

IUT de Paris Descartes – Base de la Programmation Objet

Travaux Dirigés – Sujet n°10

Objectifs

- Restructuration de code existant en vue de généralisation.
- Héritage et sous-typage.

Un (des) jeu(x) de société

Le point de départ de notre travail est le code source fourni en annexe. La classe `JeuDeMarienbad` permet de jouer à une version du jeu de Nim. Le principe du jeu est simple. Deux joueurs s'affrontent en jouant à tour de rôle. Jouer consiste à choisir une ligne et à y retirer au moins une allumette. Initialement, il y a quatre lignes contenant respectivement 1, 3, 5 et 7 allumettes. Le joueur retirant la dernière allumette a perdu.

L'objectif de la séance de travaux dirigés est de retravailler ce programme de façon à pouvoir réemployer la structure commune à de nombreux jeux de société

1. Lire le programme donné en annexe et déterminer ce qui est propre à cette version de jeu de Nim et ce qui est commun à de nombreux jeu de société.
2. Proposer une solution permettant de spécialiser *a posteriori* ce qui est particulier à un jeu (tel que le jeu de Marienbad par exemple).
3. Restructurer le programme en conséquence. Vous écrirez le code complet de la classe abstraite puis donnerez les grandes lignes de la classe concrétisant le jeu de Marienbad.
4. Une autre solution consiste à paramétrer la méthode `jouerUnePartie` par un objet représentant le jeu particulier. Déterminer la nouvelle structure du programme et le modifier en conséquence.

Annexe

```
public class JeuDeMarienbad {
    private static final int NB_JOUEURS = 2;
    private static final int[] ALLUMETTES = { 1, 3, 5, 7 };
    private int[] allumettes;

    public JeuDeMarienbad() {
        allumettes = ALLUMETTES.clone();
    }

    public void jouerUnePartie() {
        int j = 0;
        allumettes = ALLUMETTES.clone();
    }
}
```

```
while (true) {
    faireJouer(j);
    if (partieFinie())
        break;
    else
        j = (j + 1) % NB_JOUEURS;
}
System.out.println("le_joueur_n°" + ((j + 1) % NB_JOUEURS)
    + "_a_gagné_la_partie");
}

public void faireJouer(int j) {
    assert (!partieFinie());
    Scanner sc = new Scanner(System.in);
    int ligne, nombre;
    do {
        System.out.println(this);
        System.out.println("Joueur_n°" + j);
        System.out
            .print("n°_de_la_ligne_où_prendre_des_allumettes:_");
        ligne = sc.nextInt();
        System.out
            .print("nombre_d'allumettes_à_retirer_de_cette_ligne:_");
        nombre = sc.nextInt();
    } while (ligne < 0 || ligne >= allumettes.length || nombre <= 0
        || nombre > allumettes[ligne]);
    allumettes[ligne] -= nombre;
}

public boolean partieFinie() {
    for (int n : allumettes)
        if (n != 0)
            return false;
    return true;
}

@Override
public String toString() {
    String s = "";
    for (int i = 0; i < allumettes.length; ++i)
        s += "ligne_n°" + i + "_:_ " + allumettes[i] + "_allumette(s)\n";
    return s;
}

public static void main(String[] args) {
    JeuDeMarienbad j = new JeuDeMarienbad();
    j.jouerUnePartie();
}
}
```