TP SHIFUMI

Objectif: Réaliser le jeu "pierre feuille ciseau" contre un ordinateur en Java.

Introduction

Dans ce TP, j'ai créé un jeu de SHIFUMI en Java. Le but est de jouer contre l'ordinateur. Ce projet m'a permis d'utiliser ce que j'ai appris en cours : les variables, les boucles, les conditions et la saisie utilisateur.

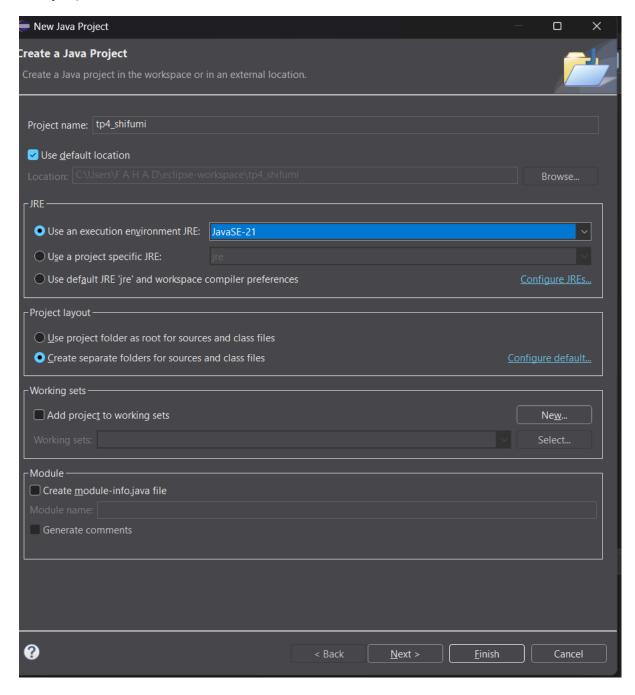
Outils utilisés

Pour faire ce projet, j'ai utilisé :

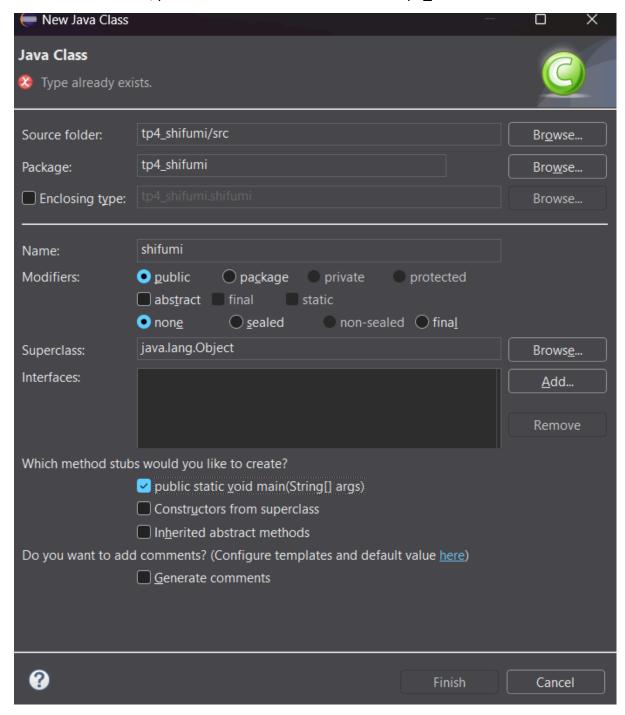
- Le langage Java avec le JDK 17
- L'IDE Eclipse pour écrire mon code
- Un projet nommé tp4_shifumi
- Une classe principale appelée shifumi

Étape 1 : Définir le nombre de points, et ajouter des commentaire sur chaque pour bien comprendre,	
Étape 2 : Choix du joueur,	
Étape 3 : Choix ordinateur,	
Étape 4 : Suspense,	8
Étape 5 : Déterminer le gagnant	
J'ai ajouté une boucle while avant l'étape 2 pour répéter le jeu jusqu'à ce qu'un joue gagne la partie Étape 7: fin de partie, et afficher le gagnant. Et demander de rejouer une partie?	
CONCLUSION	3 11

