

Sprint 3 → tasca 1

Ampliar el programa: tipus de dades i entrada des de teclat.

Enunciat

Sobre el programa fet a la setmana anterior, que implementa el càlcul de les dades sobre unes piscines rectangulars, el client que ens ho havia encarregat ens demana alguns canvis:

1. Vol que les mides d'ample, llarg i profunditat puguin ser valors reals, per exemple, 5.6 o 1.8.
2. Encara que els valors d'amples, llargs i profunditats passin a ser valors reals, es vol que els valors de les àrees i dels volums estiguin guardats sobre variables senceres.
3. Vol que les dades d'ample i profunditat les introdueixi l'usuari. Cal el programa demani per a cada piscina el seu ample i profunditat i els guardi a les variables corresponents.
4. Vol que el valor de llarg, que és el mateix per les dues piscines, sigui una constant, doncs ha decidit que això no vol que canviï. El valor serà sempre de 300,3.

A) Modifiqueu el codi del programa per a complir aquestes tres peticions del client.

Per al **pas 1**, les noves mides d'ample, llarg i profunditat seran les següents:

Piscina rectangular 1

llarg: 300,3
ample: 150,5
profunditat: 20,5

Piscina rectangular 2

llarg: 300,3
ample: 80,5
profunditat: 35,5

Per al pas 2. tingueu en compte els mètodes de conversió de tipus.

Un cop implementeu el **pas 3**, les mides les introduireu des de teclat. Podeu provar amb aquestes mides o amb d'altres.

B) Afegir traces

Les traces són sortides per pantalla que es fan durant el desenvolupament d'un programa per tal de saber quin valors van prenent les variables quan el programa s'executa. Són ajudes per programar que s'han d'eliminar un cop el programa es dona per acabat.

Un exemple d'aplicar traces per un programa.

Si partim d'aquest programa que calcula una divisió i després suma un valor:

```
public class S_traces {  
    public static void main(String[] args) {  
        double dividend = 20.0;  
        double divisor = 6.0;  
        double sumarAlFinal = 3.0;  
        double resultat = 0.0;  
        resultat = dividend/divisor;  
        resultat = resultat + sumarAlFinal;  
        System.out.println(resultat);  
    }  
}
```

Un cop afegides les traces quedaria:

```
public class S_traces {  
    public static void main(String[] args) {  
        double dividend = 20.0;  
        double divisor = 6.0;  
        double sumarAlFinal = 3.0;  
        double resultat = 0.0;  
        //Traça per veure quan val resultat al començament  
        System.out.println("TR: Resultat val inicialment:"+resultat);  
        resultat = dividend/divisor;  
        //Traça per veure quan val resultat després de dividir  
        System.out.println("TR: Resultat despres de divisio:"+resultat);  
        resultat = resultat + sumarAlFinal;  
        //Traça per veure quan val resultat després de la sumafinal  
        System.out.println("TR: Resultat despres de suma:"+resultat);  
        System.out.println(resultat);  
    }  
}
```

Cal que traceu les variables profunditat, per que en fer una execució del programa podem fer-nos una idea de com va variant el valor d'aquesta variable.

C) Ús del debugger

El Debugger és una eina que tenim per anar seguint què fa el programa en temps d'execució. Ens proporciona informació que ens ajuda a resoldre els problemes que van sorgint. És una alternativa a les traces.

Sobre el programa, un cop aquest funcioni, **i un cop eliminades les traces**, feu servir l'eina Debugger per executar-lo pas a pas.

Cal que poseu un parell de punts d'aturada:

1. Abans d'intercanviar els valors de profunditat
2. Després d'intercanviar els valors de profunditat i de calcular el nou volum que encabeix cada piscina

A cadascun d'aquests punts d'aturada haureu de visualitzar el valors de les variables:

1. Visualitzeu els valors de profunditat del cadascuna de les piscines. Visualitzeu els valors de volum que poden encabir aquestes.
2. Visualitzeu els valors de profunditat de cada piscina un cop intercanviats els valors. Visualitzeu també els valors dels volums que poden encabir ara les piscines. Cal que feu una captura de pantalla.

Lliurament

Què

Heu de :

1. Crear un repository a github.
2. Lliurar l'arxiu .java del programa que heu creat
3. Lliurar la captura de pantalla de l'apartat B on apareguin les vostres dades (nom, data de realització) i es visualitzin els valors de les profunditats de les piscines que mostren les traces.
4. Lliurar la captura de pantalla de l'apartat C on apareguin les vostres dades (nom, data de realització) i es visualitzin els valors dels volums que poden encabir les piscines un cop intercanviats els valors.

Cal que aparegui el contingut de les variables que mostra el debugger.

Com

El nom del fitxer .java serà: **Cognom1Inicial del cognom2**. Els cognoms s'escriuran sense accents. Per exemple, l'estudiant *Joan García Santos* posaria el següent nom al seu fitxer de l'exercici: **GarciaS.java**.

El nom del fitxer de la captura de pantalla seguirà la mateixa nomenclatura: **Cognom1Inicial del cognom2**.

El format serà .png o bé .jpg.

MODEL D'INFORME D'AUTOAVALUACIÓ

Autoavaluació

Avalua el teu programa segons els [criteris](#) de la taula d'avaluació i omple aquí quina ha estat la teva puntuació

Robust:

Claredat i ordre:

Variables:

Constants:

Indentacions:

Correcte:

Puntuació final:

Criteris per avaluar el programa

Criteri	2 Notable	1 Suficient	0 Insuficient	puntuació
Robust	El programa s'executa sense problemes	El programa es pot executar, però quan arriba a cert punt falla i finalitza incorrectament.	El programa no s'executa	
Claredat i ordre	És endreçat i s'entén la lògica del programa. Hi ha comentaris explicatius	Alguns fragments són difícils de seguir. Hi ha alguns comentaris però no suficients.	No hi ha un ordre clar. No hi ha comentaris explicatius	
Variables	Els noms de les variables comencen per minúscules. Els noms de les variables són significatius. Les variables es declaren a principi del codi.	Els noms de les variables no comencen per minúscules. Alguna de les variables té un nom que no és significatiu. Hi ha variables declarades enmig del codi.	Les variables tenen noms no significatius. Les variables no es declaren al principi del codi	
Constants	Les variables que tenen un valor que no canvia ni canviarà, es declaren com a constants. Els noms de les constants és tot en majúscules.	Hi ha algun valor que no canvia que no s'ha declarat com a constant, o bé s'ha posat el valor directament al codi. El nom de la variable constant no és correcte.	No es declara cap constant.	
Indentacions	El codi del programa està correctament indentat.	En general està indentat però hi ha algunes errades d'indentació.	No està gens indentat.	
Correcte	Fa tots els 4 punts que es demanen i els apartats B i C	Fa 3 punts del que es demana. O bé no fa algun dels apartats B o C.	Fa menys de 3 punts. No fa ni l'apartat B ni el C.	