

Strategy Pattern Tutorial

Written Date : October 27, 2009

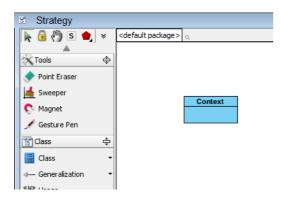
This tutorial is aimed to guide the definition and application of <u>Gang of Four (GoF)</u> strategy <u>design</u> <u>pattern</u>. By reading this tutorial, you will know how to develop a model for the strategy pattern, and how to apply it in practice.

Modeling Design Pattern with Class Diagram

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



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



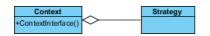
4. Right-click on the Context class, and select Add > Operation from the popup menu.

Context	1				
		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	•		

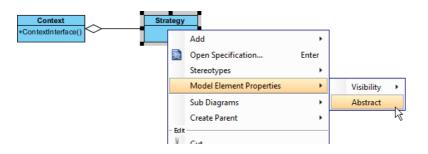
5. Name the operation ContextInterface().



6. Move the mouse cursor over the *Context* class, and drag out **Aggregation** > **Class** to create an associated class *Strategy*.



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



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

Context +ContextInterface()	Strate	gy		~
		Add	•	Attribute Alt+Shift+A
	1	Open Specification	Enter	Attribute with Getter and Setter
		Stereotypes	•	Operation Alt+Shift+C
		Model Element Properties	•	Constructor
		Sub Diagrams	•	Template Parameter
		Create Parent	•	

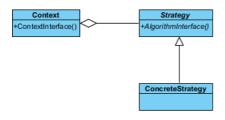
9. Name the operation *AlgorithmInterface()*.



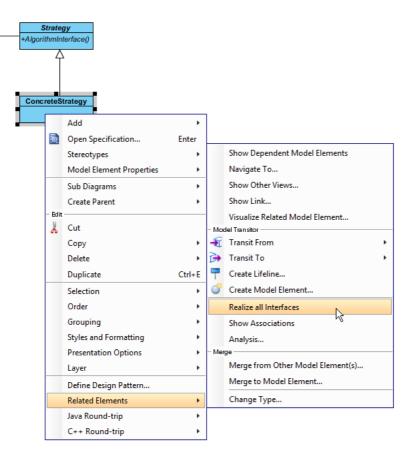
10. Right-click on *AlgorithmInterface*, and select **Model Element Properties** > **Abstract** to set it as abstract.

+AlgorithmInte	-				
	Open Specification	Enter			
	Stereotypes	•			
	Model Element Properties	•	Visi	bility	•
	New Attribute	Alt+Shift+A	Sco	pe	۲
	New Operation	Alt+Shift+O	Тур	e Modifier	۲
	Referenced Diagrams	•	Abs	tract	
Ŵ	Delete		Qu	ery "	
	Duplicate	Ctrl+E			
	Selection	•			
	Formatting	•			
	Code Detail	•			
	Related Elements	•			

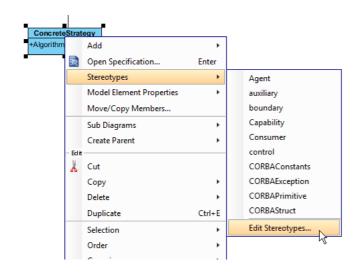
11. Move the mouse cursor over the *Strategy* class, and drag out **Generalization** > **Class** to create subclasses *ConcreteStrategy*.



 We need to make the concrete strategies inherit operations from the strategy class. Right-click on *ConcreteStrategy* and select **Related Elements** > **Realize all Interfaces** from the popup menu.



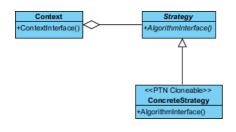
 In practice, there may be multiple concrete strategies. To represent this, stereotype the class *ConcreteStrategy* as **PTN Cloneable**. Right right on *ConcreteStrategy* and select **Stereotypes** > **Stereotypes...** from the popup menu.



14. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *ConcreteStrategy* class. Click **OK** to confirm.

