



Java

SOURCES

- ▶ ~~Le site du zéro~~ Openclassrooms
- ▶ Documentation Oracle
- ▶ StackOverflow
- ▶ Google

SOMMAIRE

- ▶ Présentation du langage (Rapidement, promis)
- ▶ La programmation orientée objet
- ▶ Gérer un projet de A à Z
 - ▶ Modélisation
 - ▶ Implémentation (Organisation des fichiers, Codage)
 - ▶ Déploiement
- ▶ Ouverture, deux ou trois idées à creuser

PRÉSENTATION DU LANGAGE

L'HISTOIRE DE JAVA

- ▶ 1991 : Création
- ▶ Green Team
- ▶ 1995 : Lancement Officiel
- ▶ 2010 : Oracle achète Sun Microsystem
- ▶ Aujourd'hui : Java SE 8

The Oracle logo, consisting of the word "ORACLE" in a white, bold, sans-serif font, with a registered trademark symbol (®) to the upper right, all set against a solid red rectangular background.

LE JAVA EN LUI MÊME

- ▶ Langage le plus utilisé au monde (selon tiobe.com, mai 2015)
- ▶ Haut niveau
- ▶ Compilé puis exécuté
- ▶ Portable
- ▶ Orienté objet

LA PROGRAMMATION ORIENTÉE OBJET

WHAT ?

- ▶ Une manière de voir la programmation
- ▶ C++, Java, Python, Php, Swift, C#, MATLAB ...
- ▶ Ensemble d'objet qui interagissent

UNE CLASSE

- ▶ Un type de variable défini par le développeur
- ▶ Possède
 - ▶ des attributs (Informations)
 - ▶ des méthodes (Actions)

UNE CLASSE – EXEMPLE

► Personne :

- Nom ➡ Parler
- Age ➡ Manger
- Taille ➡ Ecrire
- Adresse ➡ Marcher

UN OBJET

► Personne Bob :

- Nom : Bob ➡ Parler
- Age : 42 ➡ Manger
- Taille : 1,74 ➡ Ecrire
- Adresse : Paris ➡ Marcher

LA CLASS MAIN

LA CLASSE MAIN


- ▶ Exécutée en première

```
Main.java x
1 public class Main {
2
3     public static void main(String[] args) {
4         System.out.println("Hello World!");
5     }
6 }
7
```

Hello World!

Process finished with exit code 0

INITIALISER UNE VARIABLE

```
2  public static void main(String[] args) {  
3  
4     int age;  
5     age = 1;  
6  
7     System.out.println(age);  
8  
9 }
```

1

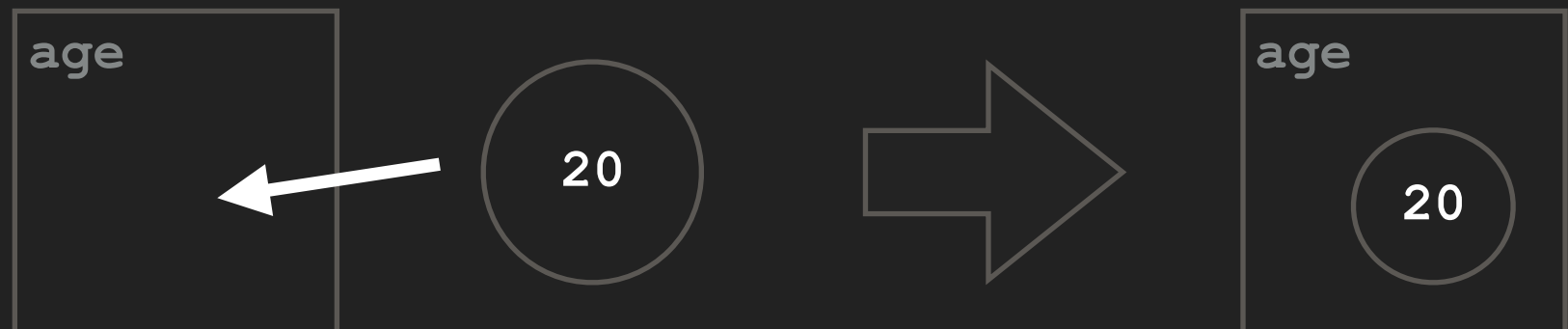
Process finished with exit code 0

INITIALISER UNE VARIABLE

► `int age;`



► `age = 20;`




INITIALISER UNE VARIABLE

```
2 public static void main(String[] args) {  
3  
4     int age = 1;  
5  
6     System.out.println(age);  
7  
8 }
```

