

Kata Eratosthène avec IntelliJ

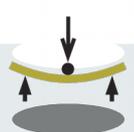
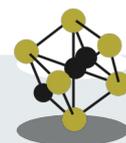
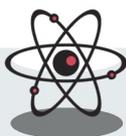
Refonte d'un code avec IntelliJ IDEA

Soit l'exemple suivant d'un code étudiant du crible d'Ératosthène. Nous listons ici les étapes et les raccourcis claviers d'IntelliJ IDEA utilisés pour améliorer la qualité du code.

1. Code initial du crible d'Ératosthène

Remarque : Tous les raccourcis présentés dans cet exercice ont une correspondance dans les menus IntelliJ : Refactor, Code et Run. Vous pouvez donc retrouver les raccourcis en ouvrant les menus concernés.

```
1 package fr.ensicaen.se.demo_5;
2 import java.util.*;
3 /**
4  * Cette classe génère les nombres premiers jusqu'à un maximum
5  * spécifié par l'utilisateur. L'algorithme utilise le crible d'Ératosthène
6  * <p>
7  * Ératosthène de Cyrène, b. c. 276 BC, Cyrène, Libye --
8  * d. c. 194, Alexandrie. Le premier homme à calculer la
9  * circonférence de la Terre.<p>
10 * L'algorithme est plutôt simple. Étant donné un tableau d'entiers
11 * commençant à 2. Éliminer tous les multiples de 2. Chercher le
12 * prochain entier non éliminé, et éliminer tous ses multiples.
13 * Répéter jusqu'à la racine carré du maximum.
14 *
15 * @author Alphonse
16 * @version 13 Feb 2012
17 */
18 public class Eratosthene
19 {
20     public static void main( String String[] args ) {
21         int[] nombres = Eratosthene.geneNombresPremiers(100);
22         for (int i = 0; i < nombres.length; i++) {
23             System.out.print(nombres[i] + ", ");
24         }
25         System.out.println("");
26     }
27
28     /**
29     * @param valeurMax le nombre limite de la génération.
30     */
31     public static int[] genereNombresPremiers( int valMax )
32     {
33         if (valMax >= 2) // le seul cas valide
34         {
```



```

35     // déclarations
36     int s = valMax + 1; // taille du tableau
37     boolean[] f = new boolean[s];
38     int i;
39     // initialise le tableau à vrai.
40     for (i = 0; i < s; i++)
41         f[i] = true;
42     f[0] = f[1] = false; // se débarrasser des nombres premiers connus
43     // crible
44     int j;
45     for (i = 2; i < Math.sqrt(s) + 1; i++)
46     { // si i n'est pas éliminé, éliminer ses multiples.
47         if (f[i] != false)
48             for (j = 2 * i; j < s; j += i)
49                 f[j] = false; // les multiples ne sont pas premiers
50     }
51     // combien de premiers sont faits ?
52     int compteur = 0;
53     for (i = 0; i < s; i++)
54         if (f[i] != false)
55             compteur++; // ajoute au compteur
56     int[] premiers = new int[compteur];
57     // mettre les nombres premiers dans le tableau des résultats
58     for (i = 0, j = 0; i < s; i++)
59         if (f[i] == true) // si premier
60             premiers[j++] = i;
61     return premiers; // retourne les nombres premiers
62 }
63 else // valeurMax < 2
64     // retourne un tableau nul si l'entrée est mauvaise.
65     return new int[0]; /
66 }
67 }

```

1.1 Commencer par exécuter le programmer

CTRL+Shift+F10

1.2 Réécriture du cartouche au-dessus de la classe.

1. Supprimer la partie HTML du commentaire : `/**` et les deux `<p>`. Ce n'est pas une API donc pas de texte de documentation Javadoc.
2. Lignes 7 à 9, supprimer les commentaires concernant Ératosthène (**CTRL+Y** sur le bloc sélectionné). Ce n'est pas un cours d'histoire.
3. Lignes 15-16, supprimer `@version` et `@author` (**Ctrl+Y** sur les 2 lignes). Ces commentaires relèvent d'un gestionnaire de version comme `git` et pas des commentaires.

