Examen mi-session 21 octobre à 9h
4 devoirs->15% 1 projet->20% (4 phases)
Projet : App mobile(2 implémentations, 2 documentations)

12 septembre 2017

Qualité : UI->utilisabilité
temps de réponse = efficacité
maintenance = robustesse
performance
flexibilité
réutilisabilité

Catégories : Sur mesure(client spécifique)
Générique(tout le monde)
Embarqués(dans le matériel électronique)
À temps réel(réaction immédiate)
Traitement de données(audio,vidéo,texte->sortie)

Modèle en Cascade(90's) : Analyse du domaine->Élicitation des exigences->Design->Implementation->Testing->Déploiement ← =/=changement
S'il y a un changement dans la boucle il faut changer tout ce qui suit!
Deux cas où il peut être utilisé aujourd'hui:
1.Si on on déjà une première version(v.1 → v.2)
2.Armé

GÉNIE LOGICIEL: Résolution de problème posés par un client

AGILE:
→ utilisateur(sollicité/impliqué en tout temps)

Architecture à couche: utilisabilité et flexibilité
1.UI->
2.BL(logique)
3.DL->Base de donnée(DB)

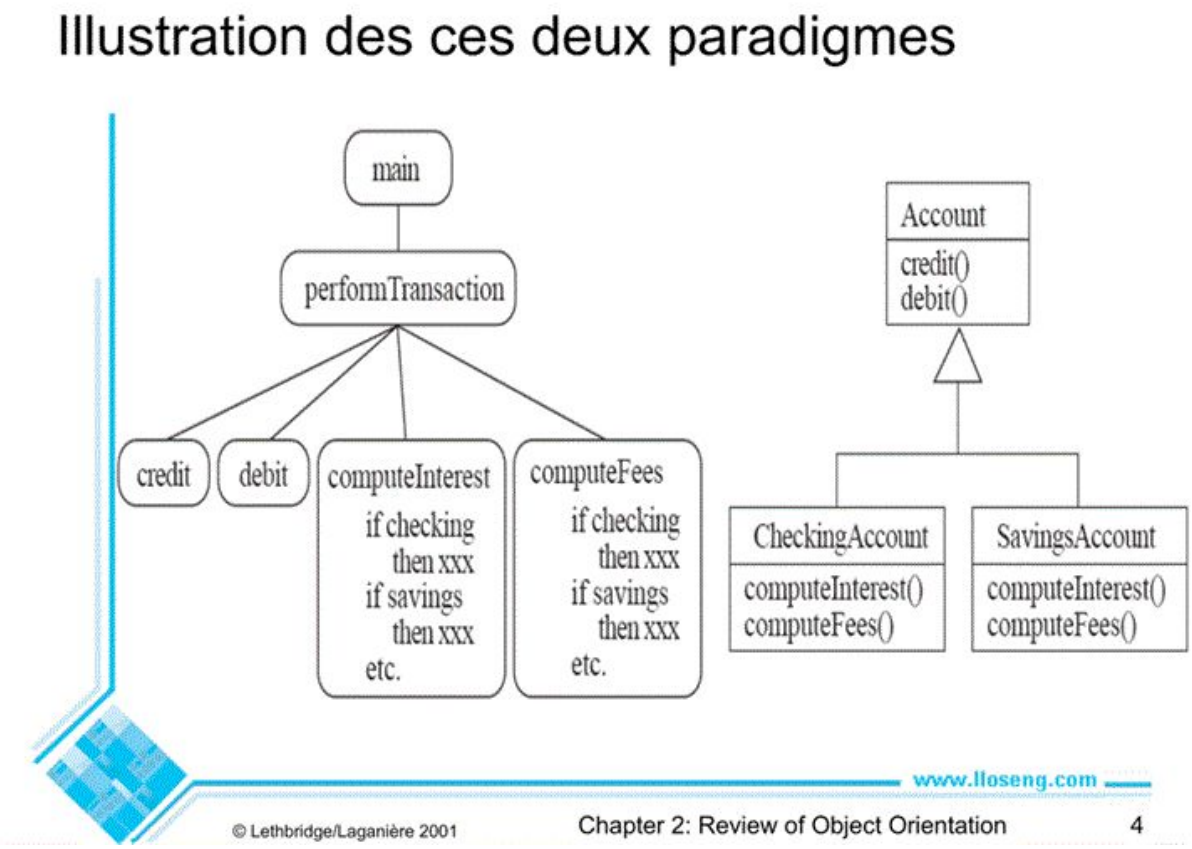
qualité du logiciel:
convivialité, efficacité, fiabilité, facilité de maintenance, réutilisabilité

