

INFORMÁTICA APLICADA A LA INGENIERÍA CIVIL SOLUCIÓN DEL EXAMEN FINAL DE JAVA 16 de Diciembre de 2006

Se va a proceder a generar un programa para la gestión de las quinielas de fútbol. Cada una de las quinielas tiene 14 partidos mas el pleno al quince. Para los primeros 14 partidos hay un total de 8 pronósticos como máximo (no es necesario completar todos), y dentro de cada pronóstico el resultado del partido puede ser '1', 'X' y '2', tal y como se muestra en la Figura 1. Se puede seleccionar para cada partido un único resultado (apuesta simple), dos resultados (apuesta doble) o tres resultados (apuesta triple).

LOTERIAS Y APUESTAS DEL ESTADO SENCILLO-MULTIPLE 261

La Quiniela

PRONOSTICOS

1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2
2	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2
3	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
4	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
5	1	X	X	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
6	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
7	X	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
8	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
9	1	X	X	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
10	X	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
11	1	X	X	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
12	1	X	X	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
13	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X
14	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X	2	1	X

PLENO AL 15 **BADAJOS-REC. HUELVA** ... X X 2 ← MARQUE UN SOLO SIGNO

COMBINACIONES

1	1
2	2
3	3
X	4
5	5
6	6
7	7
8	8
9	9
10	1
11	TRIPLES
12	
13	
14	1
15	VALLES

Figura 1: Representación de la quiniela, con sus 14 partidos, 8 pronósticos y su pleno al quince.

Para la elaboración del programa se comienza creando una clase denominada **Quiniela** que tiene las características siguientes:

1. La clase pertenece al paquete **quiniela**. (2 puntos)

`package quiniela;`

2. Las variables de instancia que caracterizan un objeto de la clase **Quiniela** son el número de partidos sin contar el pleno al quince, y el número máximo de pronósticos, las variables se llamarán **partidos** y **pronosticos**, respectivamente. Tendrán un valor fijo que no se puede modificar y que deberá fijarse en el momento de la declaración de las variables, y serán comunes para todos los elementos de la clase **Quiniela**. (4 puntos)

NOTA: Para la realización del examen puede ser especialmente útil la siguiente expresión

$$(k - 1) * 3 + j,$$

que permite conocer la columna de la matriz almacenada en la variable **apuestas** suponiendo que se está en el bloque número k y que j toma valores 1, 2, o 3, correspondientes, respectivamente, con los valores '1', 'X' y '2'. Así por ejemplo, el resultado asociado a la 'X' del pronóstico 3 está en la columna $(3-1)*3+2 = 8$ (octava columna).

```
int numeropronosticos;
int apuestas [][];
int aciertos [];
int pleno [] = new int[3];
```

4. Se genera un constructor para permitir la inicialización de un objeto de esta clase sin dar ningún tipo de argumento mediante la sentencia **super()**. (2 puntos)

```
public Quiniela() {
    super();
}
```

5. El segundo constructor recibe un número entero que se corresponde con el número de pronósticos que se quieren rellenar, y se encarga de: (10 puntos)