1.3 Mettre les `{}` sur les instructions unilignes

4. Utiliser le reformatage automatiquement qui le fait automatiquement (**CTRL+ALT+L** qui est le raccourci pour le menu `Code::Format code`).
5. Finir à la main parce que les commentaires empêchent le

reformatage complet (lignes 32, 47, 66, 72).

6. Vérifier que le programme fonctionne toujours avec **CTRL+Shift+F10**.

1.4 Renommer le paramètre de la méthode statique

7. Renommer le paramètre en `nombreMaximum` (**Shift+F6** ou menu Refactor:Rename). Remarque que le nom actuel est en désaccord avec la documentation HTML.
8. Supprimer la documentation Javadoc (**CTRL+W** en ligne 27 **CTRL+Y**). Encore une fois, nous ne développons pas une API.

1.5 Éliminer les commentaires inutiles

9. Ligne 31 : `// Déclaration` (**Ctrl+Y** sur la ligne pour la supprimer).
10. Ligne 35 : `// Initialise le tableau à vrai` (**Ctrl+Y**).
11. Ligne 56 : `// Ajoute au compteur` (**Ctrl+Y**). Mettre le curseur sur le début du commentaire et faire **Shift+End puis Delete**.
12. Ligne 68 : `// Retourne les nombres entiers`. Même procédure que précédemment.

1.6 Inverser le test de validité

Il est toujours préférable d'évacuer les cas particuliers en premiers.

13. Inverser le test en se plaçant sur le `if` puis **Alt+Enter** puis « invert if condition ».
14. Supprimer les commentaires `// le seul cas valide` et `// valeurMax < 2`, puis reformater le code entier avec **CTRL+ALT+L**.

En fait, nous allons faire trois cas différents :

1. Taper le code suivant à la place de l'ancien. Pour cela, on s'aide de la complétion qui s'affiche automatiquement quand on commence un mot. Puis, **Alt+enter** sur `IllegalArgumentException` pour ajouter le `throw` dans la signature de la méthode `genereNombresPremiers` :

```

1  if (nombreMaximum < 1) {
2      throw new IllegalArgumentException(Integer.toString(nombreMaximum));
3  } else if ( nombreMaximum < 2) {
4      return new int[0];
5  } else {

```

1.7 Remplacer `s` par `f.length`

Le tableau `f` porte sa longueur, il ne faut donc pas utiliser de variable non liée comme `s` :

15. Mettre la valeur de `s` dans la taille du tableau `f` à ligne :

```
boolean[] f = new boolean[s];
```

 Se positionner sur `s` et remplacer la variable `s` par sa définition (**Ctrl+Alt+N** ou le menu Refactor::Inline Method).
16. Supprimer la ligne de la déclaration de `s` (**Ctrl+Y**).
17. Remplacer chaque occurrence de `s` par `f.length` en utilisant le multicurseur :
 1. Se positionner sur le premier `s`.
 2. **Alt+J** pour sélectionner chaque occurrence de `s` (ici 4 fois).
 3. Remplacer `s` par `f.length` en tapant le texte.
 4. Terminer par la touche **Esc**.

1.8 Renommer « `f` » en « `candidats` »

18. Changer le nom `f` en `candidats` (**Shit+F6**).

1.9 Décomposition de la méthode en sous méthodes plus explicites

Pour chaque partie, sélectionner le code puis faire **Ctrl+Alt+M** (ou menu Refactor::Extract Method).

