

[Área personal](#) / [Mis cursos](#) / [DAM M09B0 Programació de serveis i processos](#) / [UF2. Processos i fils](#) / [Questionari EAC1P2](#)

Comenzado el	jueves, 21 de octubre de 2021, 18:25
Estado	Finalizado
Finalizado en	jueves, 21 de octubre de 2021, 18:38
Tiempo empleado	12 minutos 46 segundos
Calificación	10,00 de 10,00 (100%)

Segons el següent codi, ompliu els buits amb les opcions que teniu

```
synchronized void metodeWait () {  
while (!condicio)  
try {  
    wait();  
}  
catch (InterruptedException e) {  
// codi d'excepció  
}  
// codi que s'executa quan la condició és certa.  
}  
  
synchronized void metodeCanviCondicio(boolean valor) {  
    condicio = valor;  
    notify(); // o notifyAll()  
}
```

- El mètode metodeWait() ha d'executar-se  per tal d'assegurar que en canviar la condició del while i produir-se la notificació, només un fil aconseguirà executar el codi situat a la sortida del bucle.
- El mètode metodeCanviCondicio() modifica el valor de la condició usada en el while del mètode anterior i notifica el canvi per tal que si hi ha algun fil esperant dins del bucle, s'alliberi. El fil desbloquejat tornarà a bloquejar-se en cas que la condició avaluada prengués el valor cert, però aconseguiria sortir del bucle i continuar l'execució en cas que la condició fos falsa. Aquestes dues operacions s'han d'executar sempre de forma atòmica, doncs en cas contrari perdria la seva efectivitat, perquè podria passar que es cridés el mètode  entre l'alliberació del bloqueig i el canvi d'estat a en espera. En aquest cas el mètode notify() no afectaria al fil.
- Un fil que és alliberat a través d'un notify() passarà a l'estat de preparat. És a dir, no vol dir que s'executi immediatament sinó que tornarà a lluitar per agafar el bloqueig amb altres fils en el mateix estat.
- Quan un fil passa pel  es queda en espera immediatament, i no fa res per veure si allò que estava esperant ja ha passat. Això només passa si el fil que executa el mètode  , invoca notify().

La resposta és correcta.

La respuesta correcta es:
Segons el següent codi, ompliu els buits amb les opcions que teniu

```
synchronized void metodeWait () {  
while (!condicio)  
try {  
    wait();  
}  
catch (InterruptedException e) {  
    // codi d'excepció  
}  
// codi que s'executa quan la condició és certa.  
}  
  
synchronized void metodeCanviCondicio(boolean valor) {  
    condicio = valor;  
    notify(); // o notifyAll()  
}
```

- El mètode metodeWait() ha d'executar-se [sincronitzat] per tal d'assegurar que en canviar la condició del while i produir-se la notificació, només un fil aconseguirà executar el codi situat a la sortida del bucle.
- El mètode metodeCanviCondicio() modifica el valor de la condició usada en el while del mètode anterior i notifica el canvi per tal que si hi ha algun fil esperant dins del bucle, s'alliberi. El fil desbloquejat tornarà a bloquejar-se en cas que la condició avaluada prengués el valor cert, però aconseguiria sortir del bucle i continuar l'execució en cas que la condició fos falsa. Aquestes dues operacions s'han d'executar sempre de forma atòmica, doncs en cas contrari perdria la seva efectivitat, perquè podria passar que es cridés el mètode [notify()] entre l'alliberació del bloqueig i el canvi d'estat a en espera. En aquest cas el mètode notify() no afectaria al fil.
- Un fil que és alliberat a través d'un notify() passarà a l'estat de preparat. És a dir, no vol dir que s'executi immediatament sinó que tornarà a lluitar per agafar el bloqueig amb altres fils en el mateix estat.
- Quan un fil passa pel [wait()] es queda en espera immediatament, i no fa res per veure si allò que estava esperant ja ha passat. Això només passa si el fil que executa el mètode [metodeCanviCondicio()], invoca notify().

2

Puntúa 1,00 sobre 1,00

Correcta

Digueu si la següent afirmació és certa o falsa

Una classe no pot tenir mètodes sincronitzats i no sincronitzats.

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

3

Puntúa 1,00 sobre 1,00

Correcta

Digueu si la següent afirmació és certa

ThreadGroup conté tota la funcionalitat per la manipulació de grups de fils. Un grup pot tenir un número de fils limitat, com a màxim (2^8). Habitualment tots el fils d'un grup tenen alguna cosa en comú: qui els va crear, quina funció tenen o quan han d'executar-se o finalitzar.

