

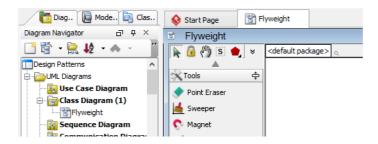
# **Flyweight Pattern Tutorial**

Written Date : October 14, 2009

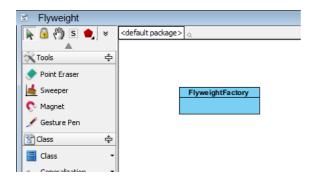
This tutorial is aimed to guide the definition and application of <u>Gang of Four (GoF)</u>flyweight <u>design</u> <u>pattern</u>. By reading this tutorial, you will know how to develop a model for the flyweight pattern, and how to apply it in practice.

## Modeling Design Pattern with Class Diagram

- 1. Create a new project *Design Patterns*.
- 2. Create a class diagram *Flyweight*.



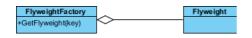
3. Select **Class** from diagram toolbar. Click on diagram to create a class. Name it as *FlyweightFactory*.



4. Right-click on the *FlyweightFactory* class, and select **Add** > **Operation** from the popup menu.

Flyweight	Facto			_
7		Add	•	Attribute Alt+Shift+A
		Open Specification	Enter	Attribute with Getter and Setter
		Stereotypes	•	Operation Alt+Shift+O
		Model Element Properties	+	Constructor
		Sub Diagrams	+	Template Parameter
		Create Parent	+	

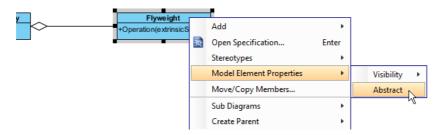
- 5. Name the operation *GetFlyweight(key)*.
- 6. Move the mouse cursor over the FlyweightFactory class, and drag out Aggregation > Class to create an aggregated class Flyweight.



7. Create an operation in Flyweight, name it as Operation and takes an argument extrinsicState.

tFactory		Flyweight
ght(key)	◇	+Operation(extrinsicState)

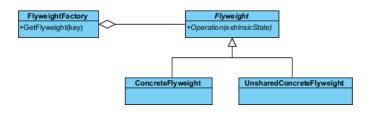
8. Right-click on *Flyweight*, and select **Model Element Properties > Abstract** to set it as abstract.



9. Right-click on *Operation* in *Flyweight*, and select **Model Element Properties** > **Abstract** to set it as abstract.

Preversion (extrinsic)	State)			
		Open Specification	Enter	
		Stereotypes	•	
		Model Element Properties	•	Visibility 🕨
		New Attribute	Alt+Shift+A	Scope 🕨 🕨
		New Operation	Alt+Shift+O	Type Modifier →
		Referenced Diagrams	•	Abstract
	ŵ	Delete		Query
		Dunlicato	CHAIL E	

10. Move the mouse cursor over the *Subject* class, and drag out **Generalization** > **Class** to create subclasses *ConcreteFlyweight* and *UnsharedConcreteFlyweight*.



 Make ConcreteFlyweight and UnsharedConcreteFlyweight inherit the abstract operations provided from Flyweight by right clicking on them, and selecting Related Elements > Realize all Interfaces from the popup menu.

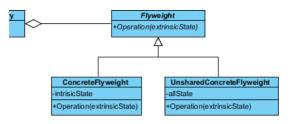
+Operation	(extrinsicState)				
ConcreteFlyweight	UnsharedConcret	eFlyv	eight <sup>©</sup>		
1(	<b>.</b>	2	Add Open Specification Stereotypes	Enter	
		- Edit	Model Element Properties Sub Diagrams Create Parent	• •	Show Dependent Model Elements Navigate To Show Other Views
		X	Cut Copy Delete	*	Show Other Views Show Link Visualize Related Model Element
			Duplicate	Ctrl+E	€ Transit From → Transit To
			Selection Order	*	Create Lifeline
			Grouping Styles and Formatting	*	Of         Create Model Element           Realize all Interfaces
			Alignment Presentation Options	*	Show Associations Analysis
			Layer Layout	+	- Meige Meige from Other Model Element(s) Maria to Model Element
			Define Design Pattern Related Elements	•	Merge to Model Element Change Type
			Java Round-trip C++ Round-trip	*	

<u>r</u>)

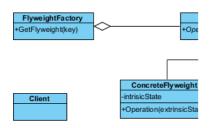
12. Add an operation to *ConcreteFlyweight* by right clicking on *ConcreteFlyweight*, and selecting **Add** > **Attribute** from the popup menu. Name the attribute *intrisicState*.