 Création de projets sur Eclipse, pour cela il faut aller sur file → New → Java Project)



- Création de **classe**, pour cela il faut New > Class > NOM: tp4_shifumi



- Et puis on commence par les **codage**, donc au tout début on a ajouter des commentaire de ce que on vas faire sur chaque etape donc et puis on a mit les 4 règles principales du jeu et puis juste en bas on voit le **résultat** :

le résultat :

Étape 1 : Définir le nombre de points, et ajouter des commentaire sur chaque étape pour bien comprendre,

Une boucle do-while demande 3, 5 ou 10 points. La boucle se répète tant que la valeur n'est pas valide.

Et voici le résultat,

```
voici les régles du jeu :
feuille bat pierre :
Pierre bat ciseau :
ciseau bat feuille :
égalite si 2 éléments similaire :
En combien de points se déroule la partie ? (3.5 ou 10 )
```

```
égalite si 2 éléments similaire :
En combien de points se déroule la partie ? (3.5 ou 10 )
6
vous avez saisie 6
En combien de points se déroule la partie ? (3.5 ou 10 )
5
vous avez saisie 5
```

Étape 2 : Choix du joueur,

Une boucle do-while demande p, f ou c. J'utilise charAt(0) pour récupérer le caractère saisi.

```
//étape 2: Choisir l'outil pour le jeux P,F,C

do {

System.out.println(" choisissez pierre (p), feuille (f) ou ciseau (c):");

choix=sc.next().charAt(0);

System.out.println("vous avez choisi: "+ choix);

ywhile(choix !='p' && choix != 'f' && choix != 'c');

while(choix !='p' && choix != 'f' && choix != 'c');
```

Et voici le résultat:

```
En combien de points se déroule la partie ? (3.5 ou 10 )

5
vous avez saisie 5
choisissez pierre (p), feuille (f) ou ciseau (c):

P
vous avez choisi: P
choisissez pierre (p), feuille (f) ou ciseau (c):
```

Étape 3 : Choix ordinateur,

Je génère un nombre aléatoire entre 1 et 3 avec Math.random(). Des conditions if/else attribuent p, f ou c selon le nombre.

```
//étape 3 : Choix aléatoire de l'ordinateur
char choixordi = 0;

//Déclaration et utilisation de la variable aléatoire
int aleatoire = (int)(Math.random() * 3) + 1;

//Condition d'attribution de p, F et C à partir du random
if(aleatoire == 1) {
choixordi = 'p';}

else if (aleatoire == 2) {
choixordi = 'f';}

else if(aleatoire == 3) {
choixordi = 'c';}

System.out.println("L'ordinateur a choisi: " +choixordi);
}
```

Et voici le **résultat**.

```
En combien de points se déroule la partie ? (3.5 ou 10 )

5

vous avez saisie 5

choisissez pierre (p), feuille (f) ou ciseau (c):

f

vous avez choisi: f

L'ordinateur a choisi: p
```

Étape 4 : Suspense,

J'ai ajouté **Thread.sleep(3000)** dans un try-catch pour une pause de 3 secondes avant d'afficher le résultat.

```
//étape 4 : Attente 2,4 pour le suspense

System.out.println("l'ordinateur a choisi");
// try catch permet de continuer le programme en cas de la fonction thread

try {
  Thread.sleep(3000);
} catch (InterruptedException e) {
  // TODO Auto-generated catch block
  e.printStackTrace();
}
System.out.println("fin du suspense");
```

Et voici le résultat,

```
vous avez choisi: p
L'ordinateur a choisi: p
l'ordinateur a choisi
```