14 septembre 2017
Génie Logiciel:
Résumé
-Client(Utisateur(centre du projet))
-Standards(IEEE)(universel)
-Contrats(Temps réels(rétroaction))

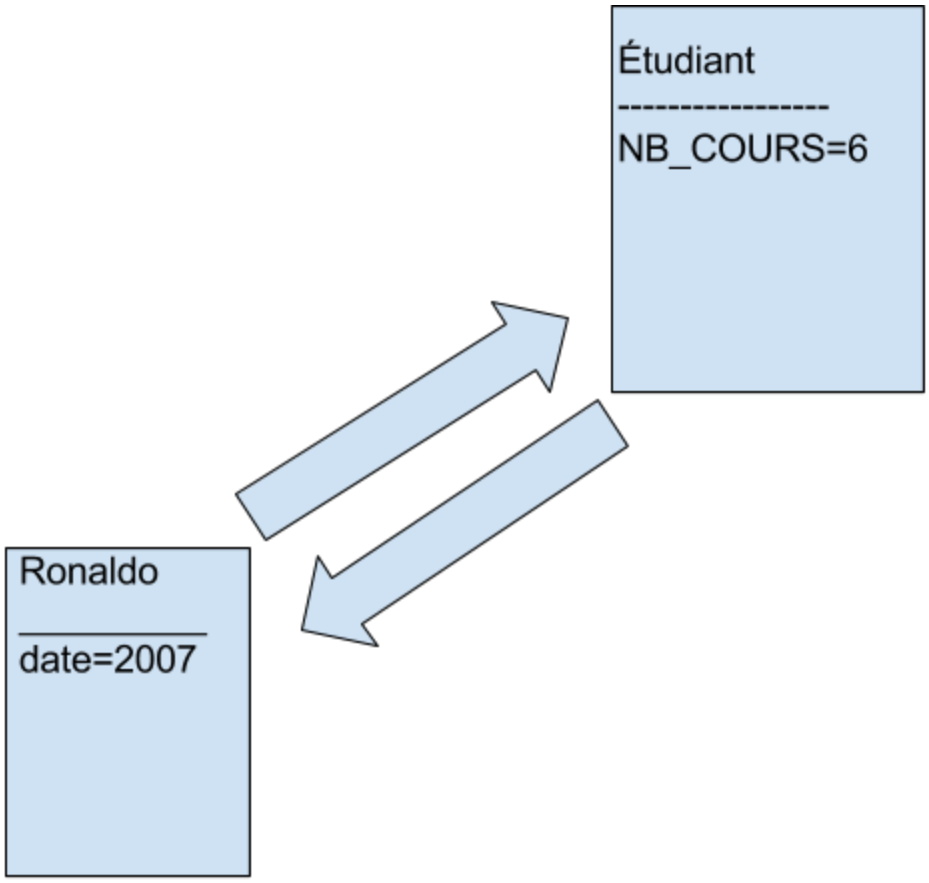
projet Existant:
Correctif/adaptif/refactoring
ou qui n'existait pas(à partir de 0/green field)

