

IIP Primer Parcial - ETSInf
10 de Novembre de 2014. Durada: 1 hora i 30 minuts.

Es desitja fer una aplicació per a representar un joc amb blocs de diferents colors i dimensions, apilables en torres. Cadascun d'aquests blocs té associat un **color** (blau o roig) i una **dimensió** (nombre enter entre 1 i 50, tots dos inclosos).

Les regles d'aquest joc indiquen que:

- Els blocs apilats en una torre han de seguir colors alterns (damunt d'un bloc blau solament pot haver-hi un bloc roig i viceversa).
- Damunt d'un bloc de dimensió x solament pot haver-hi un bloc de dimensió y , on $y \leq x$ (la torre s'estreny cap a la punta, és a dir, s'eixampla cap a la base).
- Un bloc pot ser un **comodí**, en aquest cas pot anar damunt de qualsevol altre bloc, independentment dels seus colors. Ara bé, un bloc comodí, com qualsevol altre, ha de respectar la regla de la dimensió.

1. 6 punts Es demana implementar la classe **Bloc**, per al que s'ha de:
 - a) (0.5 punts) Definir els atributs de classe públics i constants que representen els dos colors possibles: **BLAU** i **ROIG**, amb valors enters 0 i 1, respectivament. S'han d'utilitzar sempre que es requereixca (tant en la classe **Bloc** com en la classe **TorreBlocs**).
 - b) (0.5 punts) Definir els atributs d'instància privats **color** (**int**), **dimensio** (**int**), **comodi** (**boolean**).
 - c) (1.5 punts) Implementar dos constructors:
 - Un constructor general amb els paràmetres apropiats per a inicialitzar tots els atributs d'instància.
 - Un constructor per defecte que cree un **Bloc** blau que no siga un comodí i la dimensió del qual es determine aleatòriament dins del rang [1,50].
 - d) (0.5 punts) Escriure el mètode consultor i el mètode modificador de l'atribut **dimensio**.
 - e) (1 punt) Escriure el mètode **equals** (que sobreescriu el d'**Object**) per a comprovar si un bloc és igual a un altre, açò és, si l'altre és un bloc i els atributs d'un i l'altre coincideixen un a un.
 - f) (1 punt) Escriure el mètode **toString** (que sobreescriu el d'**Object**) perquè el resultat tinga un format com el mostrat en els següents exemples: "(Color: roig, dimensió: 22 i SÍ és comodí)",
"(Color: blau, dimensió: 15 i NO és comodí)".
 - g) (1 punt) Escriure un mètode **potEstarDamuntDe(Bloc b)** que comprove si un bloc, aplicant les regles ja esmentades, pot estar o no damunt d'un altre **Bloc b** que reb com a argument. Per exemple, donades dues variables **a** i **b** de tipus **Bloc**, **a.potEstarDamuntDe(b)** serà cert si i solament si la dimensió del bloc **a** és menor o igual que la del bloc **b** i, bé **a** és un comodí, o bé els colors de **a** i **b** són diferents.
2. 4 punts Es demana implementar la classe programa **TorreBlocs**, l'objectiu de la qual és poder fer proves amb torres menudes d'alguns blocs. Per a açò, cal implementar aquesta classe amb un mètode **dimensioValida** que:
 - a) (1.25 punts) limite el valor de la dimensió **d** que se li passa com a paràmetre a un dels valors de l'interval [1,50]. Açò significa que si **d** és menor que 1, torna 1; si **d** és major que 50, torna 50; i, en qualsevol altre cas, torna **d** sense modificar-lo.i amb un mètode **main** que realitze les següents accions:
 - b) (0.25 punts) Crear un **Bloc b1** amb el constructor per defecte.
 - c) (0.5 punts) Crear un **Bloc b2** de color blau, dimensió 30 i que siga un comodí (amb el constructor general).
 - d) (1 punt) Després de llegir el seu color i dimensió des de teclat, crear un **Bloc b3** que no siga comodí.
El color se sol·licitarà a l'usuari com un **String**, amb valors "roig" o "blau"; si l'usuari introdueix qualsevol altre valor, el color del **Bloc** serà roig.
El valor de dimensió se sol·licitarà a l'usuari com un enter en el rang [1,50] i, per tal de garantir que l'usuari no ha introduït qualsevol altre valor, s'haurà d'invocar al mètode **dimensioValida**.
 - e) (0.25 punts) Mostrar per pantalla els tres objectes creats.
 - f) (0.75 punts) Comprovar si es pot formar una torre situant el bloc **b3** damunt del **b2** i aquest últim damunt del bloc **b1**. A continuació, mostrar per pantalla un missatge amb el resultat d'aquesta comprovació, açò és, si s'ha pogut formar la torre o no.

S'han d'importar les classes que es consideren necessàries i utilitzar les constants definides en la classe **Bloc** on siga oportú.

Solució:

```
public class Bloc {
    private int color;
    private int dimensio;
    private boolean comodi;

    public static final int BLAU = 0;
    public static final int ROIG = 1;

    public Bloc() {
        this.color = BLAU;
        this.dimensio = (int) (1 + Math.random() * 50);
        this.comodi = false;
    }

    public Bloc(int color, int dimensio, boolean comodi) {
        this.color = color;
        this.dimensio = dimensio;
        this.comodi = comodi;
    }

    public int getDimensio() {
        return dimensio;
    }

    public void setDimensio(int dim) {
        dimensio = dim;
    }

    public boolean potEstarDamuntDe(Bloc b) {
        return this.dimensio <= b.dimensio
            && (this.comodi || this.color != b.color);
    }

    public boolean equals(Object o) {
        return o instanceof Bloc
            && this.color == ((Bloc) o).color
            && this.dimensio == ((Bloc) o).dimensio
            && this.comodi == ((Bloc) o).comodi;
    }

    public String toString() {
        String col = "blau";
        if (this.color == ROIG) col = "roig";
        String com = "NO";
        if (this.comodi) com = "Sí";
        return "(Color: " + col + ", dimensió: " + dimensio + ", "
            + com + " és comodi)";
    }
}
```

Solució:

```
import java.util.Scanner;
public class TorreBlocs {

    public static int dimensioValida(int dimensio) {
        if (dimensio < 1) return 1;
        else if (dimensio > 50) return 50;
        else return dimensio;
    }

    public static void main(String[] args) {
        Scanner teclat = new Scanner(System.in);

        Bloc b1 = new Bloc();

        Bloc b2 = new Bloc(Bloc.BLAU, 30, true);

        System.out.print("Introdueix el color del bloc 3 (blau/roig): ");
        String color = teclat.next().trim().toLowerCase();
        int codColor = Bloc.ROIG;
        if (color.equals("blau")) codColor = Bloc.BLAU;

        System.out.print("Introdueix la dimensió del bloc 3 [1,50]: ");
        int dimensio = dimensioValida(teclat.nextInt());

        Bloc b3 = new Bloc(codColor, dimensio, false);

        System.out.println("Bloc 1: " + b1 + "\nBloc 2: " + b2 + "\nBloc 3: " + b3);

        boolean res = b3.potEstarDamuntDe(b2) && b2.potEstarDamuntDe(b1);
        System.out.print("\nLa torre formada amb aquests tres blocs");
        if (!res) System.out.print(" NO");
        System.out.println(" és vàlida");
    }
}
```