Enseignement de la programmation orientée objets (POO)

Jacques Dugas et Lucien Roy
Département d'informatique
CEGEP de Rimouski

1- Généralités

- Liminaire

- **Objectifs** de projet : proposer et expérimenter des modalités concrètes d'intégration de la POO à notre enseignement
- «**Thèse**» : il faut maîtriser la POO d'ici 3 ans
- **Approche** : globale et fondée sur des concepts de base en faisant des liens avec la programmation conventionnelle

1.1- À propos de la POO... ou une genèse de la POO

- Programmation structurée puis POO : une continuité
- Débuts de la POO : recherches en IA, Alan Kay, Smalltalk
- Antécédents POO : modularité, structuration et hiérarchie
- Objectifs de la POO : **efficacité** par modularité, standardisation et réutilisation
- Pénétration de la POO : variable selon les milieux
- POO : une transition

1.2- Concepts de base et terminologie

- Approche fonctionnelle VS déclarative [Comment VS Quoi]
- Notion d'objet :
  - liste de propriétés : **attributs et méthodes** [Quoi et comment]
  - amalgame de code et de données
- **Encapsulation** ∫ module avec données et procédures
  - assure la modularité et l'indépendance des modules
  - interface d'un objet ∫ sa partie déclarative
  - code d'un objet ∫ sa partie procédurale
- **Classes et héritage** ∫ organisation hiérarchique pour transmettre des propriétés
  - Classe ∫ liste des propriétés communes à un groupe d'objets

(gabarit ou *template* commun à un ensemble d'objets)
- Exemple [page 11]

- **Instances** ∫ un exemplaire d'une classe ∫ un objet

- **Messages** : sélecteurs de message = nom d'une méthode
  - *HandleEvent* et la programmation par événements
  - cheminement d'un message dans une hiérarchie

- Polymorphisme ou l'art des méthodes génériques

- Un modèle : encapsulation, héritage, liens entre classes et instances [page 15]

1.3- Un survol des environnements de type POO

- Diversité des outils, du vocabulaire et des mécanismes d'implantation

- Critères de classification

- Une classification [schéma, page 23]
  - orientés objets «purs»
  - basés objets
  - ateliers
  - langages et librairies de classes

- Une méthode de développement : le prototypage

1.4- À propos de l'analyse orientée objets (AOO)...

- Nécessité d'une méthodologie, d'une approche

- État de la question :
  - à la recherche de LA méthode
  - adaptation des méthodes d'analyse structurée

- La **recherche des objets** ==> encapsulation assurée

- Une méthode émergente : celle de Coad & Yourdon

- Étapes d'analyse de Coad & Yourdon : objets, hiérarchie, sujets, attributs et méthodes (Notation et exemple de Coad & Yourdon) [page 97]

- Des questions : agencement hiérarchique de classes, etc.

- Évaluation : ressemblances et différences avec une approche conventionnelle

- Outils CASE et AOO

1.5- La POO et l'industrie informatique

- État de la POO :
  - pénétration variable selon les milieux
  - méthodologie à préciser, à raffermir et à généraliser
  - prototypage comme technique de réalisation

- Tendances perceptibles
  - Grands constructeurs imposent **standards** avec leurs produits

- outils et méthodes de programmation
- applications (interfaces)

- Accent sur la réutilisation de code
- Généralisation de librairies de classes
- Intégration d'objets dans un environnement homogène [page 109]

- Transition vers la POO
  - Généralisation des outils et méthodes POO d'ici peu
  - À partir de la programmation structurée et des L4G
  - Encapsulation : une autre technique de programmation

1.6- Quelques références [page 121]

2- Notre expérience POO à Rimouski

2.1- Notre projet de recherche (A.T.P.E.)

Nous avons bénéficié d'une subvention dans le cadre d'un projet A.T.P.E. pour libérer deux enseignants de 0,4 E.T.C. chacun en 1991-92. Notre objectif visait principalement trois volets :

- dégager les concepts de base de la POO;
- explorer quelques outils orientés objets;
- proposer un scénario d'intégration de la POO dans le DEC en informatique.

Le document *Introduction à la programmation orientée objets (POO)* constitue notre rapport de recherche.

2.2- Capsule de 15 heures [p. 127-128]

À la fin de notre recherche, à la demande des élèves de troisième année, nous avons aménagé 15 heures à l'intérieur d'un cours pour présenter la POO. La capsule se divise en trois parties de 5 heures chacune :

- Généralités, concepts de base, classification des outils, approche orientée objets;
- HyperCard sur Macintosh : un outil basé objets où il n'est pas possible de définir une nouvelle classe d'objets;
- La POO en Turbo Pascal et la librairie Turbo Vision.

Beaucoup de matière pour très peu de temps.

2.3- Cours de programmation orientée objets [p. 131-146]

Nous nous dotons d'un véritable cours de POO au cinquième trimestre du DEC depuis l'année 92-93. Il nous semble qu'il faut consacrer au moins 45 heures sur le sujet, soit dans le cadre d'un des cours 8xx ou dans le cours 591 (Projet de fin d'étude 1).