1

Process finished with exit code 0

LES OPÉRATIONS SUR LES VARIABLES

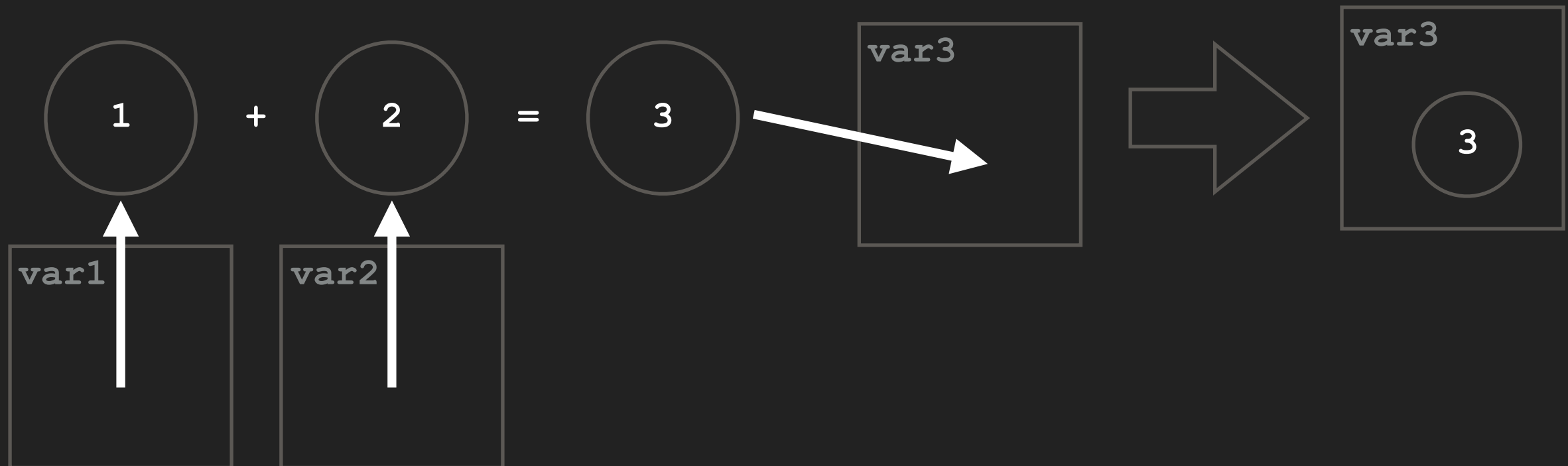
```
2  public static void main(String[] args) {  
3  
4     // 1 + 2 = 3  
5     int var1 = 1;  
6     int var2 = 2;  
7  
8     int var3 = var1 + var2;  
9  
10    System.out.println(var3);  
11  
12 }
```

3

Process finished with exit code 0

LES OPÉRATIONS SUR LES VARIABLES

► `int var3 = var1 + var2;`



LES OPÉRATIONS SUR LES VARIABLES

```
int var3 = var1 + var2;  
var3 = var1 - var2;  
var3 = var1 * var2;  
var3 = var1 / var2;  
var3 = var1 % var2;  
  
var3++; var3--;
```

LES DIFFÉRENTS TYPES DE VARIABLE

```
byte byteVar = 1;           // -128      ...      127
short shortVar = 1;         // -32,768   ...      32767
int intVar2 = 1;            // -2^31    ...      2^31 - 1
long longVar = 1;           // -2^63    ...      2^63 - 1
```

```
float floatVar = 1.0f;
double doubleVar = 1.0;
```

```
boolean boolVar = true; // true      ...      false
char charVar = 'a';
```

LES CONDITIONS

```
int var1 = 1;
int var2 = 2;

if (var1 == var2) {
    System.out.println("Yep");
} else {
    System.out.println("Hm, nope .. ");
}
```

LES CONDITIONS

```
if (var1 == var2) {}  
if (var1 != var2) {}  
if (var1 < var2) {}  
if (var1 > var2) {}  
if (var1 <= var2) {}  
if (var1 >= var2) {}
```