- Reservar espacio en memoria para la variable **apuestas**.
- En segundo lugar se inicializarán los valores de las apuestas para el número de pronósticos dado como argumento, de forma que se seleccionará para cada partido una única apuesta (apuesta simple). Para escoger el resultado se emplea la expresión `(int) Math.ceil(3*Math.random())` que devuelve un número aleatorio entre 1 y 3, de forma que si sale 1 significa que se ha seleccionado el '1' y ha de ponerse un uno en la primera columna correspondiente al pronóstico, si sale 2 significa que se ha seleccionado la 'X' y ha de ponerse un uno en la segunda columna correspondiente al pronóstico, y si sale 3 significa que se ha seleccionado el '2' y ha de ponerse un uno en la tercera columna correspondiente al pronóstico. Para los resultados no elegidos su valor correspondiente es cero.
- Por último se selecciona el resultado del pleno al quince empleando la misma expresión `(int) Math.ceil(3*Math.random())`, poniendo un 1 en la posición indicada por ese valor y cero en las otras dos casillas.

```
public Quiniela(int numeropronosticos) {
    int posicion;
    this.numeropronosticos = numeropronosticos;
    apuestas = new int [partidos][3*pronosticos];
    for(int j=0;j<numeropronosticos;j++){
        for(int i=0;i<partidos;i++){
            for(int k=1;k<=3;k++){
```

```

        posicion = (int) Math.ceil(3*Math.random());
        if(posicion==k){
            apuestas[i][3*j+k-1] = 1;
        }else{
            apuestas[i][3*j+k-1] = 0;
        }
    }
}
}
}
for(int k=1;k<=3;k++){
    posicion = (int) Math.ceil(3*Math.random());
    if(posicion==k){
        pleno[k-1] = 1;
    }else{
        pleno[k-1] = 0;
    }
}
}
}

```

6. El tercer constructor recibe como argumentos una lista de listas con las apuestas, y una lista de tres elementos con la información del pleno al quince. (10 puntos)

- Reservar espacio en memoria para la variable **apuestas** de la clase.
- Comprobar si el número de filas de la variable con la información de las apuestas concuerda con el número de partidos de la clase **Quiniela**. En caso de que sea correcto se continua la ejecución, si no lo es se escribe un mensaje en pantalla que diga:

El número de partidos es incorrecto.

- Si la dimensión es correcta se procederá a asignar valores a la variable **apuestas** pero sólo con los datos disponibles.
- Asignar valores a la variable **pleno** de la clase **Quiniela**.

```

public Quiniela(int apuestas [][] , int pleno []) {
    this.apuestas = new int [partidos][3*pronosticos];
    int npar, npro;
    npar = apuestas.length;
    npro = (int) apuestas[0].length / 3;
    numeropronosticos = npro;
    if (npar != partidos) {
        System.out.println("El número de partidos es incorrecto.");
    }
    else {
        for (int j = 0; j < npro; j++) {
            for (int i = 0; i < npar; i++) {
                this.apuestas[i][3 * j] = apuestas[i][3 * j];
                this.apuestas[i][3 * j + 1] = apuestas[i][3 * j + 1];
                this.apuestas[i][3 * j + 2] = apuestas[i][3 * j + 2];
            }
        }
    }
}

```

```

    }
    this.pleno = pleno;
  }
}

```

7. La clase contiene un método que se llama **aciertospronosticos** que recibe como argumento una lista con 14 cadenas de caracteres en los que está la solución de la quiniela '1', 'X' o '2' para cada uno de los partidos. Con esa información se recorrerán todos los pronósticos y se contarán el número de aciertos, almacenándolos en la variable **aciertos**. (8 puntos)

```

public void aciertospronosticos (String resultados[]){
    aciertos=new int[pronosticos];
    for(int j=0;j<numeropronosticos;j++){
        aciertos[j] =0;
    }
    for(int i=0;i<partidos;i++){
        for(int j=0;j<numeropronosticos;j++){
            if(resultados[i]=="1" && apuestas[i][3*j]==1){
                aciertos[j] += 1;
            } else if (resultados[i]=="X" && apuestas[i][3*j+1]==1){
                aciertos[j] += 1;
            } else if (resultados[i]=="2" && apuestas[i][3*j+2]==1){
                aciertos[j] += 1;
            }
        }
    }
}

```

8. Se creará otro método llamado **aciertopleno** que recibe como argumento un string con el resultado '1', 'X' o '2' correcto del pleno al quince, que devuelve "verdadero" si se ha acertado, y "falso" en caso contrario. (8 puntos)

```

public boolean aciertopleno(String plenos){
    boolean plenotrue = false;
    if(plenos=="1" && pleno[0]==1){
        plenotrue = true;
    } else if (plenos=="X" && pleno[1]==1){
        plenotrue = true;
    } else if (plenos=="2" && pleno[2]==1){
        plenotrue = true;
    }
    return(plenotrue);
}

