

The background of the slide is a spiral-bound notebook with a light beige, textured paper. The metal spiral binding is visible on the left side. The text is centered on the page.

Analyse et Programmation Orientées Objets / C++

Gestion de packages

Le concept de *package* (1)

➔ Définition

Ensemble fonctionnel cohérent de :

- Classes C++ (fichiers .h et fichiers .cpp)
- Classes C++ (fichiers .obj)
- Classes C++ (fichiers .lib et .dll)

Présence d'une classe abstraite de même nom que le package.

Le concept de *package* (2)

➔ Structure

Tous les constituants d'un package sont contenus dans un même répertoire physique.

Les classes sont représentées par des sous répertoires (structure récursive).

Implémentations opérationnelles

➡ Langage C++

Le concept n'est pas implémenté dans le langage

➡ Langage Java

- *java.lang* (classes *Object*, *System*, *String*, ...)
- *java.applet*
- *java.awt*
- *java.awt.event*
- *java.net*
- - - -

Simulation en APO_C++ (1)

➔ Définition dans le cadre des TD&TP

- Package **_Tests**
- Package **_Maths**
- Package **_Banque**
- Package **_Trafalgar**
- Package **_Archives**
- Package **_Puzzle**

Mise en œuvre à partir de l'unité virtuelle T:

Simulation en APO_C++ (2)

➔ Schéma déclaratif d'une classe

Toute classe C++ doit être déclarée

Déclarations multiples invalides

Identification formelle par `#define _XXXX_`

Mise en oeuvre des directives `#ifndef` et `#endif`

Package **_Tests**

➔ **Exécution des T.U. (cas nominaux et cas d'anomalies)**

- Classe **Tests**

Conforme à la charte des tests unitaires

Comparaisons limitées aux types de base

Extension aisée à toutes les classes des packages cités précédemment

Package **_Banque**

➔ **Entités fonctionnelles « métier »**

- Classe **Titulaire**
- Classe **OperationBancaire**
- Classe **CompteBancaire**
- Classe **CompteCourant**
- Classe **CompteEpargne**
- Classe **AgenceBancaire**
- Classe **Banque**

Package **_Maths** (1)

➔ **Extensions de types du langage C/C++**

- Classe **RxR** Corps \mathbb{C} des complexes
- Classe **Fraction** Corps \mathbb{Q} des rationnels
- Classe **Vecteur** Espace vectoriel $\mathbb{R} \times \mathbb{R}$
- Classe **Matrice** Matrices sur \mathbb{R}
- Classe **MatriceF** Matrices sur \mathbb{Q}

Package `_Maths` (2)

➔ **Gestion de l'espace affine $\mathbb{R} \times \mathbb{R}$, doté des isométries**

- Classe **Point** Isomorphe à **$\mathbb{R} \times \mathbb{R}$**
- Classe **Segment** Portion d'une droite bornée par deux points distincts
- Classe **Triangle** Polygone à trois sommets distincts et non alignés
- Classe **Angle** Définie à partir du produit scalaire du plan euclidien

Package `_Maths` (3)

➔ 8 classes d'origine

- Classe **Droite** Sous espace affine de dimension 1
- Classe **DemiDroite** Sous ensemble d'une droite, défini par une équation vectorielle
- Classe **LigneBrisee** Suite finie de points distincts, avec 2 extrémités distinctes
- Classe **Polygone** Suite finie de points distincts, avec 2 extrémités confondues

Package `_Maths` (4)

➔ Extensions pédagogiques

- Classe **Ensemble** Au sens le plus général !
- Classe **Monome** Ensemble $C \times Z$
- Classe **Polynome** Ensemble de monômes
- Classe **FractionR** Rapport de 2 polynômes

Package _Trafalgar

➔ Simulateur du champ de bataille (**simu**)

- Classe **Navire** Élément d'une flotte
- Classe **Radar** Système de détection
- Classe **Arme** Système de combat
- Classe **Territoire** Environnement de combat
- Classe **Jeu** Interface avec **Protee**

Package **_Archives**

➔ **Persistence des objets sur disque**

- Classe **Object** Abstraction de tout objet
- Classe **File** Abstraction des fichiers
- Classe **Archive** Abstraction du stockage

*Interface fonctionnelle analogue aux M.F.C,
avec tous les codes sources fournis.*

Package **_Puzzle**

➔ **Entités fonctionnelles « métier »**

- Classe **Piece** Couple de vecteurs ($\mathbb{R}^2 \times \mathbb{R}^2$)
- Classe **Puzzle** Ensemble fini de pièces et un polygone de référence