LES CONDITIONS

```
if (var1 < var2 && var3 == var4) {}
```

```
if (var1 != var2 || var3 >= var4) {}
```

LES CONDITIONS

```
if (var1 < var2 && var3 == var4) {}
```

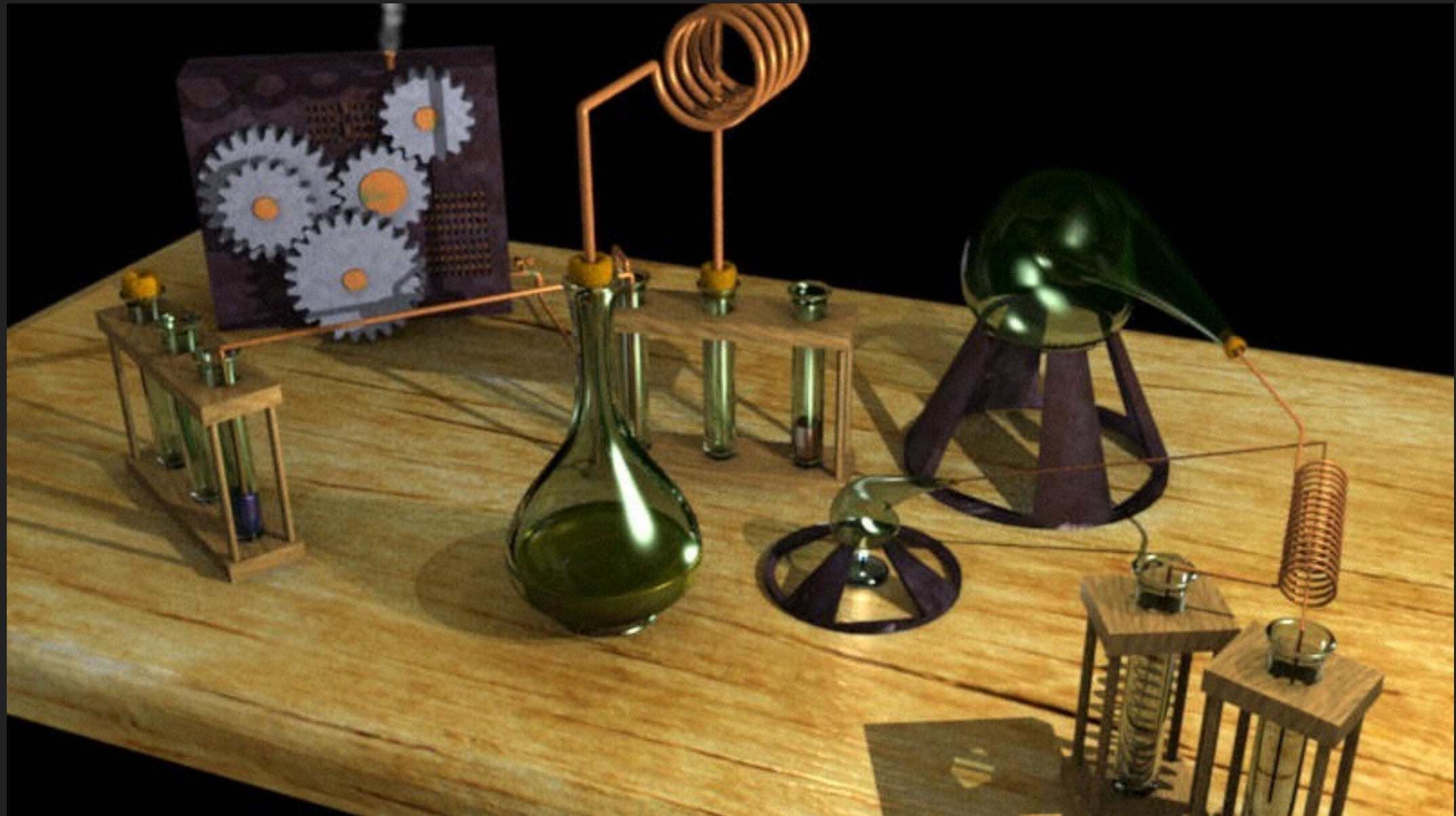
```
if (var1 != var2 || var3 >= var4) {}
```