Qualité:
utilisabilité/efficacité/

Chapitre 2
Langage orienté objet
procédural vs oo

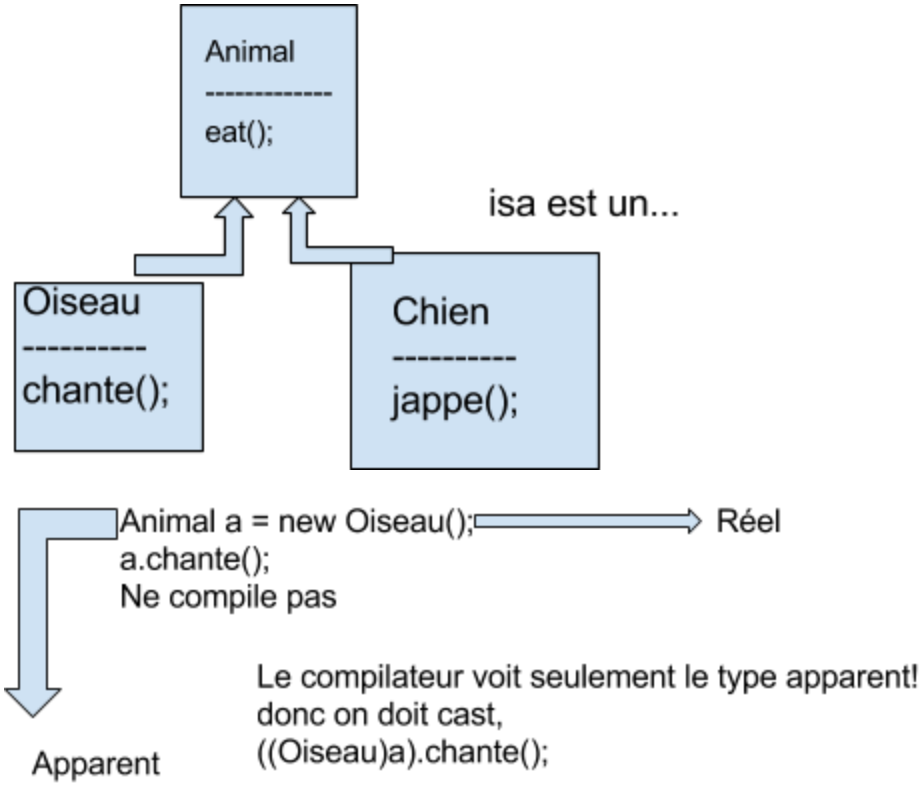


un objet: ensemble structuré de données s'exécutant dans un logiciel.
instance: spécifique(deux étudiant(deux instance de la classe étudiant)).
classe: unité d'abstraction dans un programme oo.
nom de classe débute trjs avec une majs!
propriété d'un objet(-nom ->(dans une classe valeur d'instance))
comportement d'un objet(-inscrire->(dans une classe méthodes))
Étudiant
Valeur()
Instance | Classe
unique partagé par toutes les instances

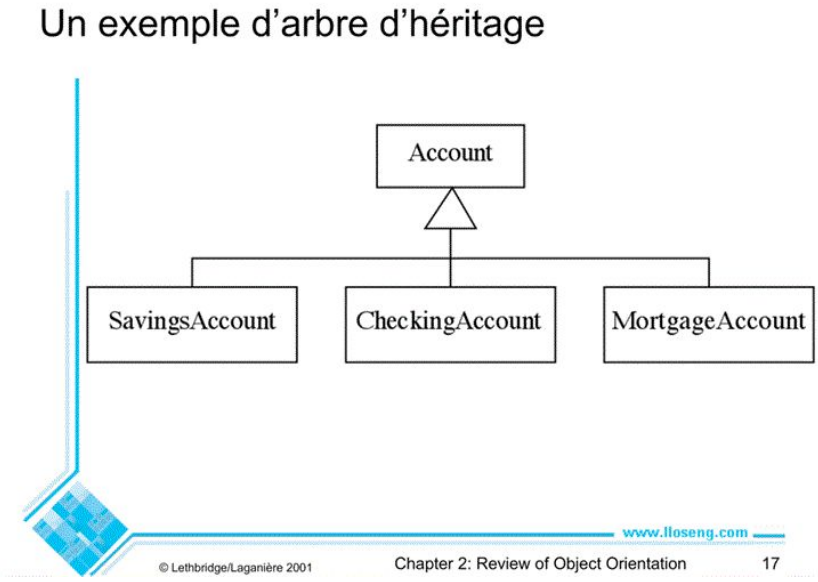


Étudiant
NB_COURS=5;
Etudiant e = new Etudiant();
e.NB_COURS=6;

