

Éléments de base

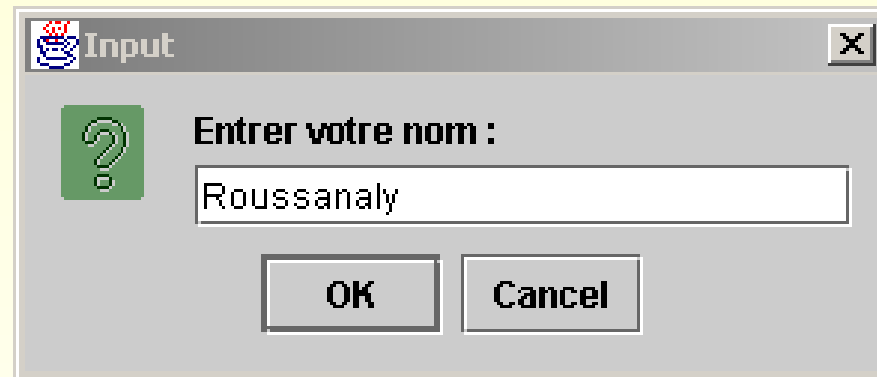
Programmation objets
et Java

(B. Sonntag)

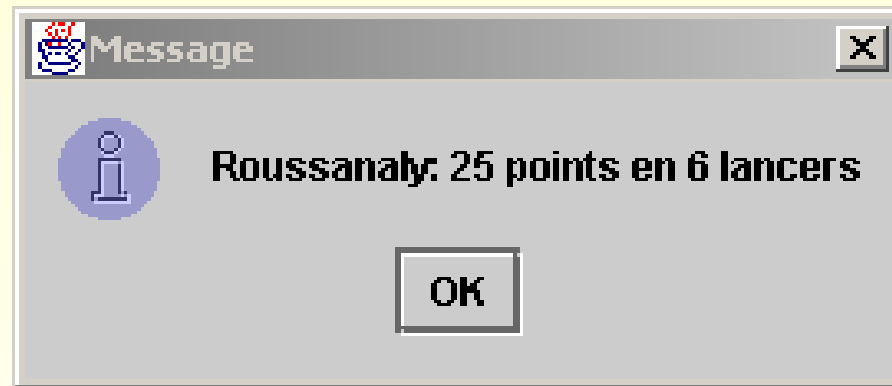
Problème

- ⇒ Simuler une jeu de dé
- ⇒ On lance plusieurs fois un dé et, à chaque lancer, on cumule le nombre de points marqués sur le dé. Le but est d'atteindre 21 points avec un minimum de lancer.

Exemple



Exemple



Classe

```
public class Score {
```

```
...
```

```
} //classe
```

Propriétés

```
public class Score {  
    //propriétés  
    //-----  
    String nom;  
    int points;  
    int max;  
    ...  
} //classe
```

Constructeurs

```
//constructeurs
```

```
//-----
```

```
public Score(String nom,int points,int max){  
    this.nom = nom;  
    this.points = points;  
    this.max = max;  
}
```

Autres constructeurs

```
public Score() {  
    this("inconnu", 0, 100);  
}  
  
public Score(String nom, int points) {  
    this(nom, points, 100);  
}
```


Accès direct aux propriétés

```
//accès aux propriétés
```

```
Score sc = new Score();
```

```
sc.nom = "Roussanaly" ;
```

```
sc.points = 6;
```

Accès direct aux propriétés

```
//accès aux propriétés
```

```
Score sc = new Score();
```

```
sc.nom = "Roussanaty" ;
```

```
sc.points = 6;
```

**Un objet doit rester maître
de la valeur de ces propriétés !**

Propriétés privées

```
public class Score {  
    //propriétés  
    //-----  
    private String nom;  
    private int points;  
    private int max;  
    ...  
} //classe
```

Exposer les propriétés

```
//setters et getters
//-----
/**
 * getMax
 * @return int
 */
public int getMax() {
    return max;
}
```

Exposer les propriétés

```
/**  
 * setMax  
 * @param int  
 */  
public void setMax(int i) {  
    max = i;  
}
```

Exposer les autres propriétés

```
/**
 * getNom
 */
public String getNom() {
    return nom;
}

/**
 * getPoints
 */
public int getPoints() {
    return points;
}
```

Exposer les autres propriétés

```
/**
 * setNom
 */
public void setNom(String string) {
    nom = string;
}

/**
 * setPoints
 */
public void setPoints(int i) {
    points = i;
}
```

Accès aux propriétés proprement

```
//accès aux propriétés
```

```
Score sc = new Score();
```

```
sc.setNom("Roussanaly");
```

```
sc.setPoints(6);
```

```
System.out.println(sc.getNom());
```


Méthodes

```
/**  
 * ajouter un point  
 */  
public void ajouter(){  
    points++;  
    if (points > max) points = max;  
}
```

Autre méthode (=Surcharge)

```
/**
 * ajouter n points
 * @param int : nombre de points à ajouter
 */
public void ajouter(int n){
    points += n;
    if (points > max) points = max;
}
```

Classe dérivée

```
public class ScoreDé extends Score{  
  
    //propriétés  
    //-----  
    private int lancers;  
  
    ...  
}
```

Constructeurs

```
public ScoreDé(String nom,  
                int points,  
                int lancers) {  
    super(nom, points);  
    this.lancers = lancers;  
}
```

Autres constructeurs

```
public ScoreDé() {  
    this("inconnu", 0, 0);  
}  
public ScoreDé(String nom) {  
    this(nom, 0, 0);  
}
```

Surcharge de méthode

```
/**  
 * ajouter n points  
 */  
public void ajouter(int n){  
    super.ajouter(n);  
    if (getPoints() > getMax())  
        setPoints(getMax());  
    lancers++;  
}
```

Méthode toString()

```
/**
 * toString
 */
public String toString(){
    String res = getNom() + ": "
                +getPoints()+" points en "
                +getLancers()+" lancers";
    return res;
}
```

Roussanaly: 22 points en 4 lancers

Application

```
import javax.swing.JOptionPane;

public class UnJoueur {
    public static void main(String[] args) {
        ...
    }
}
```


Application

```
String nom = JOptionPane.showInputDialog(  
    null,  
    "Entrer votre nom :");  
ScoreDé sc = new ScoreDé(nom);  
while ( sc.getPoints()<21){  
    Double d = new Double(6*Math.random()+1);  
    int tirage = d.intValue();  
    sc.ajouter(tirage);  
}  
JOptionPane.showMessageDialog(  
    null,  
    sc.toString());
```