LES CONDITIONS

```
if (var1 < var2 && var3 == var4) {}
```

```
if (var1 != var2 || var3 >= var4) {}
```

LES FONCTIONS



LES FONCTIONS



LES FONCTIONS

```
public static void main(String[] args) {  
    int firstNumber = 2;  
    int numberAdded = addTwo(firstNumber);  
    System.out.println(numberAdded);  
}  
  
public static int addTwo(int number) {  
    int numberAdded = number + 2;  
    return numberAdded;  
}
```

LES FONCTIONS

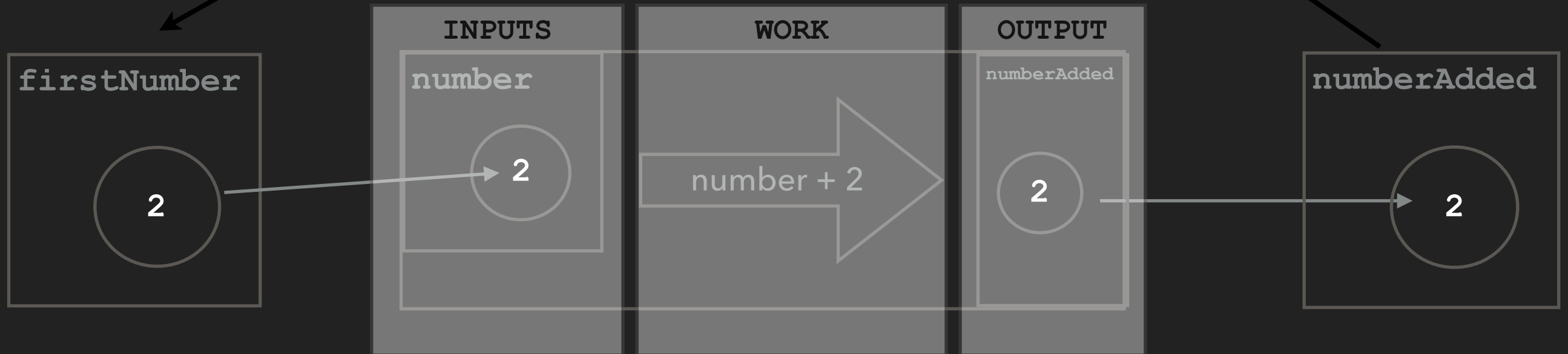
```
public static void main(String[] args) {  
    int firstNumber = 2;  
    int numberAdded = addTwo(firstNumber);  
    System.out.println(numberAdded);  
}
```

```
public static int addTwo(int number) {  
    int numberAdded = number + 2;  
    return numberAdded;  
}
```

LES FONCTIONS

```
public static void main(String[] args) {
    int firstNumber = 2;
    int numberAdded = addTwo(firstNumber);
    System.out.println(numberAdded);
}

public static int addTwo(int number) {
    int numberAdded = number + 2;
    return numberAdded;
}
```



LES POLYNÔMES DU SECOND DEGRÉS

- ▶ $a * x^2 + b * x + c = 0$
- ▶ Calculer les racines d'un polynôme du second degré
- ▶ Inputs : A, B et C
- ▶ Calculer delta, l'afficher
- ▶ Si Delta > 0
 - ▶ Calculer X1 et X2, les afficher
- ▶ Sinon si Delta == 0
 - ▶ Calculer X1, l'afficher
- ▶ Sinon : Répondre qu'il n'y a pas de solution réelle possible

