

Tutorat Info 2021 - JAVA

1 - Les bases : Types et logiques de bases

Markus Person (aka Notch), le créateur du jeu Minecraft a besoin de ton aide ! En effet, il vient de se réveiller et tout le code de Minecraft a disparu !!! À vous de l'aider ! (c'est promis, je trouverai une meilleure histoire la prochaine fois)

Exercice 1 : Le menu principal

Ta mission

Il faut tout recommencer, alors Notch décide de refaire le menu principal en premier.

Le menu principal est tel qu'il suit :

```
-----Minecraft v1.11.3-----  
  
Bienvenue <NOM DU JOUEUR> !  
  
Tu as déjà joué <JOURS>j <HEURES>h <MINUTES>m <SECONDES>s
```

Dans le cas où le nombre de jours est un multiple de 365 et que le nombre d'heures, le nombre de minutes et le nombre de secondes sont tous égaux à 0, alors il faut afficher :

```
-----Minecraft v1.11.3-----  
  
Bienvenue <NOM DU JOUEUR> !  
  
Tu as déjà joué <JOURS>j <HEURES>h <MINUTES>m <SECONDES>s  
  
Merci d'avoir joué à Minecraft pendant <NOMBRE D'ANNÉES> année(s) !
```

Consignes

Le code de départ est le suivant :

```
// classe principale  
public class Minecraft {  
  
    // constantes dont il faut se servir  
    private static int VERSION_MAJEURE = 1;  
    private static int VERSION_MINEURE = 11;  
    private static int NUMERO_REVISION = 3;
```

```

private static String NOM_JEU = "Minecraft";
private static char LIGNE = '-';
private static int NB_LIGNES = 9;

private static int[] tempsJouéJHM = new int[] {
    365, // Jours
    23,  // Heures
    3    // Minutes
};
private static float tempsJouéSecondes = 16.3;

/**
 * Méthode main affichant le menu principal
 * @param args Arguments de lancement (à ignorer)
 * @return Pas de valeur retournée (type void)
 */
public static void main(String[] args) {
    String affichage = "Quel est ton nom ?";

    System.out.println(affichage);

    Scanner sc = // création du Scanner
    String nomJoueur = // récupération du nom par le scanner

    // construction de l'affichage du menu principal

    System.out.println(affichage);
}
}

```

Complétez le code sans créer de variables, vous devrez uniquement utiliser les constantes existantes, les opérateurs (+, / ...) et les instructions de *flow control* (if, else, for ...).

Indices et mémos

private -> Utilisable uniquement dans sa classe

public -> Utilisable depuis n'importe quelle classe

static -> Pas besoin d'avoir une *instance* de la classe pour s'en servir

Récupération d'une ligne écrite au clavier :

```

Scanner sc = new Scanner(System.in); // création de l'outil de lecture
String ligneEcrit = sc.nextLine(); // lecture

```

Concaténation de strings :

- "partie1" + "partie2" -> "partie1partie2"

- `string = "a"; string += "b"; -> "ab"`

Vérification d'une condition booléenne (= `false` ou `true`) :

```
boolean jeSuisVrai = true;

if(jeSuisVrai) {
    // on arrivera toujours ici
} else {
    // on arrivera jamais ici
}

if(!jeSuisVrai) {
    // on arrivera jamais ici
} else {
    // on arrivera toujours ici
}
```

Boucle exécutant N fois une suite d'instructions :

```
int n = 10; // on veut faire 10 tours

// int i = 0 -> on commence au tour 0
// i++ -> on augmente le numéro du tour à chaque tour
// i < n -> on s'arrête lorsque le numéro du tour est égal à N
// (on fait donc les tours n° 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
// (mais pas le n°10), c'est à dire 10 tours en tout)
for(int i = 0; i < n; i++) {
    // tout ce qui se passe ici sera répété N fois
}
```

Création et manipulation d'un tableau :

```
String[] tableau = new String[3];
tableau[0] = "ce code ";
tableau[1] = "initialise ";
tableau[2] = "un tableau\n";

System.out.println(tableau[0]); // "ce code"

// Boucle qui affiche "ce code initialise un tableau\n"
for(int i = 0; i < 3; i++) {
    System.out.print(tableau[i]);
}

// Petit raccourci qui fait la même chose
for(String partieMessage : tableau) {
```

```
        System.out.print(partieMessage);  
    }
```

Correction

Version de base

```
public class Minecraft {  
  
    ...  
  
    public static void main(String[] args) {  
        String affichage = "Quel est ton nom ?";  
  
        System.out.println(affichage);  
  
        Scanner sc = new Scanner(System.in);  
        String nomJoueur = sc.nextLine();  
  
        // Ajout du premier -----  
        for(int i = 0; i < NB_LIGNES; i++) {  
            // concaténer un String et un char est possible !  
            affichage += LIGNE;  
        }  
  
        // Ajout du titre  
        affichage += NOM_JEU;  
        affichage += " ";  
        affichage += Integer.toString(VERSION_MAJEURE);  
        affichage += ".";  
        affichage += Integer.toString(VERSION_MINEURE);  
        affichage += ".";  
        affichage += Integer.toString(NUMERO_REVISION);  
  
        // Ajout du deuxième -----  
        for(int i = 0; i < NB_LIGNES; i++) {  
            affichage += LIGNE;  
        }  
  
        // Passage à la ligne et saut d'une ligne  
        affichage += "\n\n";  
  
        // Message de bienvenue  
        affichage += "Bienvenue " + nomJoueur;  
  
        affichage += "\n\n";  
  
        // Affichage du temps de jeu  
        affichage += "Tu as déjà joué " + tempsJouéJHM[0] + "j";  
        affichage += " " + tempsJouéJHM[1] + "h";  
        affichage += " " + tempsJouéJHM[2] + "m";  
    }  
}
```

```

    affichage += " " + tempsJouéSecondes + "s.";

    // on vérifie que Le nb de jours joué est multiple de 365
    if(tempsJouéJHM[0] % 365 == 0) {
        affichage += "\n\n";
        affichage += "Merci d'avoir joué à " + NOM_JEU;
        affichage += " pendant " + (tempsJouéJHM[0] / 365) + " année(s)";
    }

    System.out.println(affichage);
}

}