19. `InitialiseTableauCandidatsMoinsNombresPremiersConnus()`.
 pour les lignes 35 à 39 (`int i; ... candidats[0]=candidats[1]=false;`).
20. Améliorer le code la méthode. Mettre le curseur sur `int i;` et **ALT+ENTER** et choisir « Move declaration closer to usage ». Faire commencer la boucle à 2.
 Supprimer le commentaire `// se débarrasser des nombres premiers connus`
21. `crible()`
 Sélectionner le code concerné puis **Ctrl+Alt+M**.
22. Remplacer les deux lignes de code

```
if(andidats[i] != false)
```

 par `if (candidats[i] == true)`
 Construire une nouvelle méthode avec l'instruction `candidats[i] == true` nommée `estPremier(candidats[i])`.

Supprimer le code naïf `candidat[i] == true;` **ALT+ENTER** sur le ligne de code et choisir « Simplify ».

23. Faire de `candidats` un attribut de la classe avec le nom `_candidats` (**Ctrl+Alt+F** puis donner le nom).
24. Changer le prototype des deux premières fonctions en supprimant le paramètre `candidats` (**Ctrl+F6** puis choisir « - » sur le paramètre à supprimer).
25. Avec le reste : `construitNombresPremiersAvecCandidatsRestants`. Sélectionner le code concerné puis **Ctrl+Alt+M**.

1.10 Exécution

26. Exécution **Shit+F10**
27. puis **Shit+Esc** pour fermer la console.

1.11 Code propre ?

Vérifier que vous n'avez plus de lignes non conformes signalées en jaune rendant compte du fait que le code n'est pas propre.

Lancer l'analyse globale du code :menu Analyse:: Inspect Code.

1.12 Code final

```

6 package fr.ensicaen.se.demo_5;
7 /*
8  * Cette classe génère les nombres premiers jusqu'à un maximum
9  * spécifié par l'utilisateur. L'algorithme utilise le crible d'Ératosthène
10 *
11 * L'algorithme est plutôt simple.
12 * Étant donné un tableau d'entiers commençant à 2.
13 * Éliminer tous les multiples de 2.
14 * Chercher le prochain entier non éliminé, et éliminer tous ses multiples.
15 * Répéter jusqu'à la racine carré du maximum.
16 */
17 public class Eratosthene {
18     public static void main( String[] args ) {
19         int[] nombres = Eratosthene.genereNombresPremiersInferieursA(100);
20         for (int nombre : nombres) {
21             System.out.print(nombre + ", ");
22         }
23         System.out.println("");
24     }
25
26     private static boolean[] _candidats;
27
28     public static int[] genereNombresPremiersInferieursA( int nombreMaximum ) {
29         if ( nombreMaximum < 1 ) {
30             throw new IllegalArgumentException(Integer.toString(nombreMaximum));
31         } else if ( nombreMaximum < 2 ) {
32             return new int[0];

```

```
33     } else {
34         _candidats = new boolean[nombreMaximum + 1];
35         initialiseTableauCandidatsSansNombresPremiersConnus();
36         crible();
37         return construitNombresPremiersAvecCandidatsRestants();
38     }
39 }
40 private static int[] construitNombresPremiersAvecCandidatsRestants() {
41     int nombreNombresPremiers = 0;
42     for (boolean _candidat : _candidats) {
43         if (isPremier(_candidat)) {
44             nombreNombresPremiers++;
45         }
46     }
47     int[] premiers = new int[nombreNombresPremiers];
48     for (int i = 0, j = 0; i < _candidats.length; i++) {
49         if (isPremier(_candidats[i])) {
50             premiers[j++] = i;
51         }
52     }
53     return premiers;
54 }
55 private static void crible() {
56     for (int i = 2; i < Math.sqrt(_candidats.length) + 1; i++) {
57         if (isPremier(_candidats[i])) {
58             for (int j = 2 * i; j < _candidats.length; j += i) {
59                 _candidats[j] = false;
60             }
61         }
62     }
63 }
64 private static boolean isPremier( boolean candidat ) {
65     return candidat;
66 }
67 private static void initialiseTableauCandidatsSansNombresPremiersConnus() {
68     for (int i = 2; i < _candidats.length; i++) {
69         _candidats[i] = true;
70     }
71 }
72 }
```