LA GESTION D'UN PROJET

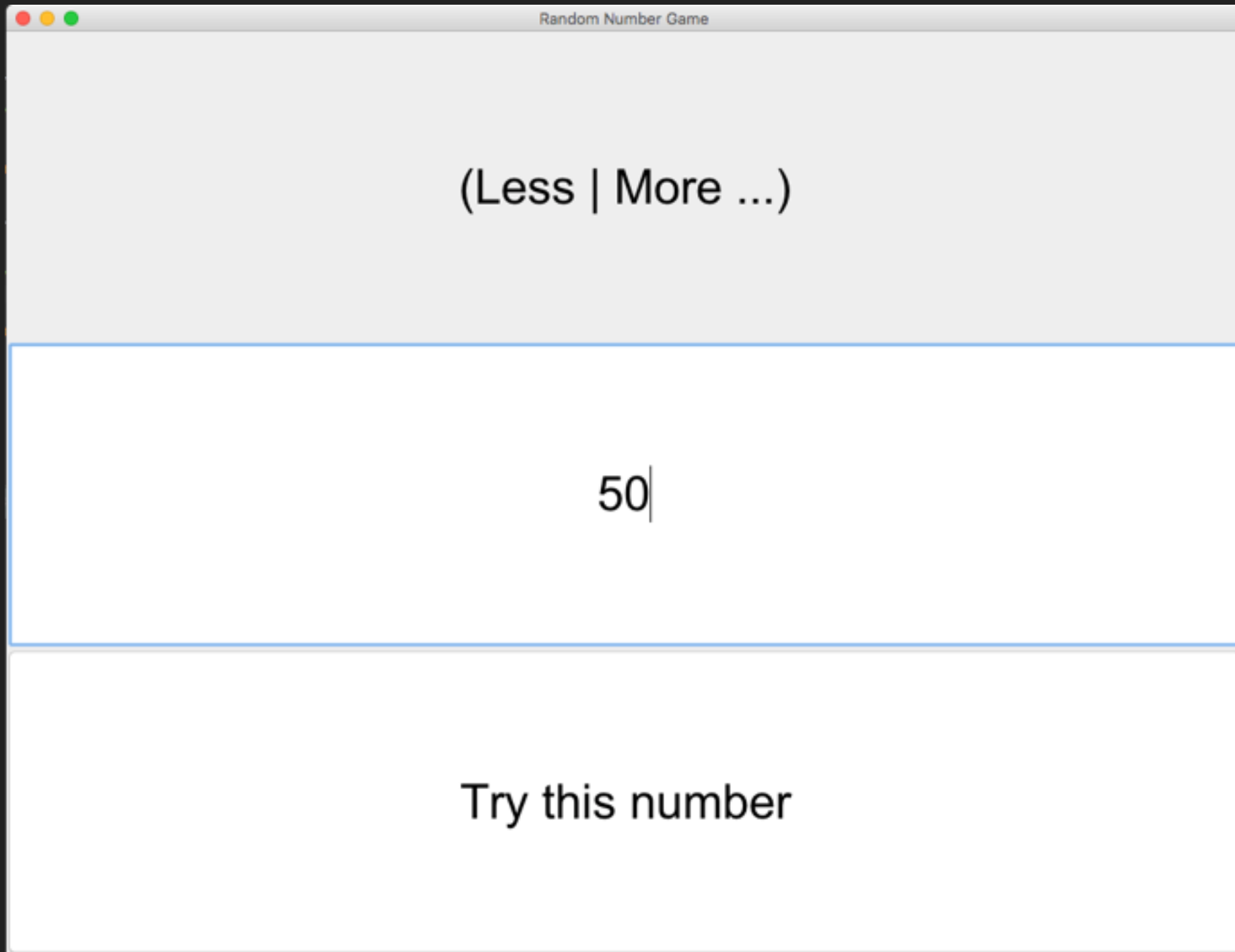
MODÉLISATION

- ▶ 1 - Qu'est ce que mon programme dois faire ?
- ▶ 2 - Comment le faire ?

MODÉLISATION

- ▶ 1 - Qu'est ce que mon programme dois faire ?
- ▶ 2 - Comment le faire ?

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

A screenshot of a Java Swing window titled "Random Number Game". The window has a light gray title bar with standard macOS window controls (red, yellow, green buttons). The main content area is white and divided into three horizontal sections. The top section is a light gray bar containing the text "(Less | More ...)". The middle section is a white text input field containing the number "50" with a vertical cursor to its right. The bottom section is a white area containing the text "Try this number".

Random Number Game

(Less | More ...)

50|

Try this number

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

SECRET NUMBER

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

SECRET NUMBER

INTERFACE GRAPHIQUE

LOGIQUE DU JEU

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

SECRET NUMBER

INTERFACE GRAPHIQUE

LOGIQUE DU JEU

PLATEAU DE JEU

- Nombre à trouver
- Se lancer
- Recevoir un nombre du joueur et lui donner l'info
- S'arrêter si le joueur gagne

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

SECRET NUMBER

INTERFACE GRAPHIQUE

LOGIQUE DU JEU

PLATEAU DE JEU

- Nombre à trouver
- Se lancer
- Recevoir un nombre du joueur et lui donner l'info
- S'arrêter si le joueur gagne

JOUEUR

- Nom
- Récupérer le nombre de l'utilisateur

QU'EST CE QUE MON PROGRAMME DOIT FAIRE ?

