

Factory Pattern

Jason Wei

Factory patterns

- Simple factory/Static factory method
- Factory method
- Abstract method

Context

A GUI application to deal with different widgets,
such as button, text...

Widgets

```
class  
    BUTTON  
inherit  
    WIDGET  
...  
end
```

```
class  
    TEXT  
inherit  
    WIDGET  
...  
end
```

Simple factory

class

WIDGET_FACTORY

feature

new_button: BUTTON is

-- New created button

do

create Result

end

new_text: TEXT is

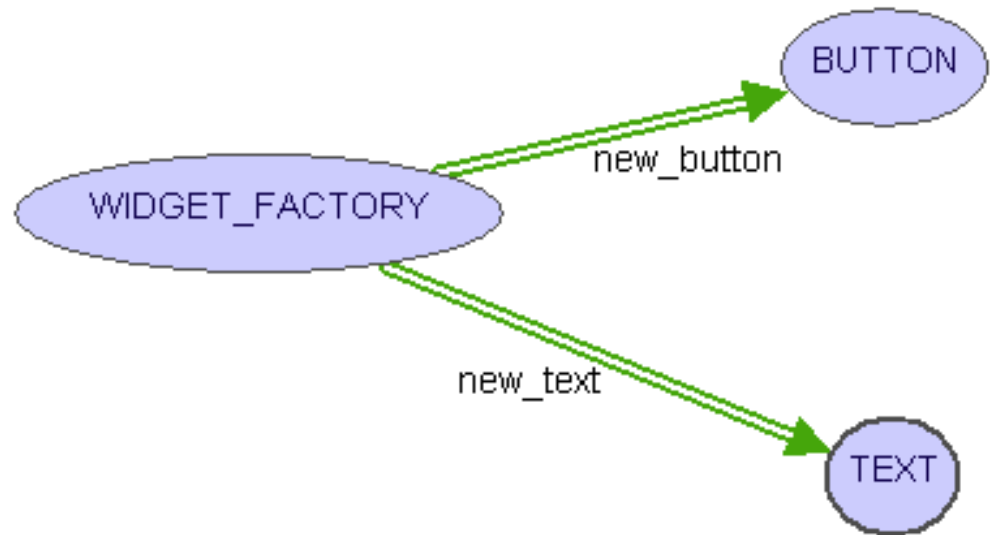
-- New created text field

do

create Result

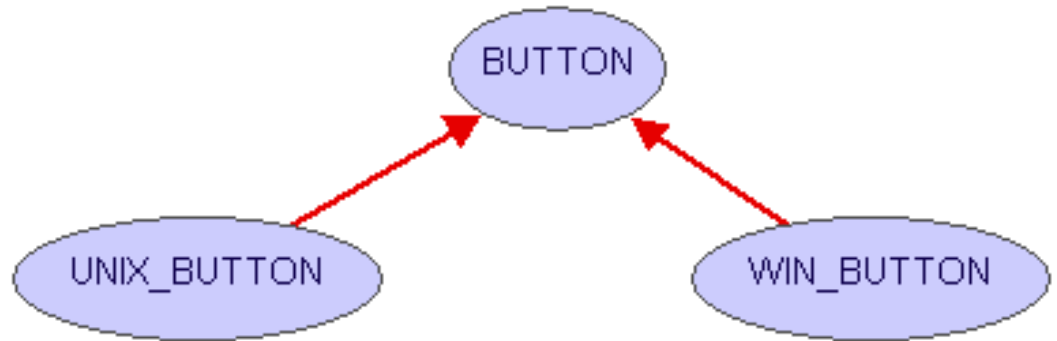
end

end

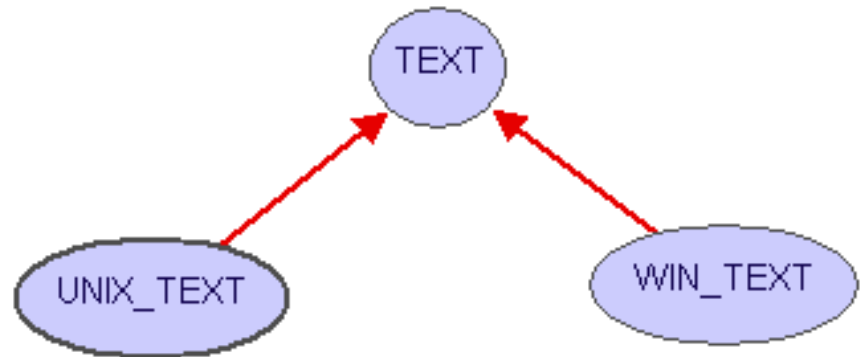


What if we want to support different OSs?

```
class
  UNIX_TEXT
inherit
  TEXT
end
```



```
class
  WIN_TEXT
inherit
  TEXT
end
```