```

Amélioration 1 : Optimisation avec StringBuilder

La concaténation de String en Java est parfois lente, penser à utiliser **StringBuilder**.

```

public class Minecraft {

    ...

    public static void main(String[] args) {
        System.out.println("Quel est ton nom ?");

        Scanner sc = new Scanner(System.in);
        String nomJoueur = sc.nextLine();

        // création du StringBuilder
        StringBuilder sb = new StringBuilder();

        // Ajout du premier -----
        for(int i = 0; i < NB_LIGNES; i++) {
            // ajouter des éléments au StringBuilder avec append
            sb.append(LIGNE);
        }

        // Ajout du titre
        sb.append(NOM_JEU);
        sb.append(" ");
        sb.append(VERSION_MAJEURE); // append fonctionne avec n'importe quel type
        !
        sb.append(".");
        sb.append(VERSION_MINEURE);
        sb.append(".");
        sb.append(NUMERO_REVISION);

        // Ajout du deuxième -----
        for(int i = 0; i < NB_LIGNES; i++) {
            sb.append(LIGNE);
        }
    }
}

```

```
// Passage à la ligne et saut d'une ligne
sb.append("\n\n");

// Message de bienvenue
sb.append("Bienvenue ");
sb.append(nomJoueur);

sb.append("\n\n");

// Affichage du temps de jeu avec String.format
// https://dzone.com/articles/java-string-format-examples
sb.append(String.format("Tu as déjà joué %dj", tempsJouéJHM[0]));
sb.append(String.format(" %dh", tempsJouéJHM[1]));
sb.append(String.format(" %dm", tempsJouéJHM[2]));
sb.append(String.format(" %fs.", tempsJouéSecondes));

// on vérifie que le nb de jours joué est multiple de 365
if(tempsJouéJHM[0] % 365 == 0) {
    sb.append("\n\n");
    sb.append(String.format("Merci d'avoir joué à %s", NOM_JEU));
    sb.append(" pendant %d année(s)", tempsJouéJHM[0] / 365);
}

// Passage d'un StringBuilder à un String qu'on peut afficher
String affichage = sb.toString();
System.out.println(affichage);
}

}
```

Amélioration 2 : Découpage en fonctions

Faire une grosse fonction main est pas super lisible pour les autres, il vaut mieux découper en morceaux plus petits et bien nommés, afin de permettre à quelqu'un découvrant le code de mieux s'y retrouver.

```
public class Minecraft {

    ...

    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder();
        String nomJoueur = obtenirNomJoueur();

        sb.append(construireLigne());

        sb.append(construireTitre());

        sb.append(construireLigne());

        sb.append(sautDeLigne());
    }
}
```

```
        sb.append(construireBienvenue());

        sb.append(sautDeLigne());

        sb.append(construireTempsDeJeu());

        if(estAnniversaire()) {
            sb.append(sautDeLigne());

            sb.append(construireMessageAnniversaire());
        }

        String affichage = sb.toString();
        System.out.println(affichage);
    }

    private static String obtenirNomJoueur() {
        System.out.println("Quel est ton nom ?");
        Scanner sc = new Scanner(System.in);
        return sc.nextLine();
    }

    private static String construireLigne() {
        StringBuilder sb = new StringBuilder();

        for(int i = 0; i < NB_LIGNES; i++) {
            sb.append(LIGNE);
        }

        return sb.toString();
    }

    private static String construireTitre() {
        StringBuilder sb = new StringBuilder();

        sb.append(NOM_JEU);
        sb.append(" ");
        sb.append(VERSION_MAJEURE);
        sb.append(".");
        sb.append(VERSION_MINEURE);
        sb.append(".");
        sb.append(NUMERO_REVISION);

        return sb.toString();
    }

    private static String sautDeLigne() {
        return "\n\n";
    }

    private static String construireBienvenue(String nomJoueur) {
        return String.format("Bienvenue %s", nomJoueur);
    }
}
```

```
private static String construireTempsDeJeu() {
    StringBuilder sb = new StringBuilder();

    sb.append(String.format("Tu as déjà joué %dj", tempsJouéJHM[0]));
    sb.append(String.format(" %dh", tempsJouéJHM[1]));
    sb.append(String.format(" %dm", tempsJouéJHM[2]));
    sb.append(String.format(" %fs.", tempsJouéSecondes));

    return sb.toString;
}

private static boolean estAnniversaire() {
    return tempsJouéJHM[0] % 365 == 0;
}

private static String construireMessageAnniversaire() {
    StringBuilder sb = new StringBuilder();

    sb.append(String.format("Merci d'avoir joué à %s", NOM_JEU));
    sb.append(" pendant %d année(s)", tempsJouéJHM[0] / 365);

    return sb.toString;
}
}
```