À Rimouski, nous consacrons le cours 591 à la POO (60 heures). Le cours se divise en quatre parties :

- les concepts de base, le vocabulaire de la POO et l'approche orientée objets (20 %);
- la POO avec le langage C++ (25 %);
- utilisation d'une librairie orientée objets permettant de gérer l'interface utilisateur pour réaliser des applications d'allure professionnelle (25 %);
- utilisation d'un prototypeur générant du code orienté objets en C++ (30 %).

L'omniprésence de la POO dans leurs outils de tous les jours (Turbo Pascal, C++, etc) constitue en soie une motivation pour les élèves. Même si l'utilité de la POO n'apparaît pas tout de suite comme évidente auprès des élèves, son importance n'est pas mise en doute. Il pourrait être tentant d'aborder la POO en se contentant d'en voir les modalités d'implantation dans un langage précis comme le C++. Nous pensons toutefois que la compréhension de la POO doit passer par l'étude plus formelle des notions de base.

Peut-on encore ignorer le développement orienté objets dans un environnement GUI tel que Windows ? À Rimouski, nous n'avons pas encore fait ce pas mais nous l'envisageons.

2.4- Utilisation d'objets dans les cours 201 et 301

Dans le dernier exercice pratique du cours 201, les élèves utilisent une petite librairie orientée objets. Le langage utilisé étant dépourvu du type *date*, les élèves utilisent une classe *date* définie par le professeur. L'élève s'initie ainsi à la syntaxe d'appel des méthodes d'un objet et entrevoit l'avantage de la POO du point de vue de l'indépendance des modules.

Dans le cours *Structures de données*, une librairie orientée objets est mise à la disposition des élèves pour permettre la gestion de fichiers indexés. Les notions de POO ne sont pas requises. Seule la syntaxe particulière d'appel des méthodes doit être connue. Un tout petit pas vers la POO qui facilite la transition vers un véritable cours de POO et établit un lien supplémentaire entre un cours de deuxième année et un cours de troisième année.

2.5- Un bilan sommaire et des questions

L'enseignement de la POO nous apparaît répondre à un besoin de l'élève, à une nécessité pour demeurer compétitif et à une tendance des années 90. L'expérience montre que ce sont des notions abordables au niveau du DEC. Cependant, nous avons des interrogations :

- Qu'en est-il de l'approche orientée objets ?
- Qu'advient-il de la méthode d'analyse enseignée dans le cours 401 ?
- Doit-on faire abstraction des objets en modélisation des données ?
- Quelle est la pénétration des SGBD orientés objets sur le marché ?
- Doit-on enseigner la POO sous Windows ?

3- Intégration de la POO à l'enseignement

3.1- Nos hypothèses de travail [p. 111]

- La POO : une autre technique de programmation à maîtriser
- La POO : un standard dans un proche avenir
- La POO : un thème à intégrer à notre enseignement dès que possible

3.2- Motivations [p. 111-112]

- Valeur des idées et des réalisations POO
- POO : une autre technique de programmation
- Tendances du marché
- Que nos élèves soient au fait des technologies récentes en informatique

3.3- Scénarios : préalables, moyens, séquences, cours [p. 111 à 116]

- Préalables
    - Algorithmique
    - Langage de programmation
    - Méthode d'analyse
- Outils
    - Matériel didactique adapté
    - Références
    - Banques de programmes exemple
    - Logiciels : basés objets, librairies de classes et prototypeurs
- Stratégie : favoriser la réutilisation de code déjà écrit (classes, exemples, etc.)
- Pour les finissants «néophytes» : un cours vers la fin du DEC
  [Voir plans de capsule et de cours]
- Pour les «nouveaux» : dès le deuxième trimestre du DEC
  [Voir exercices POO]
    - Programmation de l'encapsulation (Exercice DatePOO)
    - Utilisation d'une librairie de classes (Exercice EditLogo)
    - Programmation de l'héritage et du polymorphisme
    - Analyse, conception et programmation selon approche objets
    - Initiation aux langages avec leur extension objets
- Les cours visés et leur adaptation nécessaire
    - Séquence *Programmation* (201, 301)
    - Séquence *Analyse et développement de systèmes*

(401, 501 et 601)

- Les cours sur des langages de programmation
- Les champs d'application (choix institutionnels)

3.4- Perspectives

- Quel langage de programmation pour s'initier à la POO ?

- Développement d'applications ponctuelles dans un environnement graphique comme Windows

- Analyse orientée objets : quand ? Comment ? Avec quoi ?

- Développement de systèmes complets selon une approche objets à toutes les étapes : analyse, conception, prototypage et mise au point

- Alternative : approche «globale» ou approche «analytique» ?

3.5- Et alors...

- La POO arrive ! Il faut prendre le train !

- Intégrer la POO à notre enseignement d'ici 3 ans.