We need to change our factory class

```
class
  MULTI_OS_FACTORY
feature
  new_button: BUTTON is
    do
      if platform.is_windows then
        create {WIN_BUTTON} Result
      elseif platform.is_unix then
        create {UNIX_BUTTON} Result
      end
    end
  end
  new_text: TEXT is do ... end

  platform: PLATFORM

end
```

This shows bad smells of code

Abstract factory

deferred class

ABSTRACT_FACTORY

feature

new_button: BUTTON is

-- New created button

deferred

end

new_text: TEXT is

-- New created text field

deferred

end

end

Concrete factory for UNIX

class

UNIX_FACTORY

inherit

ABSTRACT_FACTORY

feature

new_button: BUTTON is

-- New created button

do

create {UNIX_BUTTON} Result

end

new_text: TEXT is

-- New created text field

do

create {UNIX_TEXT} Result

end

end

Concrete factory for Windows

class

WIN_FACTORY

inherit

ABSTRACT_FACTORY

feature

new_button: BUTTON is

-- New created button

do

create {WIN_BUTTON} Result

end

new_text: TEXT is

-- New created text field

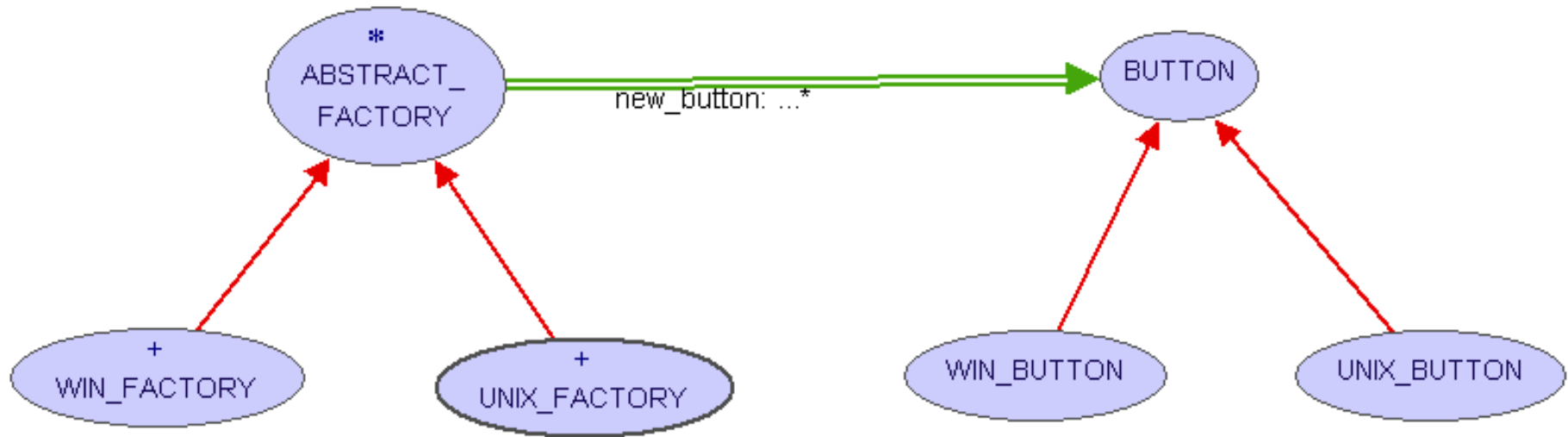
do

create {WIN_TEXT} Result

end

end

Class diagram



Abstract factory

deferred class

ABSTRACT_FACTORY

feature

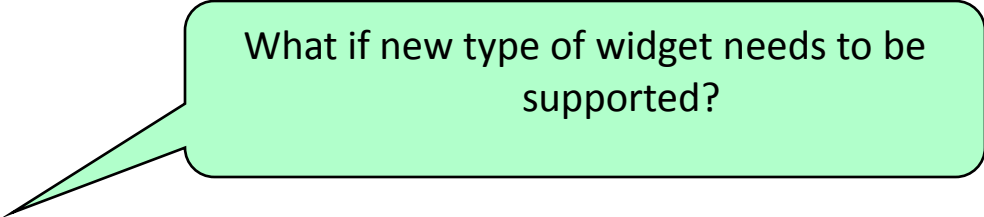
```
new_button: BUTTON is  
  -- New created button
```

```
deferred  
end
```

```
new_text: TEXT is  
  -- New created text field
```

```
deferred  
end
```

```
end
```



What if new type of widget needs to be supported?