```

9. Otro de los métodos calcula el número máximo de aciertos almacenado en la variable **aciertos** para los pronósticos en los que hay apuestas. (8 puntos)

```

public int maxacierto (){
    int maximo = aciertos[0];
    for(int j=1;j<numeropronosticos;j++){
        if(aciertos[j]>maximo){
            maximo = aciertos[j];
        }
    }
    return(maximo);
}

```

10. El siguiente de los métodos llamado **numsimples** calcula el número de apuestas sencillas de la quiniela. (5 puntos)

```

public int numsimples (){
    int contador = 0;
    int aux;
    for(int i=0;i<partidos;i++){
        for(int j=0;j<numeropronosticos;j++){
            aux = apuestas[i][3*j]+apuestas[i][3*j+1]+apuestas[i][3*j+2];
            if(aux==1){
                contador++;
            }
        }
    }
    return(contador);
}

```

11. El método **numdobles** calcula el número de apuestas dobles de la quiniela. (5 puntos)

```

public int numdobles (){
    int contador = 0;
    int aux;
    for(int i=0;i<partidos;i++){
        for(int j=0;j<numeropronosticos;j++){
            aux = apuestas[i][3*j]+apuestas[i][3*j+1]+apuestas[i][3*j+2];
            if(aux==2){
                contador++;
            }
        }
    }
    return(contador);
}

```

12. El método **numtriples** calcula el número de apuestas triples de la quiniela. (5 puntos)

```

public int numtriples (){
    int contador = 0;
    int aux;
    for(int i=0;i<partidos;i++){
        for(int j=0;j<numeropronosticos;j++){
            aux = apuestas[i][3*j]+apuestas[i][3*j+1]+apuestas[i][3*j+2];
            if(aux==3){
                contador++;
            }
        }
    }
    return(contador);
}

```

13. Por último, el método precio calcula el precio de la quiniela teniendo en cuenta que el precio de la apuesta simple es de 0,5 euros, el de la doble 2 euros, y 5 euros el de la triple. Emplear los métodos creados anteriormente. (5 puntos)

```

public double precio () {
    double coste;
    coste = this.numsimples()*0.5+this.numdobles()*2+this.numsimples()*5;
    return(coste);
}

} // Fin la clase Sudoku

```

A continuación se genera una segunda clase llamada **Apuesta** que contiene el método **main** y que pertenece al mismo paquete **quiniela**:

1. Crear un objeto quiniela con 5 pronósticos en los que los resultados sean aleatorios y almacenarlo en una variable con tu mismo nombre. (4 puntos)

```

Quiniela roberto = new Quiniela(5);

```

2. Escribir el precio de la quiniela creada en una única línea. (4 puntos)

```

System.out.println("El precio de la Quiniela es de "+roberto.precio()+" euros.");

```

3. Almacenar en una lista de **Strings** 14 resultados cualesquiera para suponer que son los valores correctos de los partidos de la quiniela. (4 puntos)

```

String resultado []= {"X","1","1","2","X","2","1","X","1","1","1","1","2","X","X"};

```

4. Calcular los aciertos de cada uno de los pronósticos:(4 puntos)

```

roberto.aciertospronosticos(resultado);

```

5. Escribir en pantalla el número máximo de aciertos. (4 puntos)

```
System.out.println("El número máximo de aciertos es de "+roberto.maxacierto());
```

6. Escribir en pantalla si se ha acertado el pleno o no suponiendo que el valor correcto del partido asociado al pleno es '1'. (4 puntos)

```
if(roberto.aciertopleno("1")){  
    System.out.println("Ha acertado el pleno.");  
} else {  
    System.out.println("No ha acertado el pleno.");  
}  
  
}  
  
}
```

Tiempo: 2h.