Étape 5 : Déterminer le gagnant

Je compare les choix avec des conditions **if/else.** Si égalité, j'affiche "égalité!". Si le joueur gagne, j'augmente son score. Sinon, j'augmente le score de l'ordinateur. J'affiche ensuite les scores.

```
if (choix == choixordi) {
   System.out.println("égalité !");
} else if ((choix=='p' && choixordi=='c') ||
   (choix=='f' && choixordi=='p') ||
   (choix=='c' && choixordi=='f')) {
   //afficher si joueur gagne ou perd
   System.out.println("vous avez gagné la manche");
   scorejoueur++;
} else {
   System.out.println("l'ordinateur a gagné la manche");
   scoreordi++;
}
```

Voici le résultat,

```
yous avez choisi: c
L'ordinateur a choisi
l'ordinateur a choisi
fin du suspense
égalité!
score : joueur 0 - score ordi : 0
choisissez pierre (p), feuille (f) ou ciseau (c):
```

Etape 6: Boucle des manches,

J'ai ajouté une boucle while avant l'étape 2 pour répéter le jeu jusqu'à ce qu'un joueur gagne la partie.

```
//etape 6: boucle des manches
//faire une boucle qui permet de relancer le choix entre P, F et C jusqu'a ce que le score
// d'un joueur soit égale au nombre de points de la manche choisi par le joueur
while (scorejoueur<nbpoints && scoreordi<nbpoints) {

vous avez choisi: f
l'ordinateur a choisi
L'ordinateur a choisi: p
fin du suspense
vous avez gagné la manche
score : joueur 1 - score ordi : 1
choisissez pierre (p), feuille (f) ou ciseau (c):</pre>
```

Étape 7: fin de partie, et afficher le gagnant. Et demander de rejouer une partie?

J'ai créé une boucle do-while qui englobe tout le jeu (de l'étape 1 à l'étape 6). Cette boucle commence avant l'étape 1 et se termine après avoir demandé si le joueur veut rejouer.

```
//étape 7:
// REJOUER LA PARTIE
do {
scorejoueur = 0;
scoreordi = 0;
```

```
// Afficher le gagnant final
if(scorejoueur == nbpoints) {
    System.out.println("Félicitations! Vous avez gagné la partie!");
} else {
    System.out.println("L'ordinateur a gagné la partie!");
}

// Demander si on rejoue
System.out.println(" voulez vous rejouer ? (o/n)");
reponse = sc.next().charAt(0);
rejouer = (reponse == 'o' || reponse == '0');
} while(rejouer);
System.out.println("Merci d'avoir joué!");
sc.close();
```

Et voici le **résultat**,

```
l'ordinateur a choisi
L'ordinateur a choisi: Ciseaux
fin du suspense
vous avez gagné la manche ! Pierre bat Ciseaux
score : joueur 5 - score ordi : 1
Félicitations! Vous avez gagné la partie!
voulez vous rejouer ? (o/n)
```

Etape 8 : J'ai ajouté la conversion des lettres en noms complets. Après la saisie du joueur, des conditions if/else transforment 'p' en "Pierre", 'f' en "Feuille" et 'c' en "Ciseaux". Cette conversion rend l'affichage plus clair,

```
String symboleJoueur = "";
String symboleOrdi = "";
```

CONCLUSION

J'ai réussi à développer un jeu de Shifumi complet et fonctionnel. Le programme permet de choisir le nombre de points, joue autant de manches que nécessaire jusqu'à ce qu'un joueur gagne, affiche les résultats de manière claire avec les noms complets des objets au lieu de simples lettres, et permet de rejouer autant de fois qu'on le souhaite.

Ce TP m'a permis de mettre en pratique toutes les notions vues en cours : les variables, les différents types de boucles (do-while et while), les conditions if-else, et la gestion des entrées utilisateur avec Scanner. Le code est bien structuré avec des commentaires qui expliquent chaque étape, ce qui facilite la compréhension et les modifications futures.