

Objectifs : Consolider les notions de **conditions** et d'**alternatives**. Introduction des boucles à compteur.

1 Retour sur les alternatives

Exercice 1 : Si c'est pas celui-là, c'est que c'est l'autre !

Dans la fonction principale qui suit, essayez sans chercher à comprendre ce que fait le programme, de le compléter par les bons mots-clés.

```
void algorithm() {
    int anjzegzog = readInt();
    double ejigez = readInt();
    .... (ejigez == anjzegzog * ejigez) {
        println("znez_ffez_bpiu");
    } ..... (ejigez == anjzegzog - ejigez) {
        println("znez_ffez_fa_fa_fa");
    } ..... {
        println("io_p'haf");
    }
}
```

2 Nouvelle structure de contrôle : les boucles à compteur

2.1 Potato pirates

Afin de manipuler les notions abordées depuis le début du cours, nous allons utiliser un autre jeu de cartes permettant d'utiliser ces concepts dans un contexte moins technique. Le but du jeu *Potato pirates* est terriblement simple : couler les bateaux de tous les autres joueurs et joueuses afin de devenir le roi ou la reine des pirates. Pour cela, les participant-e-s définissent de courts algorithmes définissant des attaques sur les bateaux adverses.

Au début du jeu, tous les joueurs et joueuses disposent de deux bateaux avec un équipage de 20 personnes (il faut noter cela à l'aide d'un stylo effaçable sur chacun de vos bateaux) ainsi que 5 cartes.

Pour attaquer les bateaux adversaires, il faut définir des stratégies à l'aide de vos cartes en les disposant en dessous de vos bateaux. Attention : on ne peut utiliser plus de 3 cartes pour définir un algorithme d'attaque et une fois celui-ci définit, il ne sera possible de le déclencher qu'au tour suivant (le temps que vos équipiers s'entraînent à cette nouvelle stratégie!).

Lorsque c'est à votre tour de jouer, vous devez d'abord piocher deux cartes puis :

- définir un ou plusieurs algorithmes d'attaque (autant que de bateaux possédés),
- déclencher une attaque d'un algorithme définit lors d'un précédent tour,

Les cartes *interruption* peuvent être utilisées à n'importe quel moment !

Si suite à une attaque un bateau n'a plus d'équipage, alors il coule et est retiré du jeu. Selon le temps disponible, le jeu s'arrête dès qu'un bateau est coulé (partie courte) ou lorsqu'un des joueurs et joueuses a coulé tous les bateaux des adversaires !

Exercice 2 : Les boucles à compteur, c'est for

Voici un programme utilisant une boucle `for`.

```
class ProgrammeFor extends Program{
    void algorithm() {
        for (int compteur=0; compteur<10; compteur=compteur+1) {
            println(compteur);
        }
    }
}
```

1. Quelle est la valeur de `compteur` lors du premier tour de la boucle `for`?
2. Quelle est la valeur de `compteur` lors du dernier tour de la boucle `for`?
3. Combien de fois la fonction `println` sera-t-elle appelée lors de l'exécution de ce programme?
4. Qu'affiche précisément ce programme?
5. Modifier ce programme pour qu'il affiche les multiples de 5 allant de 0 à 50 inclus.

Exercice 3 : Ça tourne en boucle

Voici un programme quelque peu répétitif :

```
class ProgrammeRepetitif extends Program{
  void algorithm() {
    int numero = 1;
    println("affichage_numéro_" + numero);
    numero = numero + 1;
    println("affichage_numéro_" + numero);
    numero = numero + 1;
    println("affichage_numéro_" + numero);
    numero = numero + 1;
    println("affichage_numéro_" + numero);
    numero = numero + 1;
    println("affichage_numéro_" + numero);
    numero = numero + 1;
  }
}
```

1. Qu'affiche précisément ce programme?
2. Écrivez un programme produisant le même résultat mais dont l'algorithme tient en 3 lignes.

Exercice 4 : Déterminer la mention [ALT-CASC]

La mention obtenue aux examens en fin d'année dépend de la moyenne obtenue:

| | | | | | |
|------------------|-------------|--------------|--------------|--------------|---------------|
| 0 <= m < 5 | 5 <= m < 10 | 10 <= m < 12 | 12 <= m < 14 | 14 <= m < 16 | 16 <= m <= 20 |
| TRES INSUFFISANT | INSUFFISANT | PASSABLE | ASSEZ BIEN | BIEN | TRES BIEN |

Concevez un programme `Mention` qui affiche la mention obtenue étant donnée la moyenne saisie (dont on supposera qu'elle est comprise entre 0 et 20).