Y +Oper	 (extrinsic State)	i h-s		
+Operation(extrinsicState	Add	+	Attribute	Alt+Shift+A
	Open Specification	Enter	Attribute with Getter and Setter	
	Stereotypes	+	Operation	Alt+Shift+O
	Model Element Properties	•	Constructor	
	Move/Copy Members		Template Parameter	
	Sub Diagrams	•		
	C		I	

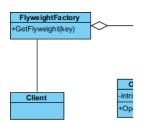
13. Repeat the previous step to add attribute allState to UnsharedConcreteFlyweight.



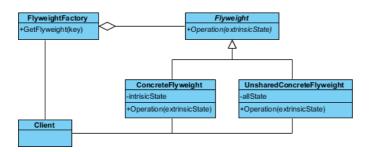
14. Create a class *Client* at the empty region of the diagram.



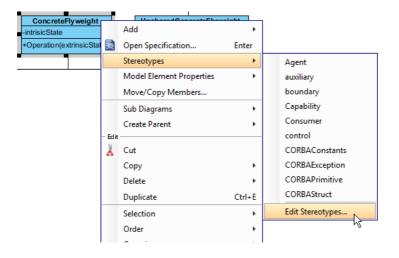
15. Make use of resource centric interface to associate Client and FlyweightFactory.



16. Associate Client with ConcreteFlyweight and UnsharedConcreteFlyweight.



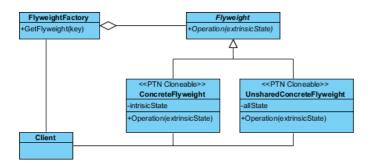
17. In practice, there may be multiple *ConcreteFlyweight* classes. To represent this, stereotype the *ConcreteFlyweight* class as **PTN Cloneable**. Right-click on *ConcreteFlyweight* class and select **Stereotypes > Stereotypes...** from the popup menu.



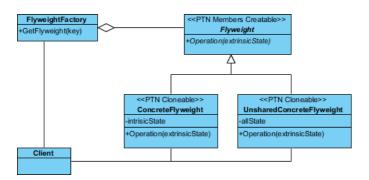
18. In the **Stereotypes** tab of class specification, select **PTN Cloneable** and click > to assign it to the class. Click **OK** to confirm.

<b>\$</b>	Class Spec	ification			×
General Attributes Operations Diagrams Traceability Refere Class Code Details Java Annota All: Message Driven Bean Message Type Message Type Message Type Message ORM Abstract Persistable ORM Component ORM D Generator ORM Parameterized Type ORM Persistable ORM Persistable ORM User Type Participant Sprimitive	Relations	Chart Relations	d Values Co	rameters omments instraints	×
Provider     PTN Members Creatable     ServiceInterface     Session Bean     Struct     type     Typedef     Struct     Edit Stereotypes   Reset	v V	Cancel	Apply	Help	

19. Repeat step 17 and 18 on UnsharedConcreteFlyweight.

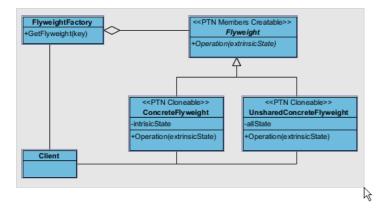


There maybe multiple operations in *Flyweight*. To represent this, stereotype the *Flyweight* class as **PTN Members Creatable**. Repeat steps 17 and 18 to stereotype *Flyweight* as **PTN Members Creatable**. Up to now, the pattern should look like this:

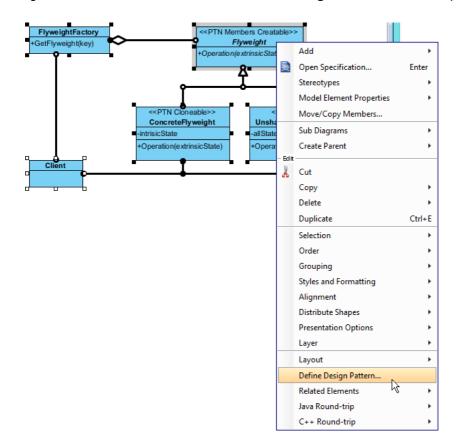


### **Defining Pattern**

1. Select all classes on the class diagram.



2. Right-click on the selection and select **Define Design Pattern...** from the popup menu.



3. In the **Define Design Pattern** dialog box, specify the pattern name *Flyweight*. Keep the file name as it. Click **OK** to proceed.

<b>\$</b>	Define Design Pattern ×
Name:	Flyweight
File name:	Flyweight.pat
Location	
Save	to workspace:
O Save	to directory:
Director	r: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo v
Destination	n: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo\Flyweight.pat
	OK Cancel

#### **Applying Design Pattern on Class Diagram**

In this section, we will try to make use of the flyweight pattern to model a part of a diagram editor.

