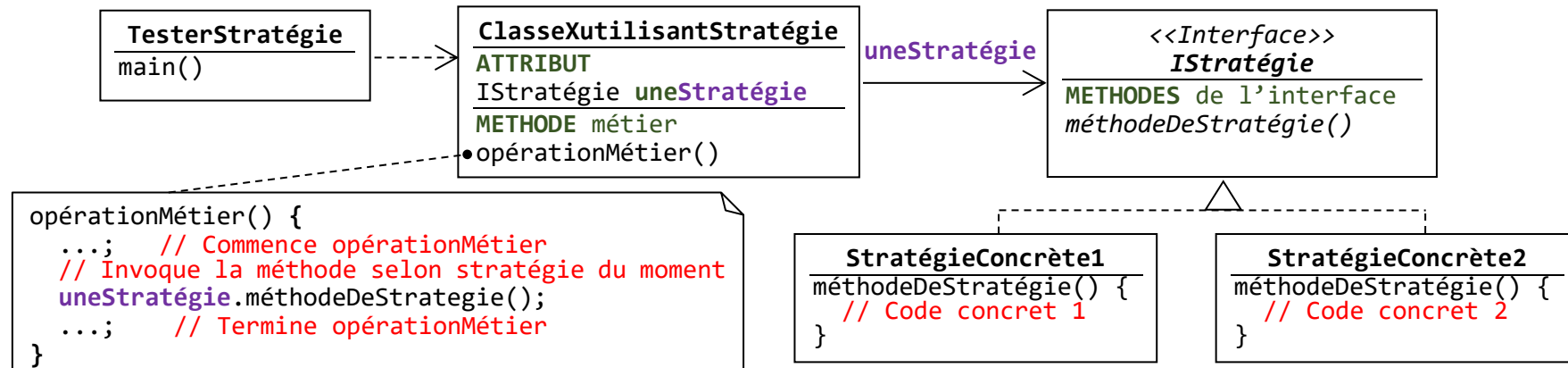


TD-TP : Le Design pattern « Stratégie »

Le design pattern Stratégie permet d'affecter dynamiquement à une opérationMétier() un comportement spécifique/concret différent, selon la stratégie voulue du moment (cf. sélection d'une méthodeDeStratégie() parmi les 'n' classes StratégieConcrète disponibles.



Travail à faire

1. Compléter le schéma ci-dessus avec les objets du main ci-dessous.
2. Créer un projet Eclipse nommé TesterStratégieThéorie dont le **Résultat de l'exécution** voulu est donné ci-dessous. Pour cela, déclarer et coder ClasseUtilisantStratégie, IStratégie, StratégieConcrète1, StratégieConcrète2 et enfin la classe principale TesterStratégie avec son main().

```

int main() {
    ClasseUtilisantStratégie objetX;
    objetX = new ClasseUtilisantStratégie();

    IStratégie uneStratégie;

    uneStratégie = new StratégieConcrète1();
    objetX.setLaStratégie(uneStratégie);
    objetX.opérationMétier(); // Comportement 1

    uneStratégie = new StratégieConcrète2();
    objetX.setLaStratégie(uneStratégie);
    objetX.opérationMétier(); // Comportement 2
}
  
```

Résultat d'exécution du projet TesterStratégieThéorie
 Commencement de l'opération métier
 (résultat du Code Concret 1)
 Termination de l'opération métier

 Commencement de l'opération métier
 (résultat du Code Concret 2)
 Termination de l'opération métier