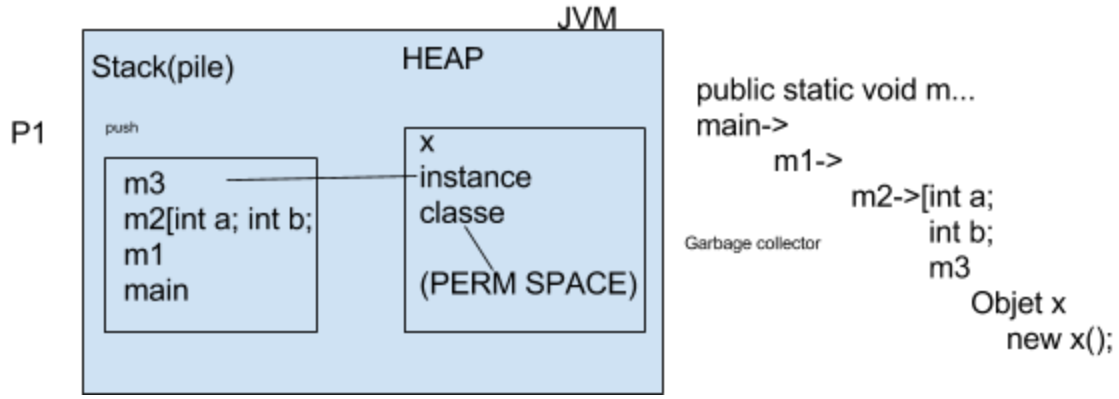
Polymorphisme



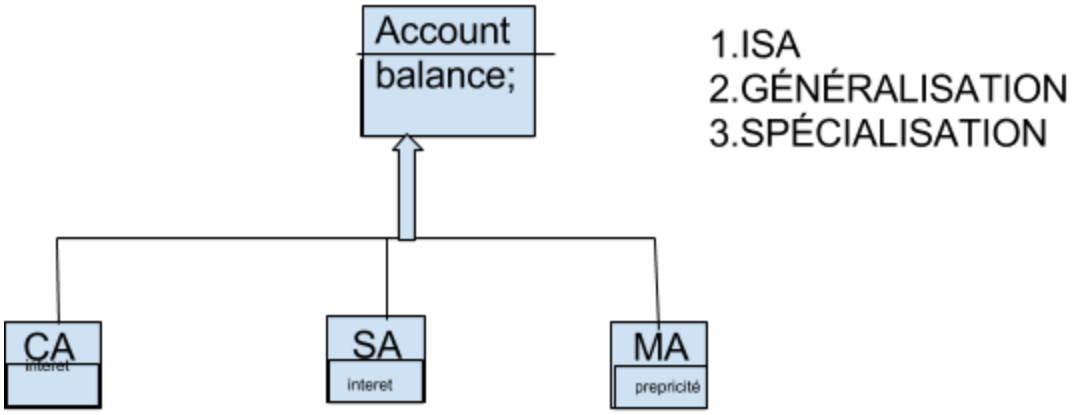
Exemple de cast
Exemple d'UML



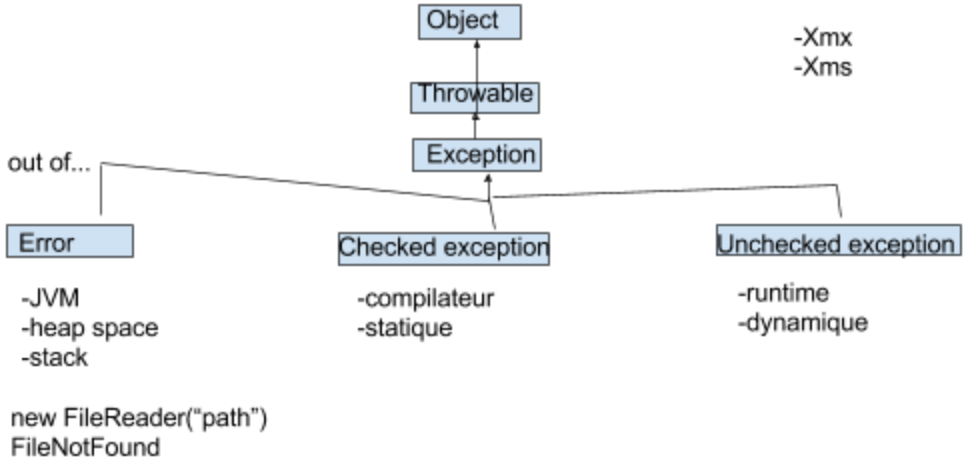
19 septembre 2017



OO: Polymorphisme(méthode qui prends plusieurs forme)
Héritage



EXCEPTIONS:



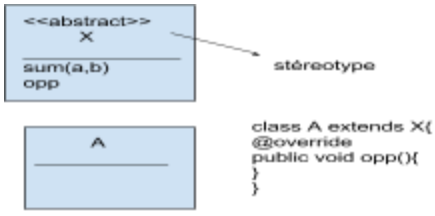
try catch	declarer
try{...; catch(IOException)...; Pas besoin de réécrire l'exception pour les autres	throws IOException; faire d'un try catch pour les autres

Traitement des exceptions:
méthode abstraite:abstraite et concrète
liaison dynamique(si la methode à executer se prend pendant l'exec du prog)

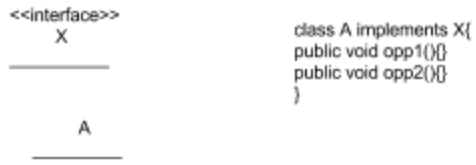
21 septembre 2017
(Java7)
Classe:
public class A extends B implements C{
membres//instance
membres//class
membres//constructor
membres//method
}
MOT CLÉ CONTRAT

Classe Abstraite(contrat partiel):
->méthodes concrètes et méthodes abstraite

```
public abstract class X{
        //méthodes concrète(puisque l'implication est décrite)
        public int sum(int a, int b){
                return a+b;
        }
        //abstraite
        public abstract void app();