SECRET NUMBER

INTERFACE GRAPHIQUE

- Afficher les informations du jeu
- Permettre à l'utilisateur de choisir un nombre

LOGIQUE DU JEU

PLATEAU DE JEU

- Nombre à trouver
- Se lancer
- Recevoir un nombre du joueur et lui donner l'info
- S'arrêter si le joueur gagne

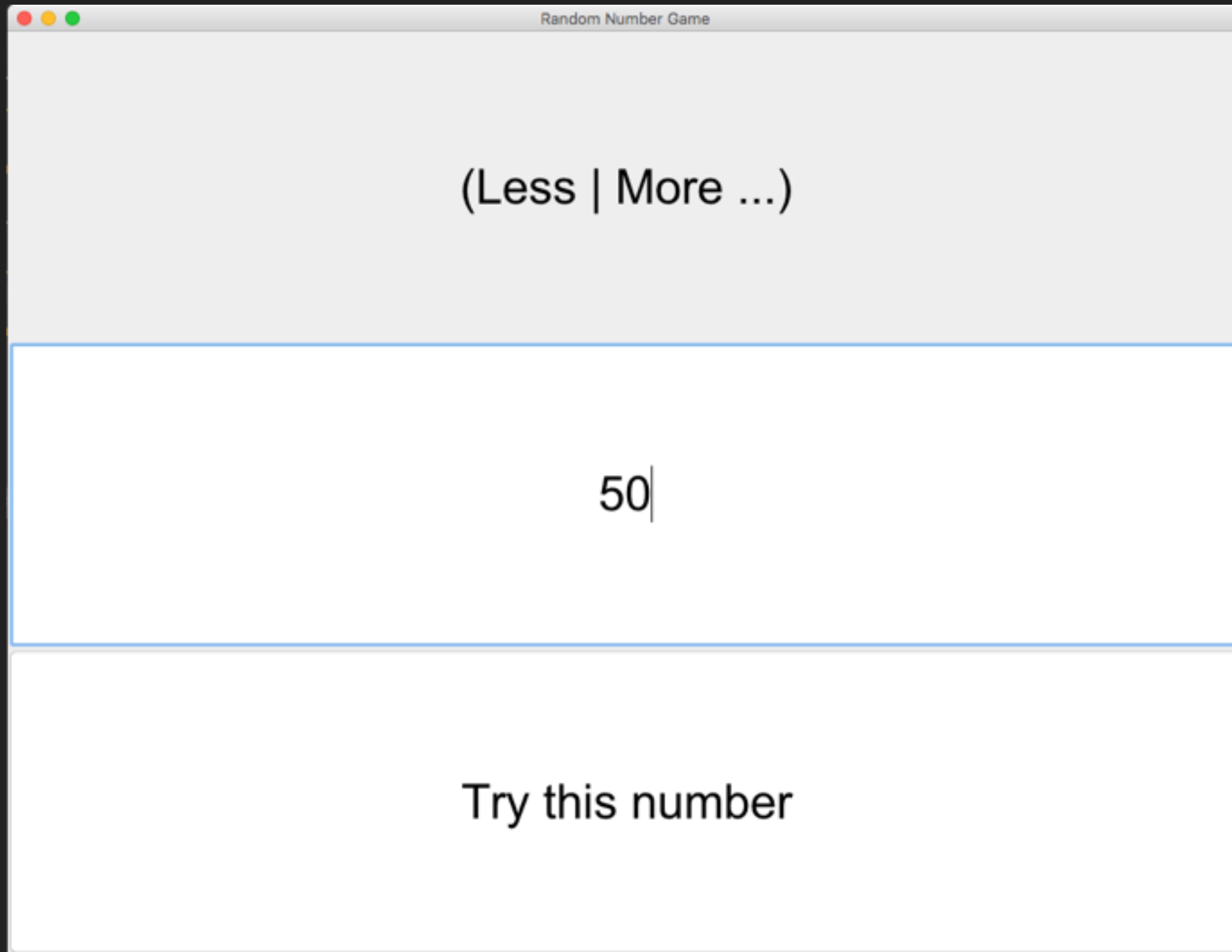
JOUEUR

- Nom
- Récupérer le nombre de l'utilisateur

PENSER LE PROJET

- ▶ 1 - Qu'est ce que mon programme dois faire ?
- ▶ 2 - Comment le faire ?

