

Tema laborator 8

Mostenire II.

In tema de azi veti vedea un alt mod de a aplica mostenirea pentru a refolosi codul. Uneori clasa parinte stie sa execute „in linii mari” un algoritm (ganditi-va la o cautare in graf) dar nu stie detaliile:

Ex: *stie cum sa gaseasca vecinii unui nod sau cum sa calculeze ponderile muchiilor
*stie cum sa faca un quicksort dar nu stie cum sa compare elementele intre ele

Aveti implementata o clasa parinte ce stie sa plimbe un turist intr-o excursie. Din pacate, nu stie ce activitati sa ii dea turistului in fiecare zi. Asa ca implicit, excursia e cam plictisitoare.

Veti folosi mostenirea pentru a completa detaliile algoritmului. (in cazul nostru, completam activitatile din fiecare zi a excursiei)

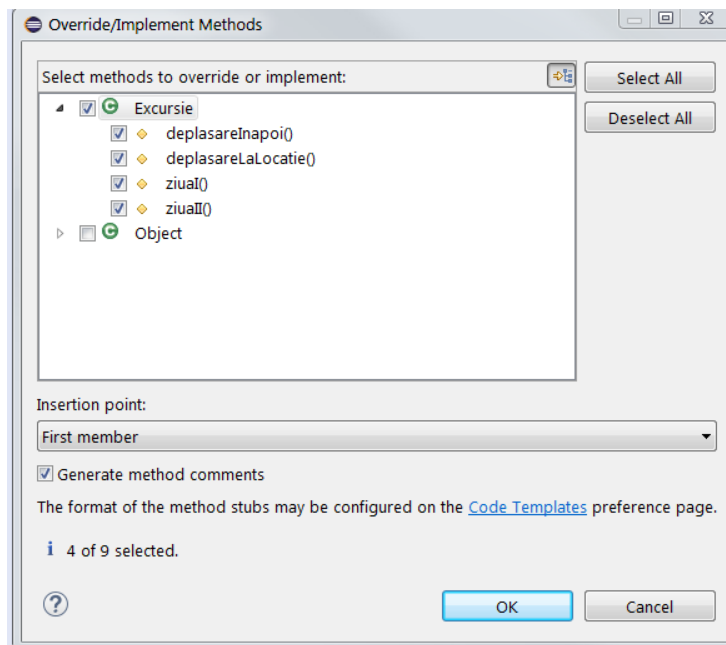
1 Practica

Descarcati proiectul

Faceti o noua clasa ce extinde Excursie

Click dreapta -> Source-> Override/Implement Methods.

Selectati ziuaI, ziuaII, deplasareLaLocatie, deplasareInapoi



La fiecare metoda suprascrisa, scrieti ceva interesant! (Ex: deplasare la locatie, cu avionul sau scufundari la ziua I)

Creati 2 astfel de clase. In main() „vindeti” diverse excursii la clienti. Observati output-ul.

A 2-a clasa sa accepte unul sau mai multi parametri in constructor, in functie de care, se modifica oferta. De exemplu, numarul de activitati din ziua I sau II (ex. in constructor trimiteti parametrul 5 iar turistul va face Bungee Jumping de 5 ori in ziua I. Sau inaltimea de la care va face asta ;).

2 Evaluare

Executati toti pasii din laborator.

3 Referinte:

<http://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html>

http://sourcemaking.com/design_patterns/template_method

http://www.tutorialspoint.com/design_pattern/template_pattern.htm