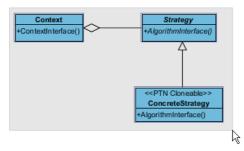
\$			Class Sp	pecificatio	n			×
General Diagrams Class Cod	Attributes Traceabi e Details	Operations lity Refere Java Annota	ences P	ns Chart R Proiect Manao Stereotypes		Templat Quality Values	Comments Constraints	*
Mes meta ORI ORI ORI ORI ORI ORI ORI ORI ORI ORI	sage Driven sageType iclass A bastract Pt I Component I D Generati I Persistable I User Type cipant tive ider Members Cr riceInterface sion Bean ct	ersistable t or zed Type		~	: N Cloneat	ple	*	
Reset			ОК	Cano	el	Apply	Help	

Up to now, the diagram should look like:

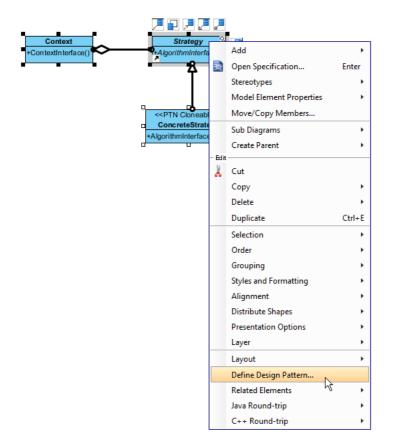


Defining Pattern

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 *Strategy*. Keep the file name as is. Click **OK** to proceed.

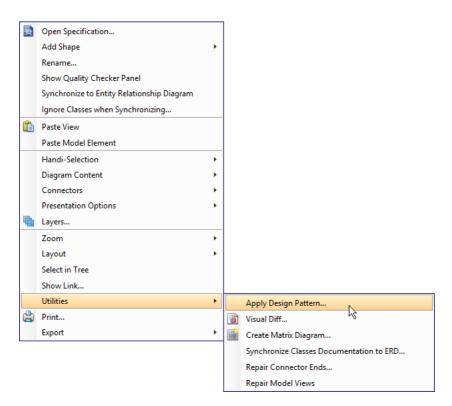
\$	Define Design Pattern	
Name:	Strategy	
File name:	Strategy.pat	
Location		
Save	to workspace:	
O Save	to directory:	
Director	y: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo v	
Destination	n: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo\Strategy.pat	
	UK Califici	

Applying Design Pattern on Class Diagram

In this section, we are going to apply the strategy pattern in modeling a video game.

- 1. Create a new project Game.
- 2. Create a class diagram Domain Model.

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



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

\$	Design Pattern	×
Patterns: Strategy	Context Strategy +ContextInterface() +AlgorithmInterface() < <ptn cloneable="">> ConcreteStrategy +AlgorithmInterface()</ptn>	
	agram Element <all> Auto Rename ConcreteStrategy ConcreteStrategy v</all>	• +
	AlgorithmInterface	-
	Context Context	~
	GentextInterface ContextInterface	~
	Strategy Strategy	×
	AlgorithmInterface AlgorithmInterface	<u> </u>
Add Remove	OK Cance	el

5. At the bottom pane, rename *Context*, *Strategy* and *ConcreteStrategy* to *Game*, *Sprite* and *Warrior*.

Context +ContextInte		Strategy +AlgorithmInterface() < <ptn cloneable="">> ConcreteStrategy +AlgorithmInterface()</ptn>	
<u>D</u> iagram Element	<all></all>		~
Auto Rename			
ConcreteSt	ategy	Warrior 🗸	+
😝 Algorithm	nterface	AlgorithmInterface	~
		-	
Context		Game	~
Context		-	* *
		Game	
⊖ ContextIn Strategy	terface	Game ContextInterface	~

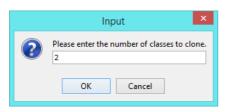
6. Rename operations *AlgorithmInterface* to display.

Warrior	× +]
display	~	-
Game	~	-
ContextInterface	~	-
Sprite	~	-
display	~	
	display Game ContextInterface	display v Game v ContextInterface v Sprite v

7. We need 2 more concrete strategy for *Monster* and *NPC*. Click on the + button at the *ConcreteStrategy* row and select **Clone...** from the popup menu.

itrategy	Warrior 🗸	+
nInterface	display	Clone N
	Game	
Interface	ContextInterface	v

8. Enter 2 to be the number of classes to clone. Click **OK** to confirm.

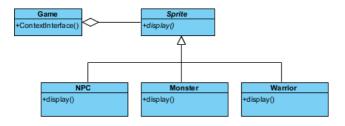


9. Rename *ConcreteStrategy2* and *ConcreteStrategy3* to *Monster* and *NPC*, and operations *AlgorithmInterface* to *display*.

ConcreteStrategy	Warrior	~ +
😂 AlgorithmInterface	display	~
ConcreteStrategy2	Monster	¥
😝 AlgorithmInterface	display	¥
ConcreteStrategy3	NPC	¥
😝 AlgorithmInterface	display	*
📑 Context	Game	*
😝 ContextInterface	ContextInterface	*
📑 Strategy	Sprite	~
😝 AlgorithmInterface	display	¥

10. Click **OK** to apply the pattern to diagram.

11. Tidy up the diagram. Here is the result:



Resources

- 1. Design Patterns.vpp
- 2. <u>Strategy.pat</u>

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/)