LET'S GO !

A screenshot of a Java Swing window titled "Random Number Game". The window has a light gray title bar with standard macOS window controls (red, yellow, green buttons). The main content area is white and divided into three horizontal sections. The top section is a light gray header containing the text "(Less | More ...)". The middle section is a white text input field with a blue border, containing the number "50" and a vertical cursor. The bottom section is a white area containing the text "Try this number".

Random Number Game

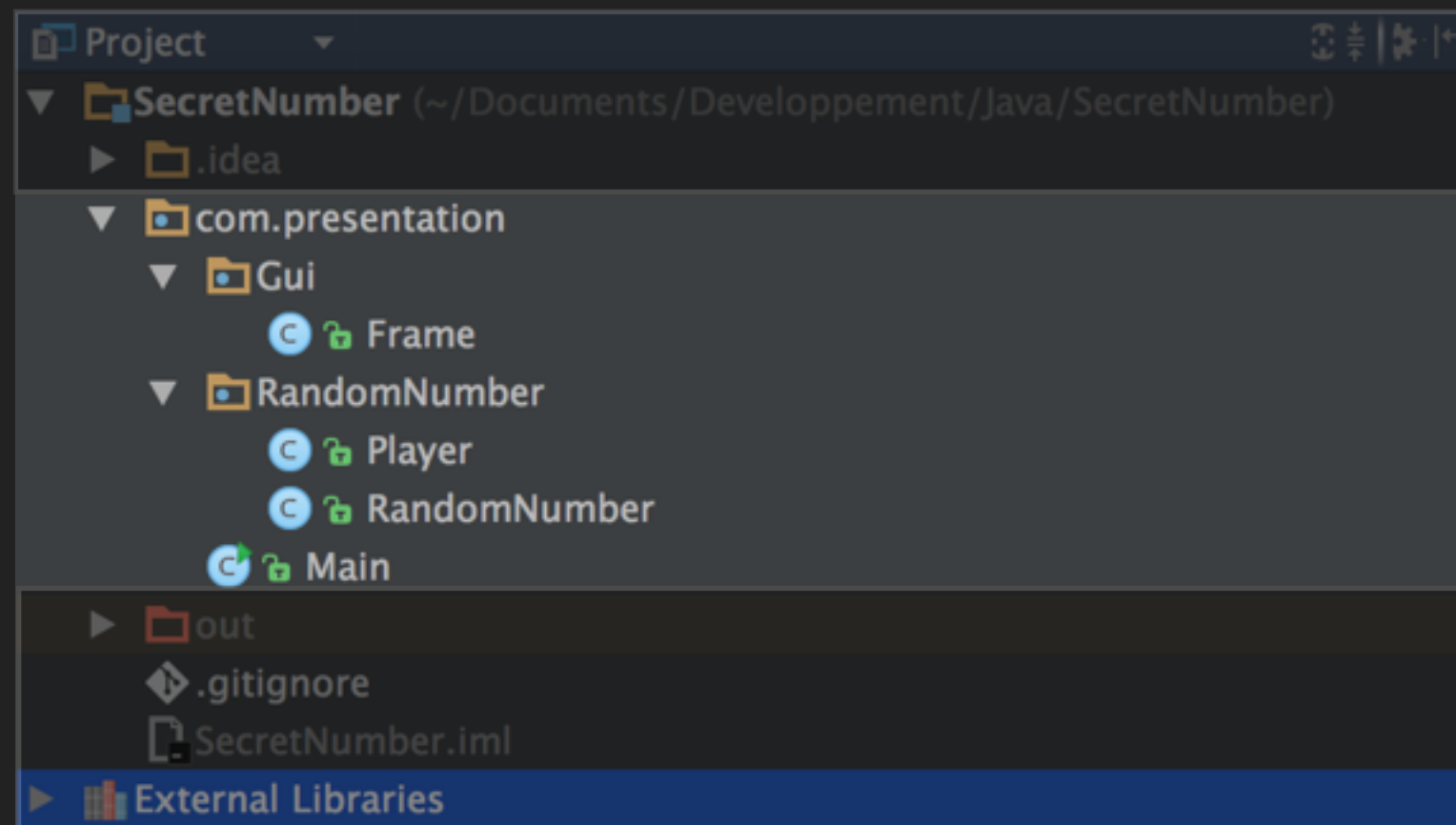
(Less | More ...)

