

Rapport SAE S1.01- Implémentation d'un besoin client

Contents

Sommaire :	2
Tirage au sort d'un étudiant :	2
Stockage du tirage dans un Tableau :	3
Contrainte : Ne pas tirer au sort un étudiant qui est passé :	3
Contrainte : Tirer au sort en priorité les personnes qui ne sont pas passées :	3
Jeux d'essai :	3

Rapport SAE S1.01- Implémentation d'un besoin client

Le travail qui nous a été demandé pour cette S.A.É est de développer une application qui pourra être utilisée par les professeurs du département informatique pour tirer au hasard un étudiant interroger. Pour cela, nous utiliserons toutes les bases de la programmation apprises jusque-là tout en respectant les normes de codages.

Sommaire :

- Tirage au sort d'un étudiant
- Stockage du tirage dans un tableau
- Contrainte : Ne pas tirer un étudiant qui est passer
- Contrainte : Tirer en priorité les étudiants qui ne sont pas passer
- Jeux d'essai

Tirage au sort d'un étudiant :

Pour tirer au sort un étudiant, nous avons utilisé la méthode "Math.random()". Cette methode permet de tirer au sort un nombre entier. Dans notre cas, nous l'avons utilisé dans un tableau, et le nombre entier sélectionner est l'indice de l'étudiant tirer au sort dans notre tableau. :

```
/** traite le choix 2
 * @param vous pouvez en ajouter
 * @return un étudiant correspondant aux critères
 */
public static int traiterChoix2(int pfhistorique[], int pfNbEltHisto){

    int tirage = (int) (Math.random() * promo.length);

    return tirage;
}
```

Ici, la fonction Math.random() sélectionne un nombre aléatoire parmi toutes les cases disponibles dans le tableau promo, c'est à dire la liste des étudiants.

Stockage du tirage dans un Tableau :

Maintenant que nous avons tiré au sort un étudiant, il faut le stocker dans un autre tableau, qui sera utilisé comme « historique ». Cela permettra par la suite de permettre de voir si pour les prochains tirages, l'étudiant sélectionné a déjà été tiré au sort ou pas.

```
System.out.println(promo [tirage][ 0]+" " +promo [tirage][1]); //élève tirer au sort
if (nbEltHisto < promo.length){
    historique[ nbEltHisto ] = tirage ; //stockage de l'étudiant dans le tableau
    nbEltHisto = nbEltHisto + 1 ; //le nbe d'étudiant stockée augmente de 1
    System.out.println ("le nouveau nombre d'étudiant tiré est " + " " + nbEltHisto) ;    //affiche le nbe d'étudiant tiré
}else{
    System.out.println("Fin, Réinitialisation de l'historique !");
    nbEltHisto = 0; //remise à 0 du l'historique
}
```

Dans cette partie de notre programme, à chaque tirage, l'indice (donc l'étudiant) sera sauvegarder dans notre tableau « Historique » qui sera initialement à 0 et le nombre d'étudiant stockée dedans augmentera de 1 jusqu'à ce que tous les étudiants seront passés. Lorsque celle-ci sera atteinte, le tableau se réinitialisera afin de recommencer les tirages.

Contrainte : Ne pas tirer au sort un étudiant qui est passé :

Afin de ne pas tirer au sort un étudiant qui est déjà passer auparavant avant tous les autres, nous avons choisis comme méthode de stocker cette étudiant dans une variable.

```
int tirage = traiterChoix2(historique, nbEltHisto);
while (nbEltHisto == tirage ) {
    tirage = traiterChoix2(historique, nbEltHisto);
}
```

Ici, à chaque tirage, une boucle va permettre de voir si l'indice de l'étudiant tiré au sort est déjà présent dans l'historique. Si c'est le cas, un nouveau tirage aura lieu et le dernier tirage qui a eu lieu ne sera pas valide et donc ne sera pas stocker.

Contrainte : Tirer au sort en priorité les personnes qui ne sont pas passées :

La dernière contrainte qui nous a été posée a été de tirer au sort en priorité les personnes qui ne sont jamais passés. En effet, pour qu'un étudiant puisse passer de nouveau, tous les étudiants de la liste doivent être passés au moins une fois. Ceci s'effectue déjà grâce au tableau qui stock les étudiants qui sont déjà passés puisque pour qu'un étudiant puisse repasser, les autres doivent déjà être passés. Les étudiants qui ne sont pas passés sont donc bien tirés en priorité.

Jeux d'essai :

Pour une meilleur clarté, nous avons réduit la liste des étudiants à 5 !

1. **Affichage du Menu au lancement du programme :**

```
nombre de lignes : 5  
nombre de lignes : 5  
Il y a : 5 etudiants  
  
/*****/  
                1 - un etudiant dans la promo  
                2 - affiche la promo  
                0 - Quitter  
/*****/  
  
Choisissez parmi les 3 propositions :
```

Nous pouvons voir dans ce menu d'une part le nombre d'étudiants présents, 5 en l'occurrence ici. De plus nous pouvons apercevoir les interaction disponibles : la touche 1 pour tirer au sort un étudiant ; la touche 2 pour afficher le nom des étudiants dans la promo ; la touche 0 pour sortir du programme.

2. **Le tirage au sort**

```
Choisissez parmi les 3 propositions : 1
Bu Amede
le nouveau nombre d'étudiant tirer est 1

/*****/
    1 - un etudiant dans la promo
    2 - affiche la promo
    0 - Quitter
/*****/

Choisissez parmi les 3 propositions : 1
Cape Andy
le nouveau nombre d'étudiant tirer est 2

/*****/
    1 - un etudiant dans la promo
    2 - affiche la promo
    0 - Quitter
/*****/

Choisissez parmi les 3 propositions : 1
Auboisdormant Abel
le nouveau nombre d'étudiant tirer est 3

/*****/
    1 - un etudiant dans la promo
    2 - affiche la promo
    0 - Quitter
/*****/

Choisissez parmi les 3 propositions : 1
Balmaske Alonzo
le nouveau nombre d'étudiant tirer est 4

/*****/
    1 - un etudiant dans la promo
    2 - affiche la promo
    0 - Quitter
/*****/

Choisissez parmi les 3 propositions : 1
Afritt Barack
le nouveau nombre d'étudiant tirer est 5

/*****/
    1 - un etudiant dans la promo
    2 - affiche la promo
    0 - Quitter
/*****/
```

```
Choisissez parmi les 3 propositions : 1
le nouveau nombre d'étudiant tiré est 5
Afritt Barack
```

```

/*****/
      1 - un etudiant dans la promo
      2 - affiche la promo
      0 - Quitter
/*****/
```

```
Choisissez parmi les 3 propositions : 1
Fin, Réinitialisation de l'historique !
Auboisdormant Abel
```

```

/*****/
      1 - un etudiant dans la promo
      2 - affiche la promo
      0 - Quitter
/*****/
```

On remarque bien ici que les étudiants tirés au sort ne sont pas tirés une fois de plus avant que les autres étudiants aient été tirés.

```
Choisissez parmi les 3 propositions : 0
AU REVOIR ...    ...
```

Nous pouvons quitter le programme quand nous cliquons sur la touche 0