



UD2: Programació multifil

Pràctica evaluable



Banca Privada d'Abastos

L'IES Abastos continua amb la seua imparable expansió i en els últims mesos ha emés una OPA hostil, aconseguint fer-se amb la majoria d'accions d'un banc estranger del qual es pretén mantindre l'anonimat.

Tot ha sigut gràcies al bon fer dels seus professors però sobretot gràcies al treball dels seus competents i ben valorats en el mercat laboral alumnes, especialment als del mòdul de Programació de serveis i Processos de 2n de DAM, que han aconseguit realitzar intervencions en el programari de gestió de la borsa, creant multitud de processos i fils de compra d'accions i d'aquesta manera anar adquirint cada vegada més accions fins a aconseguir ser l'accionista majoritari.

GRÀCIES I BONA FAENA!

BENVINGUTS A... BANCA PRIVADA D'ABASTOS!





De moment, el departament de disseny de l'IES Abastos està encara treballant en la imatge corporativa del banc, encara que com el nínxol de mercat del mateix són persones joves estudiants, s'ha optat per la imatge que podeu veure més amunt, que projecta una imatge informal, innovadora i trencadora. Però això és una altra història.... nosaltres som excel·lents programadors de fils, així que anem a assegurar que el funcionament del banc siga el correcte i per a això hem de realitzar el següent programa per a gestionar el saldo d'un compte corrent.

ESPECIFICACIONS

Les especificacions del programa són les següents:

- S'han d'implementar 3 fitxers en llenguatge java:
 - ▷ **Compte.java** → Contindrà la classe per a gestionar un compte corrent. Haurà de contenir tots els atributs i mètodes necessaris per a gestionar un compte.
 - ▷ **Persona.java** → Contindrà la classe on es gestiona el que fa una persona amb el compte.
 - ▷ **BPA.java** (Banca Privada d'Abastos) → Serà el programa principal que es llençarà.
- S'ha d'implementar **al menys** tres mètodes que permeten obtindre el saldo actual, fer un ingrés (s'incrementa el saldo), fer un reintegrament (se li resta al saldo).
- S'ha de controlar si hi ha algun error, per exemple, si es fa un reintegrament i no hi ha saldo, o si un ingrés supera el màxim que pot tindre el compte. S'han de mostrar missatges amb els moviments que es realitzen.
- El compte corrent rep en el seu constructor el saldo actual i el valor màxim que pot tindre. Els mètodes d'ingrés i reintegrament han de definir-se com *synchronized*. Així, si una persona per exemple està fent un ingrés, una altra no pot fer un reintegrament al mateix temps.
- `max_saldo` → Seria el màxim saldo que es pot tindre en el compte corrent.
- La classe Persona, en el constructor rebrà el nom de la persona i el compte corrent associat. Després farà 2 ingressos i 2 reintegraments en el compte, alternant-los.
- Per a crear els moviments de diners genera números aleatoris entre 1 i 500, el constructor de la classe ha de portar el nom de la persona. Seria algo així:

```
Random rnd = new Random();
diners = (int) (rnd.nextDouble() * 501.0);
c.Ingres(diners, nom);
```
- Crea en el mètode `main()` un objecte `Compte` amb saldo 40 i màxim 500 compartit per a 2 objectes `Persona` i inicia els fils per realitzar moviments en el compte.

RESULTATS ESPERATS

Quan executem, és fàcil que el programa es quede penjat en algun fil, és a dir, el programa mai acabarà.

Casos exemple (els números poden canviar):

1. Queda en el compte 47 i les dues persones volen retirar 112 i 345. Les dues es queden bloquejades esperant que l'altra ingrese.
2. Queda en el compte 456 i les dues volen ingressar 112 i 345. Les dues es queden bloquejades esperant que l'altra traga diners.
3. Una persona vol fer ingrés de 456 però en tot moment el saldo és molt gran (>44). Es queda esperant fins que l'altra acabe i, donat que el saldo no ha baixat de 44, es queda bloquejada.
4. A l'inrevés amb retirar.
5. Saldo en el compte 250, una vol retirar 300 i l'altra vol ingressar 300. Cap de les dues pot, es bloquegen les dues.
6. Cas en què les dues persones acaben sense estar bloquejades permanentment.