50

Try this number

ORGANISER SON PROJET

- ▶ Plusieurs fichier .java
- ▶ Plusieurs packages



L'IMPLÉMENTATION DES OBJETS

LES ATTRIBUTS ET LE CONSTRUCTEUR

```
public class Person {  
    private String name;  
    private int age;  
    private boolean isAMale;  
  
    public Person() {  
    }  
  
    public Person(String name, int age, boolean isAMale) {  
        this.name = name;  
        this.age = age;  
        this.isAMale = isAMale;  
    }  
}
```

LES GETTERS & SETTERS

```
public String getName() { return name; }
```

```
public void setName(String name) { this.name = name; }
```

```
public int getAge() { return age; }
```

```
public void setAge(int age) { this.age = age; }
```

```
public boolean isAMale() { return isAMale; }
```

```
public void setAMale(boolean AMale) {  
    isAMale = AMale;  
}
```

```
}
```

LES MÉTHODES

```
public int getAgeInTenYears() {  
    int ageInTenYears = this.age + 10;  
    return ageInTenYears;  
}
```

```
public String whatGender() {  
    if (isAMale) {  
        return "Male";  
    } else {  
        return "Female";  
    }  
}
```

UTILISER LES OBJETS

```
Person amalric = new Person("Amalric", 20, true);  
  
int ageInTenYears = amalric.getAgeInTenYears();  
String gender = amalric.whatGender();
```


UTILISER LES OBJETS

```
Person amalric = new Person("Amalric", 20, true);  
int ageInTenYears = amalric.getAgeInTenYears();
```

```
public int getAgeInTenYears() {  
    int ageInTenYears = this.age + 10;  
    return ageInTenYears;  
}
```

LA CLASS

```
Person.java x REAME.MD x
1  /**
2   * (commentaires)
3   *
4   * @author Amalric Lombard de Buffières <amalric.debuffieres@icloud.com>
5   * @version 1.0.0
6   */
7  public class Person {
8      private String name;
9      private int age;
10     private boolean isAMale;
11
12     public Person() {
13     }
14
15     public Person(String name, int age, boolean isAMale) {
16         this.name = name;
17         this.age = age;
18         this.isAMale = isAMale;
19     }
20
21     public String getName() { return name; }
24     public void setName(String name) { this.name = name; }
27     public int getAge() { return age; }
30     public void setAge(int age) { this.age = age; }
33     public boolean isAMale() { return isAMale; }
36     public void setAMale(boolean AMale) { isAMale = AMale; }
39
40
41     public int getAgeInTenYears() {
42         int ageInTenYears = this.age + 10;
43         return ageInTenYears;
44     }
45
46     public String whatGender() {
47         if (isAMale) {
48             return "Male";
49         } else {
50             return "Female";
51         }
52     }
53 }
54
```

L'OBJET

```
Person alice = new Person("Alice", 42, false);  
Person bob = new Person("Bob", 1337, true);
```

LES TABLEAUX

```
int listSize = 5;  
int intTable[] = new int[listSize];  
  
for (int it = 0; it < listSize; it++) {  
    intTable[it] = it;  
}  
  
for (int it = 0; it < listSize; it++) {  
    System.out.print(intTable[it] + " ");  
}
```

0 1 2 3 4

LES LISTES

```
List<Person> personList = new ArrayList<>();  
personList.add(new Person());  
personList.get(0);
```

LES MAP

```
Map<String, Person> userList = new HashMap<>();  
userList.put("Bob", new Person("Bob", 42, true));  
userList.put("Alice", new Person("Alice", 76, false));  
  
Person bob = userList.get("Bob");  
Person alice = userList.get("Alice");
```

GIT



DES QUESTION ?