        //instance,classe(constructeur possible pour init les valeur, mais new A(); impossible)
        }
```



Interface(contrat complet):aucun cons
-méthodes abstraites
public interface X{
public void opp1();
public void opp2();
}



(Java 8)
interface on doit les voir comme des classe abstraite
INTERFACE:
->default method(concrete)
les classes abstraites son mtn obselete
interface x{
public int opp(); (method abstraite)
default int sum(int a,int b){ (methode concrete)
return int a+b;
}
}

interface X{
default int opp(){
return 1;
}
}

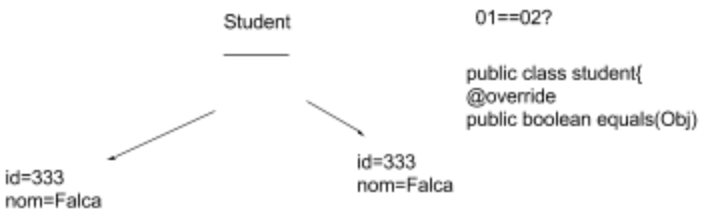
interface Y{
default int opp(){
return 1;
}
}

class A implements X,Y{-> ne compile pas seul
@override
int opp{

<u>MODIFICATEUR D'ACCÈS:</u>
TOP
-public(tout le monde)
VVVseulement la classe avec private
-private*(personne(classe imbriquée)seulemt)
-<u>package private(seulement ceux
du meme package peuvent y avoir acces)</u>

MEMBRE(pour var d'instace/classe et méthode)
protected

>> public/protected/private/___ int A;



THREAD(sous programme)
Client->Server
C1/2/3/4(T1)<->S(thread principale) 3 clients ->min 4 threads