Classe abstraite

Classe abstraite : une classe est abstraite si elle contient au moins une méthode abstraite.

Classe forme

Public void translation()

{

}

Public abstract void dessiner() // Ne peut être coder de suite, donc abstraite.

{

}

Class Rectangle extends Forme

{

Public void dessiner()

//// Code la méthode abstraite afin de

pouvoir créer des objets Rectangle ////

}

Ex.2 : Entreprise envoyant des cartes de souhaits virtuelles

Public abstract class Carte

{

Public abstract String genererMessage();

}

Class CarteNoel extends Carte

{

}

Class CarteStValentin extends Carte

{

}

* Utile dans la représentation SuperClass / sous-classe, on code les comportements des méthodes communs dans la SuperClass. On établit les comportements essentiels mais spécifiques comme des méthodes abstraites.
* Meilleur structure
* On peut créer des objets dont le type de référence est une classe abstraite.

Ex: Forme f = new Rectangle()

| |

Type de référence type de l'objet

Tableau de forme

Forme[] tab = new Forme[10];

* Aucun objet forme 🡪 classe abstraite
* Ce seront des objets rectangle, triangle, etc.

Tab[0] = new Rectangle();

Tab[1] = new Triangle();

Les interfaces

* Ce n'est pas une classe ----> public interface Modele
* Les interfaces ne contiennent que :
  + Des constantes
  + Des méthodes abstraites
* Les interfaces simulent l'héritage multiple.

Jusqu'à maintenant : 1 superclasse ---> plusieurs sous-classes

* Un seul parent --> héritage simple
* On peut simuler un héritage multiple avec des interfaces car une classe peut mettre en œuvre plusieurs interfaces. (implements)

Ex: class Eric extends Personne implements Client, Prof

Ex: Inventaire d'un entrepôt

Abstract class article

-code

-nom

-prix

Abstract calculerTempsTablette

Class kectchup extends article

-quantité

-coder calculerTempsTablette

-coder ModRemboursement()

Class 2Llait extends Article implement taxable

-quantité

-coder calculerTempsTablette

-coder calculerTaxes()

-coder ModRemboursement()

Public interface taxable

-pourcentage

Abstract double calculerTaxe();

Class CoupeCheveux implements taxable

-coder calculerTaxes()

Public interface Remboursable

Abstract ModRemboursement();