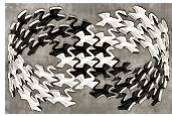




Příklad – NeXT AppKit

- Třída `View` – vykreslování do grafického kontextu
- Před samotným vykreslením potřebuje focus

```
void View::Display() {  
    SetFocus();  
    DoDisplay();  
    ResetFocus();  
}  
  
void View::DoDisplay() {  
}  
  
...  
  
void CustomView::DoDisplay {  
    // render something awesome  
}
```



Příklad – JUnit testy

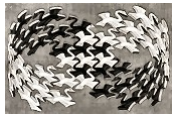
■ Při každém automatickém testu potřeba spustit několik akcí

- ❑ `setUp()` – připraví zdroje
- ❑ `runTest()` – vykoná test
- ❑ `tearDown()` – uvolní zdroje
- ❑ `runBare()` – Template method

```
public abstract class TestCase {  
    public void runBare() throws Throwable {  
        setUp();  
        try {  
            runTest();  
        }  
        finally {  
            tearDown();  
        }  
    }  
  
    protected void setUp() throws Exception {  
    }  
  
    protected void tearDown() throws Exception {  
    }  
  
    protected void runTest() throws Throwable {  
    }  
}
```



JUnit.org



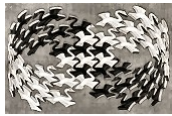
Související NV

■ Factory method

- e.g. `DoCreateDocument()` volána pomocí template metody `OpenDocument()`
- Primitivní operace
- Často volána pomocí template metody

■ Strategy

- Změna celého algoritmu delegací
- Zdědění template metody k částečné změně algoritmu



Template method - Shrnutí

■ Název

- Lehce zavádějící

■ Problém

- Definice obecného algoritmu, jehož struktura zůstane zachována i v jeho konkrétních implementacích

■ Řešení

- Mateřská třída implementuje kostru algoritmu
- Podtřída implementuje jeho jednotlivé kroky

■ Implementace

- Využití přístupových práv a virtuality metod

■ Příklady

- Obecně každý framework/knihovna