La notion d'exception en programmation

Cynthia Pitou

La notion d'exception en programmation

Cynthia PITOU, DOCTORANTE
LIM, UNIVERSITÉ DE LA RÉUNION

La notion d'exception se retrouve en programmation à travers les systèmes de gestion d'exceptions. Ces systèmes de gestion d'exceptions permettent de gérer les cas exceptionnels qui peuvent survenir lors de l'exécution d'un programme informatique. La logique d'utilisation des systèmes de gestion d'exceptions fait apparaître des liens précis entre les règles et leurs exceptions.

En programmation, les exceptions sont des cas limites d'exécution d'un programme. Elles provoquent en général l'interruption du déroulement normal du programme. Un cas limite générique et commun à tous les langages de programmation est par exemple la division par zéro. Le premier système de gestion d'exceptions est apparu dans les années 1970 avec le langage PL/I[1]. La mise en place d'un système de gestion d'exceptions présente plusieurs intérêts. D'une part, elle met à disposition une stratégie de détection et de réparation des erreurs relevant d'une exception. D'autre part, le langage bénéficie ainsi d'une syntaxe particulière pour distinguer l'exécution normale du traitement des erreurs. De plus, grâce au système de gestion d'exceptions, le programmeur a un contrôle sur le traitement des situations exceptionnelles.

Considérons le cas suivant. Un principe mathématique bien connu est que, par convention, la division par zéro n'a pas de sens. On peut donc, sans trop s'avancer, dire que zéro est une exception vis-à-vis d'un principe ou d'une règle qui autorise la division entre deux nombres (entiers naturels) quels qu'ils soient.

Imaginons un programme informatique dont le but est de calculer le résultat de la division entre deux nombres. En java[2], langage de programmation orienté objet, ce programme s'écrit :

```
public class CalculatriceSpecialDivision {
        public static void divise(int n1, int n2){
                try{

System.out.println(n1+"/"+n2+"="+n1/n2);
                }
                catch (ArithmeticException e){
```

1 Le PL/I ou PL/1 (*Programming Language number 1*) est un langage de programmation qui a été développé par IBM au début des années 1970.

2 Langage de programmation objet développé par Sun Microsystems, et apparu en 1995.

```
                                        //la division par zéro n'est pas possible
                                        System.out.println("Impossible de diviser
"+n1+" par "+n2);
                                }
                        }
                        public static void main (String[] args){
                                divise(10,2);
                                divise(10,0);}
        }
```

Ce qui nous intéresse se trouve dans le bloc défini par la méthode intitulée *divise*. Dans cette méthode, la division en elle-même est faite dans un bloc *try catch*. Ce bloc a un intérêt et une utilisation particulière en programmation java. En effet, il permet de signaler et de traiter toutes sortes d'exceptions. En langage courant, on pourrait traduire ce bloc par « essaie de calculer n1/n2 et d'en afficher le résultat à l'écran, et si jamais ce n'est pas possible, affiche "Impossible de diviser n1 par n2" ». Ici, l'impossibilité de faire la division est marquée dans l'instruction *catch* par le fait d'intercepter une exception de type *ArithmeticException*[1]. Typiquement, diviser par zéro lèvera une exception de ce type.

Revenons à présent sur le lien qu'il pourrait y avoir entre la règle et l'exception. Dans notre cas d'étude, la règle se résume à pouvoir mathématiquement diviser deux nombres passés en paramètre quels qu'ils soient. Tant que l'exécution de notre programme ne lève pas d'exception, on peut dire que la règle est confirmée. Dès lors que l'on demandera l'exécution du programme avec comme diviseur le chiffre zéro, une exception sera levée. On constate d'ores et déjà que l'exception sert ici à notifier que l'on n'a pas pu exécuter l'instruction représentant la règle mathématique.

Ce cas d'étude peut être généralisé à tous les langages de programmation qui disposent d'un système de gestion d'exceptions. Le lien qui est établi entre la règle (la norme, le déroulement correct d'un programme informatique) et l'exception, est le lien de cause à effet suivant : l'impossibilité d'appliquer la règle lève une exception identifiée pour cette règle. En effet, si une exception apparaît, alors la règle ne peut être totalement confirmée. Réciproquement, tant que la règle est confirmée alors aucune exception n'est levée. Donc, la règle est confirmée si et seulement si l'exécution du programme ne lève pas d'exception.

Par ailleurs, une règle en mathématiques est définie comme une conjecture que l'on suppose vraie en l'absence de contre-exemple, mais qui n'est pas encore prouvée. De cette manière, une règle exclut toute exception.

---

[1] Classe d'objet java. Une instance de cette classe est lancée lorsqu'une condition arithmétique n'est pas respectée.

En conclusion, les systèmes de gestion d'exceptions en programmation informatique, n'établissant aucune implication directe entre la mise en évidence d'exceptions et la confirmation des règles dont découlent ces exceptions, révèlent une contradiction flagrante d'un point de vue mathématique dans cette expression du langage courant : « C'est l'exception qui confirme la règle ».

Autrement dit, une exception ne peut pas confirmer la règle qui l'a mise en exergue, mais elle est bien là pour infirmer la véracité absolue de celle-ci. Une assertion correcte d'un point de vue mathématique, et modélisable par un programme informatique, est que : « C'est l'exception qui infirme la conjecture ».

BIBLIOGRAPHIE

DELLANOY, Claude, *Programmer en Java,* France : Eyrolles, coll. « Noire », 2006.

FLANAGAN, David, *Java en concentré*, France : O'Reilly, coll. « En concentré », 2006.

BOUGEAULT, Jérôme, *Java – La maîtrise*, France : Eyrolles, Tsoft, coll. « Les guides de formation Tsoft », 2008.

Commentcamarche.net, « Java – Les exceptions », URL <www.commentcamarche.net/contents/java/javaexc.php3>, consulté le 15/02/2012.

Oracle.com, « Class ArithmeticException », URL <http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/ArithmeticException.html>, consulté le 14/02/2012.

Wikipedia, « Java (langage) », URL <http://fr.wikipedia.org/wiki/Java_%28langage%29>, consulté le 14/02/2012.

Wikipedia, « Système de gestion d'exceptions », URL <http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_d%27exceptions#Particularit.C3.A9s>, consulté le 10/02/2012.

Wikipedia, « Conjecture », URL <http://fr.wikipedia.org/wiki/Conjecture#Exemples_de_conjectures_c.C3.A9l.C3.A8bres_actuelles>, consulté le 16/02/2012.

Wikipedia, « PL/I », URL <http://fr.wikipedia.org/wiki/PL/I>, consulté le 07/03/2012.