Test de la classe Dispositiu — X59030_ca

Això és el test de la classe `Dispositiu`, que correspon a la pràctica **Simulació d'un SO**. Aquesta classe implementa una dispositiu d'entrada i sortida que actuarà com una **cua**, és a dir, el primer a entrar serà el primer a sortir. A més, tindrà un identificador i una capacitat màxima, que es definiran quan instanciem un objecte `Dispositiu`. Sobre aquest objecte podrem aplicar 3 operacions:

1. `D << x`, on `x` és in `int` i `D` és un `Dispositiu`. Aquesta operació afegeix `x` al dispositiu si el dispositiu no està ple. Si està ple, no fa res (mireu el fitxer `program.cpp`).
2. `D >> x`, on `x` és in `int` i `D` és un `Dispositiu`. Aquesta operació treu un element del dispositiu i el posa a la variable `x` si el dispositiu no està buit. Si està buit, no fa res (de fet, no hauríem de permetre aquesta operació, mireu el fitxer `program.cpp`).
3. `D.buit()`. Avalua a `true` si i només si el dispositiu `D` està buit.

Us passem el programa principal, que cridarà a una instància de la classe, i que cridarà els seus mètodes. Cal que:

1. Completeu l'especificació de la classe al fitxer `dispositiu.hpp`.
2. Implementeu la classe `Dispositiu` al fitxer `dispositiu.cpp`.

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile, program.cpp`, apart de les versions `*.old` dels altres fitxers que heu d'acabar d'implementar.
Quan pugeu la vostra solució al jutge, només cal que pugeu un tar construït així:

```
tar cf solution.tar dispositiu.cpp dispositiu.hpp
```

L'entrada és un nombre indeterminat de línies que poden ser:

- `PUSH val`. Afegeix el valor `val` al dispositiu, si no està ple. Si ho està, treu `DISPOSITIU PLE`.
- `POP`. Treu un valor del dispositiu. Si està buit, treu `DISPOSITIU BUIT`.
- `WRITE`. Escriu tots els valors del dispositiu.
- `EMPTY`. Escriu `cert` (`fals`) si el dispositiu està buit (no està buit).

Exemple d'entrada 1

```
EMPTY
PUSH 10
PUSH 20
PUSH 30
WRITE
EMPTY
POP
POP
POP
EMPTY
PUSH 50
PUSH 60
WRITE
EMPTY
POP
POP
WRITE
EMPTY
```

```
PUSH -1000
WRITE
```

Exemple de sortida 1

```
cert
DISPOSITIU_ID: 0 : 10 20 30
fals
10
20
30
cert
DISPOSITIU_ID: 0 : 50 60
fals
50
60
DISPOSITIU_ID: 0 :
cert
DISPOSITIU_ID: 0 : -1000
```

Exemple d'entrada 2

```
EMPTY
PUSH 10
PUSH 20
PUSH 30
PUSH 40
PUSH 10
PUSH 20
PUSH 30
PUSH 40
WRITE
EMPTY
POP
POP
POP
EMPTY
POP
PUSH 50
PUSH 60
WRITE
EMPTY
POP
POP
POP
POP
POP
WRITE
EMPTY
PUSH -1000
WRITE
```

Exemple de sortida 2

```
cert
DISPOSITIU PLE
DISPOSITIU PLE
DISPOSITIU PLE
DISPOSITIU_ID: 0 : 10 20 30 40 10
fals
10
20
30
fals
40
DISPOSITIU_ID: 0 : 10 50 60
fals
10
50
60
DISPOSITIU BUIT
DISPOSITIU BUIT
DISPOSITIU_ID: 0 :
cert
DISPOSITIU_ID: 0 : -1000
```

Exemple d'entrada 3

```
EMPTY
PUSH 1
PUSH 2
PUSH 3
PUSH 4
PUSH 5
PUSH 6
PUSH 7
PUSH 8
WRITE
```

```
EMPTY
POP
POP
POP
POP
POP
POP
POP
POP
EMPTY
WRITE
```

```
PUSH -1000
WRITE
```

Exemple de sortida 3

```
cert
DISPOSITIU PLE
DISPOSITIU PLE
DISPOSITIU PLE
DISPOSITIU_ID: 0 : 1 2 3 4 5
fals
1
2
3
4
5
DISPOSITIU BUIT
DISPOSITIU BUIT
DISPOSITIU BUIT
DISPOSITIU BUIT
cert
DISPOSITIU_ID: 0 :
DISPOSITIU_ID: 0 : -1000
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:17:52.591Z