- 1. Create a new project *My Diagram Tool*.
- 2. Create a class diagram Domain Model.

3. Right-click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.

			-	
	Open Specification			
	Add Shape	•		
	Rename			
	Show Quality Checker Panel			
	Synchronize to Entity Relationship Diagram			
	Ignore Classes when Synchronizing			
(în	Paste View			
	Paste Model Element			
	Handi-Selection	•		
	Diagram Content	•		
	Connectors	•		
	Presentation Options	•		
6	Layers			
	Zoom	•		
	Layout	•		
	Select in Tree			
	Show Link			
	Utilities	•		Apply Design Pattern
٢	Print			Visual Diff
	Export	+		Create Matrix Diagram
				Synchronize Classes Documentation to ERD
				Repair Connector Ends
				Repair Model Views

<b>\$</b>	De	sign Pattern	×
Patterns: Flyweight	FlyweightFactory +GetFlyweight(key)	Control Con	ht
	<	-intrisicState -al	< <ptn cloneable="">&gt; InsharedConcreteFlyweig IState /peration(extrinsicState)</ptn>
	Diagram Element <all></all>		*
	Client	Client ConcreteFlyweight	· +
	Operation extrinsicState	Operation extrinsicState	· · · · · · · · · · · · · · · · · · ·
	intrisicState	intrisicState	~
	Flyweight	Flyweight	~ +
	Operation extrinsicState	Operation	<b>~</b>
	FlyweightFactory	FlyweightFactory	
	😝 GetFlyweight	GetFlyweight	¥
	kev	kev	
Add Re	move		OK Cancel

4. In the **Design Pattern** dialog box, select *Flyweight* from the list of patterns.

5. Click on *Flyweight* in the overview.

FlyweightFactory +GetFlyweight(key)		ers Creatable>> ve <i>ight</i>	^
<	< <ptn cloneable="">&gt; ConcreteFlyweight -intrisicState +Operation(extrinsicState)</ptn>	< <ptn cloneable="">&gt; UnsharedConcreteFlyweig -allState +Operation(extrinsicState)</ptn>	*

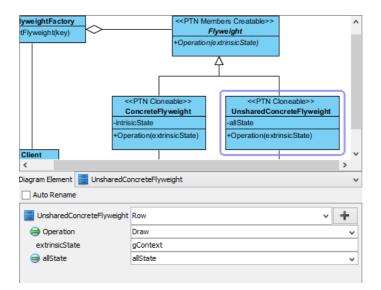
6. Rename it to *Glyph* at the bottom pane. Rename operation *Operation* to *Draw*, and parameter *extrinsicState* to *gContext*.

Flyweight	Glyph	~ <b>+</b>
😂 Operation	Draw	×
extrinsicState	gContext	
	-	

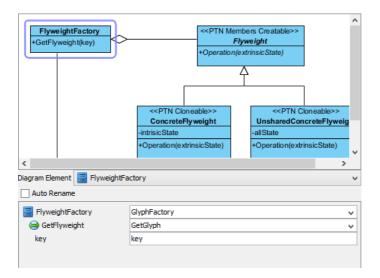
7. Select *ConcreteFlyweight* in overview. At the bottom pane, rename it to *Character*. Rename operation *Operation* to *Draw*, parameter *extrinsicState* to *gContext* and attribute *intrinsicState* to *char*.

FlyweightFactory +GetFlyweight(key)		Cloneable>>
Cliant Diagram Element 📑 ConcreteFly	-intrisicState +Operation(extrinsicState) -allState +Operation(ext weight	trinsicState)
ConcreteFlyweight  Operation  extrinsicState  intrisicState	Character Draw gContext char	* <b>+</b> *

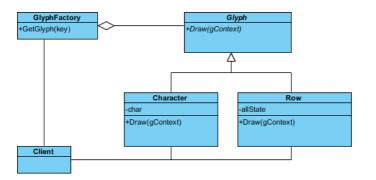
8. Select *UnsharedConcreteFlyweight* in overview. At the bottom pane, rename it to *Row*. Rename operation *Operation* to *Draw*, parameter *extrinsicState* to *gContext*.



9. Select *FlyweightFactory* in overview. At the bottom pane, rename it to *GlyphFactory* and operation *GetFlyweight* to *GetGlyph*.



- 10. Click **OK** to confirm editing and apply the pattern to diagram.
- 11. Tidy up the diagram. It should become:



Resources

- 1. Design Patterns.vpp
- 2. Flyweight.pat

#### **Related Links**

Full set of UML tools and UML diagrams



Visual Paradigm home page (https://www.visual-paradigm.com/)

Visual Paradigm tutorials (https://www.visual-paradigm.com/tutorials/)