CAPTURES DE PANTALLA

Ací teniu algunes captures de pantalla de alguns casos. Observeu que quan fica "ENTER/ENTER/Qualsevol cosa", eixa cosa l'he escrit jo des de'l teclat. Heu d'escriure el motiu pel qual s'ha bloquejat:

```
[ramon@BPA]$ java BPA
Es va a ingressar saldo (el saldo actual és: 40)
Anna ingressa => : 337. Saldo actual: 377
Es van a retirar diners (el saldo actual és: 377)
Anna retira => : 13. Saldo actual: 364
Es va a ingressar saldo (el saldo actual és: 364)
Anna ingressa => : 130. Saldo actual: 494
Es van a retirar diners (el saldo actual és: 494)
Anna retira => : 475. Saldo actual: 19
Es va a ingressar saldo (el saldo actual és: 19)
Joan ingressa => : 134. Saldo actual: 153
Es van a retirar diners (el saldo actual és: 153)
Joan retira => : 13. Saldo actual: 140
Es va a ingressar saldo (el saldo actual és: 140)
Joan ingressa => : 352. Saldo actual: 492
Es van a retirar diners (el saldo actual és: 492)
Joan retira => : 303. Saldo actual: 189
[ramon@BPA]$ OK!!!

[ramon@BPA]$ java BPA
Es va a ingressar saldo (el saldo actual és: 40)
Anna ingressa => : 188. Saldo actual: 228
Joan vol ingressar 456 diners. Massa saldo. Saldo actual: 228
Es van a retirar diners (el saldo actual és: 228)
Anna retira => : 82. Saldo actual: 146
Anna vol ingressar 433 diners. Massa saldo. Saldo actual: 146

BLOQUEJAT!!! PEL MOTIU ....!!!!
[ramon@BPA]$ java BPA
Es va a ingressar saldo (el saldo actual és: 40)
Anna ingressa => : 131. Saldo actual: 171
Es van a retirar diners (el saldo actual és: 171)
Anna retira => : 114. Saldo actual: 57
Es va a ingressar saldo (el saldo actual és: 57)
Anna ingressa => : 223. Saldo actual: 280
Anna vol retirar 296 diners però no hi ha prou saldo. Saldo actual: 280
Joan vol ingressar 307 diners. Massa saldo. Saldo actual: 280

BLOQUEJAT!!! PEL MOTIU ....!!!!
[ramon@BPA]$ java BPA
Es va a ingressar saldo (el saldo actual és: 40)
Anna ingressa => : 45. Saldo actual: 85
Anna vol retirar 126 diners però no hi ha prou saldo. Saldo actual: 85
Es va a ingressar saldo (el saldo actual és: 85)
Joan ingressa => : 273. Saldo actual: 358
Es van a retirar diners (el saldo actual és: 358)
Joan retira => : 210. Saldo actual: 148
Es va a ingressar saldo (el saldo actual és: 148)
Joan ingressa => : 187. Saldo actual: 335
Es van a retirar diners (el saldo actual és: 335)
Joan retira => : 176. Saldo actual: 159
Es van a retirar diners (el saldo actual és: 159)
Anna retira => : 126. Saldo actual: 33
Es va a ingressar saldo (el saldo actual és: 33)
Anna ingressa => : 94. Saldo actual: 127
Anna vol retirar 499 diners però no hi ha prou saldo. Saldo actual: 127

BLOQUEJAT!!! PEL MOTIU ....!!!!
```

**ENTREGA****Rúbrica d'avaluació**

Fita	Puntuació
Els casos d'exemple estan correctament identificats	1,5
Utilitza correctament wait/notify/notifyAll	3
Els blocs sincronitzats estan correctament definits	3
Estructura bé les classes (atributs, mètodes)	2
Codi ben estructurat i comentat	0,5

**Per a la correcció de la pràctica, el programa ha de compilar i no donar errades d'execució.
Si hi ha errades de compilació o execució la nota serà de 0.**

Per al lliurament d'aquesta pràctica s'ha de pujar a Aules NOMÉS els 3 fitxers .java comprimits en un zip que tindrà com a nom **UD2-NomCognom1Cognom2-BancaAbastos.zip**

En el cas de que hi hagen més de 3 fitxers la nota serà de 0.

A més, puja almenys 3 captures de pantalla que indiquen 3 execucions que s'adeqüen als casos anteriors (els valors no han de ser exactament iguals).

NO S'ACCEPTARÀ EL LLIURAMENT EN QUALESVOL ALTRE FORMAT.

**Moltíssimes gràcies pel vostre treball.
De segur que en poc de temps tots ens farem rics!**