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La resposta és certa, el número de fils és "quasi" il·limitat.

La respuesta correcta es 'Falso'

4

Puntúa 1,00 sobre 1,00

Correcta

Responen si és vertadera (V) o falsa (F) la següent afirmació:

`wait()`, `notify()` i `notifyAll()` són mètodes que sempre s'executen en blocs de codi sincronitzat

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

5

Puntúa 1,00 sobre 1,00

Correcta

Si un fil executa el mètode `wait()`, perd el bloqueig del monitor en el cas que en tingui.

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

6

Puntúa 1,00 sobre 1,00

Correcta

Digues si la següent afirmació és certa o falsa

El mètode `sleep(milisegons)` deixa el fil en estat de **bloquejat** durant un temps marcat en milisegons i un cop passat aquest temps torna a l'estat de **preparat**. Durant aquest temps si el fil tenia el bloqueig d'un objecte, el perdrà.

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

7

Puntúa 1,00 sobre 1,00

Correcta

Ompliu els buits amb les opcions disponibles

Mentre el fil està en execució, podrà sortir-ne per diversos motius:

- Ha expirat el temps d'ús del processar per part del fil o s'executa el mètode ✓ .
- Inicia una operació d'entrada/sortida i passa a un estat de ✓ fins que l'operació acabi.
- Crida al mètode ✓ d'un altre fil per esperar la finalització del mateix, El fil invocador passa a l'estat de **bloquejat** fins que el fil invocat passi a l'estat *acabat*.
- Intenta executar un codi que està sincronitzat (*synchronized*) i no pot perquè hi ha un altre fil executant-lo, passa a un estat de **bloquejat** fins que el bloc sincronitzat ✓ .
- Es crida el mètode ✓ passant-lo a l'estat *d'espera*. Quan un fil entre en estat d'espera, només en podrà sortir si s'invoca algun dels seus mètodes *notify()* o *notifyAll()*.
- S'invoca el mètode ✓ . El fil passa a estat adormit fins que hagi transcorregut el temps indicat com a paràmetre. L'estat adormit, és molt útil per mantenir un fil aturat durant un temps determinat minimitzant al màxim els recursos utilitzats. Suposem que un programa de tractament de textos, disposa d'un fil que guarda els documents automàticament cada 10 minuts.

La resposta és correcta.

La respuesta correcta es:

Ompliu els buits amb les opcions disponibles

Mentre el fil està en execució, podrà sortir-ne per diversos motius:

- Ha expirat el temps d'ús del processar per part del fil o s'executa el mètode .
- Inicia una operació d'entrada/sortida i passa a un estat de fins que l'operació acabi.
- Crida al mètode d'un altre fil per esperar la finalització del mateix, El fil invocador passa a l'estat de **bloquejat** fins que el fil invocat passi a l'estat *acabat*.
- Intenta executar un codi que està sincronitzat (*synchronized*) i no pot perquè hi ha un altre fil executant-lo, passa a un estat de **bloquejat** fins que el bloc sincronitzat quedi .
- Es crida el mètode passant-lo a l'estat *d'espera*. Quan un fil entre en estat d'espera, només en podrà sortir si s'invoca algun dels seus mètodes *notify()* o *notifyAll()*.
- S'invoca el mètode . El fil passa a estat adormit fins que hagi transcorregut el temps indicat com a paràmetre. L'estat adormit, és molt útil per mantenir un fil aturat durant un temps determinat minimitzant al màxim els recursos utilitzats. Suposem que un programa de tractament de textos, disposa d'un fil que guarda els documents automàticament cada 10 minuts.

8

Puntúa 1,00 sobre 1,00

Correcta

Un exemple clàssic de programació concurrent i multifils és el model

Selecione una:

- ☐ a. d'arquitectura client-servidor
- ☐ b. vista controlador (MVC)
- ☐ c. del patró factory
- ☒ d. productor-consumidor ✓
- ☐ e. Cap de les anteriors és correcta

La resposta és correcta.

La respuesta correcta es: productor-consumidor

9

Puntúa 1,00 sobre 1,00

Correcta

El mètode yield() envia el fil a l'estat de interbloqueig.

Selecione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

10

Puntúa 1,00 sobre 1,00

Correcta

Si un fil executa el mètode sleep(), perd el bloqueig del monitor en el cas que el tingui.

Selecione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

