TEMPLATE METHOD

Xavier Vallejo López



Descripció

- Situacions en que algun pas de l'algorisme es pot fer de diverses formes.
- El patró suggereix crear un *"template method"* dins d'una *"template class".*
- Dividir en subclasses les implementacions específiques.
- Basat en l'herència i la sobrecàrrega.

Descripció

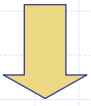
- El mètode del template només conté l'ordre d'execució.
- La classe template pot deixar tota la implementació a les subclasses...

- O definir una part comuna per totes...
- Fins i tot definir implementacions per defecte a cada subclasse.

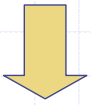
Classe Abstracte

- Adequat quan la classe Template no té implementacions per defecte.

- Diferents implementacions



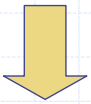
- Mètodes abstractes



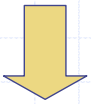
- Subclasses concretes

Classe Concreta

- Adequat quan la classe Template té una implementació per defecte.



- Es dissenya com mètodes **no abstractes** amb impl. Defecte.



- Subclasses modifiquen aquests mètodes.

Exemple: Validació Tarja Crèdit

- Considerem 3 tipus de targes: Visa, MasterCard i Diners Club.

- El mètode final isValid() de la classe abstracta executa tots els passos de l'algorisme.

Exemple: Passos validació

Table 38.1 Different Steps in the Validation Process

Step	*Check*	*Visa*	*MasterCard*	*Diners Club*
1	Expiration date	>Today	>Today	>Today
2	Length	13, 16	16	14
3	Prefix	4	51 through 55	30, 36, 38
4	Valid characters	0 through 9	0 through 9	0 through 9
5	Check digit algorithm	Mod 10	Mod 10	Mod 10
6	Account in good standing	Use custom Visa API	Use custom MasterCard API	Use custom Diners Club API

- Ordre dels passos: 1r: Comuns (1,4,5). 2n: Particulars (2,3,6).

Exemple: Diagrama classes

