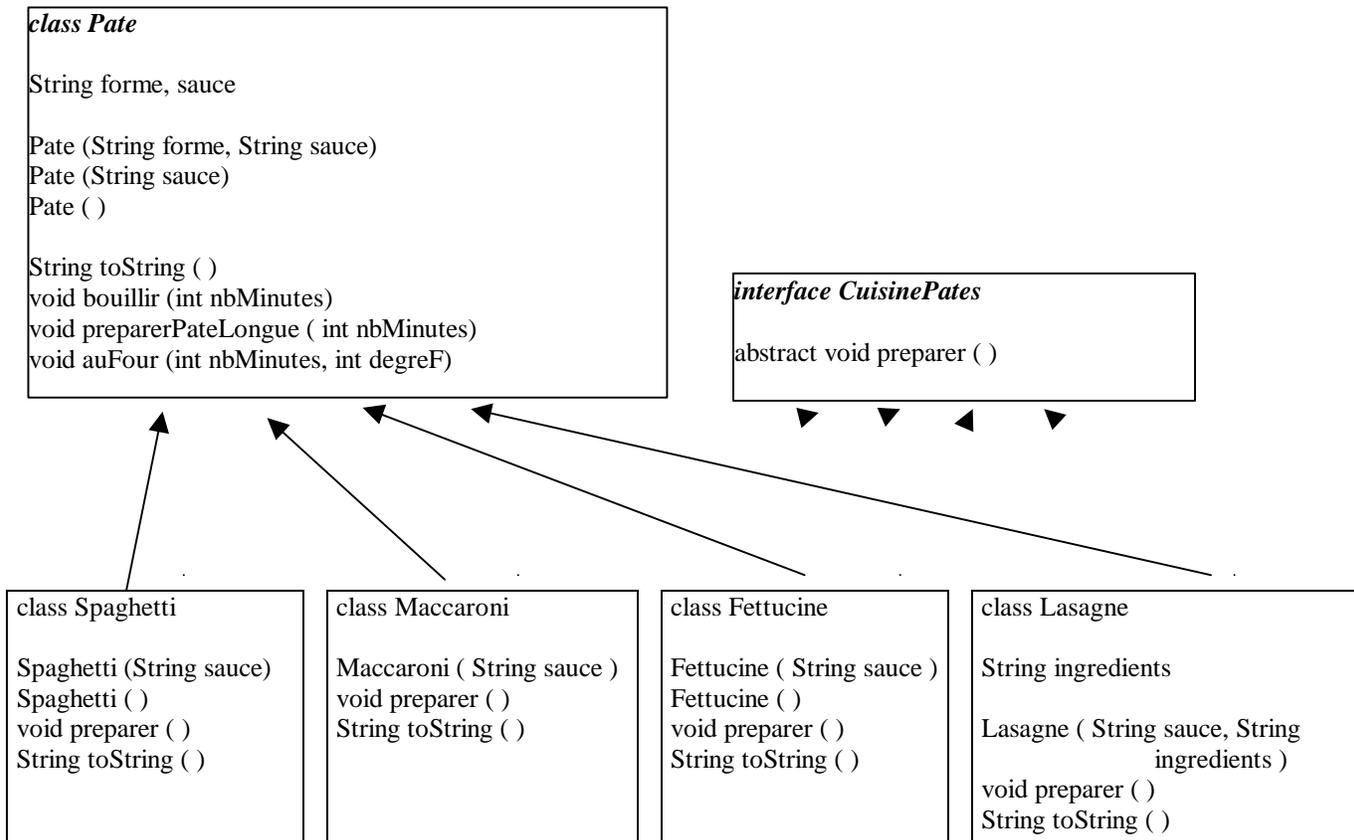


## IFT1010 Démo09 : Hiérarchie de classes, interface, héritage.

Vous devez créer la hiérarchie de classes suivante :



La super-classe **Pate** vous est fournie dans le fichier **Pate.java**

Créer l'**interface CuisinePates** qui déclare la seule méthode abstraite suivante :  
`void preparer();`

Créer aussi les classes **Lasagne**, **Fettucine**, **Maccaroni** et **Spaghetti** qui dérivent la classe **Pate** et implémentent l'interface **CuisinePates** :

Chacune de ces classes doit redéfinir la méthode suivante :

**toString** : retourne un `String` contenant le nom de la classe suivi du message produit par la classe **Pate**  
(voir l'exécution)

## Description des classes :

La classe Lasagne :

Données :

**ingrédients** : un String qui contient les ingrédients, par exemple :  
" pepperoni et fromage "

Méthodes :

**Lasagne** : le constructeur qui reçoit comme paramètres deux String : sauce et ingredients. Ces deux String servent à l'initialisation des variables d'instances sauce (Pate) et ingredients. La valeur de forme est initialisée à autre.

**preparer** : Affiche à l'écran la préparation pour une pâte longue durant 14 minutes, l'ajout des ingrédients puis cuisson au four pendant 30 minutes à 350F (voir exemple d'exécution dans TestsPates.java ).

La classe Fettucine

Méthodes :

**Fettucine** : un constructeur qui reçoit comme paramètre un String : sauce servant à l'initialisation de la variable d'instance sauce (la variable d'instance forme étant initialisée à sa valeur par défaut pour la classe Pate)

**Fettucine** : un autre constructeur sans paramètre qui initialise la variable d'instance sauce à « blanche » et la variable d'instance forme à sa valeur par défaut.

**preparer** : Affiche à l'écran la préparation pour une pâte longue durant 12 minutes

## La classe Spaghetti

### Méthodes :

**Spaghetti** : le constructeur qui reçoit comme paramètre un `String : sauce` servant à l'initialisation de la variable d'instance `sauce` (la variable d'instance `forme` étant initialisée à sa valeur par défaut pour la classe `Pate`)

**Spaghetti** : le constructeur sans paramètre qui initialise les variables d'instances `forme` et `sauce` à leur valeur par défaut.

**preparer** : Affiche à l'écran la préparation pour une pâte longue durant 10 minutes

## La classe Macaroni

### Méthodes :

**Macaroni** : le constructeur qui reçoit comme paramètre un `String : sauce` servant à l'initialisation de la variable d'instance `sauce`, la variable d'instance `forme` étant initialisée à la valeur « a motif ».

**preparer** : Affiche à l'écran la préparation pour une pâte longue durant 9 minutes

Utilisez ensuite la classe `TestsPates` du fichier `TestsPates.java` que nous vous fournissons pour tester le bon fonctionnement de vos classes.

Cette dernière contient en commentaire un exemple d'exécution de votre programme.

Vous n'êtes toutefois pas tenu à reproduire exactement ce résultat.

Utilisez autant que possible ce qui dans la classe de base, vos classes dérivées ne devraient traiter que leurs particularités.