

Morgado-Samagaio Jonathan

BPOO / TP1 Eclipse

2.1.2) Donner la valeur de chaque variable en fin d'exécution.

Point d'arrêt 1 : i, j, d, e = non initialisée.

Point d'arrêt 2 : i = 1, j = 2, d = 0, e = 10,5.

Point d'arrêt 3 : i = 1, j = 2, d = 10,5, e = 10,5.

2.2.3) Après correction, donner la valeur de chaque variable en fin d'exécution.

Point d'arrêt 1 : i, j, d, e = non initialisée.

Point d'arrêt 2 : Pas atteint car condition pas réalisé.

Point d'arrêt 3 : i = 100, j = 101, d = 0, e = 10,5.

2.2.4) La valeur de d a-t-elle été modifiée en <2> ?

Non car la condition du if n'est pas validée.

2.2.5) La valeur de j a-t-elle été modifiée en <3> ?

Lorsque le point d'arrêt 3 est atteint, la valeur n'est pas encore modifiée. Elle sera modifiée juste après ce point d'arrêt.

3.1.2) Exécuter pas à pas ce programme et indiquer ce qui est affiché

Point d'arrêt 1 : argv = String[0] (id=16), tab3 = (id=17).

Seul les identifiants sont donnés.

3.1.3) Quelles sont les valeurs de i en <2>, <3>, <4> ?

Point d'arrêt 2 : i = 5.

Point d'arrêt 3 : i = 5.

Point d'arrêt 4 : i = 3.

4.1.4) A cet endroit quelles sont les valeurs de ageDebut et anneesEtudes ?

ageDebut = 23 ;0, anneesEtudes = 5.

4.1.5) Quelles sont les valeurs de pfA et pfD ?

pfA = 23.0, pfD = 5 car ce sont des copies des valeurs d'entrées et la première ligne ne s'est pas exécuté.

4.1.7) Quelle valeur est affectée à ageFinEtudes ?

ageFinEtudes = 29 ;0.

4.2.3) Quelles sont les valeurs de ageDebut et anneesEtudes dans le main() après retour du sous-programme plusPlus() ?

ageDebut = 23.0, anneesEtudes = 5. Les deux valeurs n'ont pas changé.

5.3) Exécuter pas à pas le programme et donner les informations suivantes dans le compte rendu

a) tab (id=21)

b) pfT (id=21)

c) afficheTab() affiche 20, 21, 22, 23, 24. Les mêmes valeurs que tab.

e) tab (id=21)

f) pfT (id=21)

g) La ligne <3> affiche 110.

i) tab3 (id=17)

j) pfT (id=17)

k) La ligne<4> affiche 6.

m) tab (id=21)

n) pfT (id=21)

p) Les valeurs de tab ont changer. Elles ont été multipliées par 100. C'est normal car dans la fonction changeTab, le tableau à été directement modifié et non une copie.

q) La ligne <6> va afficher 2000, 2100, 2200, 2300, 2400.

6.1) Indiquer dans le CR ce que ce programme affiche.

Ce programme affiche les informations des 3 personnes de l'univers Star Wars qui ont été créer.

6.2.1.2) Indiquer dans le CR ce que ce programme affiche.

Ce programme créer la famille Skywalker et affiche les informations de ces différentes personnes.

6.2.1.5) Expliquer le résultat obtenu en <6> et <7>.

Dans les lignes <6> et <7>, l'affichage est le même parce que l'affectation à la ligne <7> n'affecte pas la variable p2 que la référence qui contient la variable p1.

6.2.1.6) Combien d'objets Personne sont créés entre les lignes <5> et <6> ?

Seulement une personne a été créer entre ces deux lignes.

6.3.2) Indiquer dans le compte rendu ce qu'affiche ce programme.

Ce programme créer une personne et affiche ses informations. Il change ensuite son nom et son prénom et réaffiche ses informations.

6.3.3) relever le numéro d'objet :

de schmi en <2> : schmi (id=20)

de pfPersonne en <3> dans testChangeObjet() : pfPersonne (id=20)

de pfPersonne en <4> dans setNom() : pfPersonne (id=20)

7.1.3)

Exécuter pas à pas ce programme et à **chaque point <2> à <8>**, indiquer dans le compte rendu (CR) :

- les valeurs des cases du tableau tab,
- la valeur finale de l'indice utilisé par le for

Point d'arrêt 2 : tab : [20,21,22,23,24], i = 5.

Point d'arrêt 3 : tab : [50,51,52,53,54], j = 5.

Point d'arrêt 4 : tab : [101,102,103,104,105], k = 6.

Point d'arrêt 5 : tab : [200,201,202,203,204], l = 5.

Point d'arrêt 6 : tab : [400,401,402,403,404], m = 5.

Point d'arrêt 7 : tab : [500,401,502,403,504], n = 6.

Point d'arrêt 8 : tab : [600,401,502,603,504], o = 6.

7.2.2) Dans quel cas (unique) estPresent() renvoie true ?

estPresent renvoie true si la dernière valeur du tableau est la valeur recherchée.

Quel bug se cache dans estPresent() compte tenu de l'objectif fixé plus haut

estPresent() doit chercher si une valeur est présente dans un tableau. Cependant, le programme ne s'arrête pas dès que la valeur a été trouvée et met donc toujours false même si la valeur cherchée a été trouvée. Voici la correction :

```
public static boolean estPresent (int[] pfT, int pfVal) {
    for (int i=0; i<pfT.length; i++) {
        if (pfT[i] == pfVal) {
            return true;
        }
    }
    return false;
}
```

7.3) Après plusieurs exécutions, nous pouvons voir que le sous-programme mystère trie par ordre croissant des tableaux. On peut donc l'appeler : triCroissant().

7.4) Après plusieurs exécutions, nous pouvons nous rendre compte que le sous-programme mystère2 cherche le plus grand diviseur commun entre deux valeurs. On peut l'appeler pgcd().

7.5) Après plusieurs exécutions, nous pouvons nous rendre compte que ce programme fait appel à la récursivité pour calculer le factoriel d'un nombre. On peut donc l'appeler : factoriel().

