1. Ecrire l'en-tête d'une méthode appelée meth1 n'ayant ni paramètres ni valeur de retour.

```
void meth1()
```

2. Ecrire l'en-tête d'une méthode appelée meth2, n'ayant pas de valeur de retour et prenant comme paramètre un entier

```
void meth2(int i)
```

3. Ecrire l'en-tête d'une méthode appelée meth3, n'ayant pas de valeur de retour et prenant comme paramètre un entier et une chaîne de caractères.

```
void meth3(int i,String chaine)
```

4. Ecrire une méthode ne prenant pas de paramètre et retournant toujours la valeur numérique 0,75.

```
double stupide(){
   return 0.75;
  }
```

5. Ecrire une méthode appelée inverse, prenant un paramètre de type double, et renvoyant son inverse (1/x).

```
double inverse(double x){
    return 1/x;
 }
```

6. Ecrire une méthode qui renvoie un nombre entier aléatoire compris entre 0 et 20.

```
int aleatoire20(){
   return (int)(Math.floor(Math.random()*20));
}
```

7. Ecrire une méthode qui renvoie un nombre entier aléatoire inférieur à une valeur passée en paramètre.

```
int aleatoire(int n){
   return (int)(Math.floor(Math.random()*n));
}
```

8. Ecrire une méthode qui renvoie un nombre entier aléatoire compris entre deux valeurs passées en paramètres.

```
int aleatoireBorne(int a,int b){
      return (int)(Math.floor(Math.random()*(a-b)+b));
}
```

9. Ecrire une méthode qui renvoie un nombre entier aléatoire multiple de 3.

On utilise le même principe qu'à l'exercice 9 pour obtenir un premier entier aléatoire dont la valeur soit inférieur à 2^{31}-1 / 3 (2^{31}-1 est la plus grande valeur qu'on peut écrire dans un int).
On multiplie ensuite cet entier aléatoire par 3, ce qui donnera forcément un multiple de 3 pouvant être codé sous forme de int.

```
int aleatoire3(){
  int max=((int)Math.pow(2,31)-1)/3;
  return (int)(3*Math.floor(Math.random()*max));
}
```

10. Ecrire une méthode qui renvoie le plus petit élément d'un tableau passé en paramètre

Il faut connaître le type des éléments du tableau. Celui-ci doit correspondre à un type ordonné, c'est à dire soit numérique soit char. La valeur de retour doit être de même type
Par exemple pour rechercher le minimum d'un tableau de réels :

```
double minimum(double[] tableau){
          double mini=tableau[0];
          if (tableau.length>0){
            for (int i=0;i<tableau.length;i++){
              if (tableau[i]<mini){
                mini=tableau[i];
              }
```

```
            }
        }
        return mini;
    }
```

11. Ecrire une méthode qui renvoie la somme des éléments d'un tableau passé en paramètre
Ici aussi le type du tableau doit être connu.

```
double somme(double[] tableau){
        double somme=0;
        for (int i=0;i<tableau.length;i++){
          somme=somme+tableau[i];
        }
        return somme;
    }
```

(pour bien faire il faudrait contrôler que la somme obtenue ne dépasse pas la capacité d'un double, car sinon le résultat sera faux!)

12. Ecrire une méthode qui génère un nouveau tableau rempli de 25 valeurs aléatoires inférieures à 100.

```
int[] creeTableau(){
  int [] tableau=new int[25];
  for (int i=0;i<25;i++){
    tableau[i]=(int)(Math.floor(Math.random()*100));
   }
   return tableau;
}
```

On a choisi ici de créer un tableau d'entiers mais on peut remplacer int par n'importe quel autre format numérique

13. Ecrire une méthode qui génère un nouveau tableau rempli de 25 valeurs aléatoires, inférieures à une valeur passé en paramètre.

```
int[] creeTableauMajo(int majorant){
  int [] tableau=new int[25];
  for (int i=0;i<25;i++){
    tableau[i]=(int)(Math.floor(Math.random()*majorant));
   }
   return tableau;
}
```

14. Ecrire une méthode qui génère un nouveau tableau rempli de 25 valeurs aléatoires, comprises entre deux valeurs passées en paramètre.

```
int[] creeTableauBorne(int a,int b){
  int [] tableau=new int[25];
  for (int i=0;i<25;i++){
    tableau[i]=(int)(Math.floor(Math.random()*(a-b)+b));
   }
   return tableau;
}
```

Si vous souhaitez tester ces méthodes dans une classe exécutable il faudra ajouter l'indication static au début de la déclaration de la méthode.