

ENGINYERIA ELECTRÒNICA INDUSTRIAL I AUTOMÀTICA

20302 - Programació - Informàtica I

Curs 2011-2012

PRÀCTICA FINAL – EL JOC DEL PENJAT

Víctor Alonso Garrigós

41537802-D

Dimarts, 31 de gener de 2012

ÍNDEX

Índex	Pàgina 1
Enunciat de la pràctica	Pàgina 2
Disseny descendent	Pàgina 3
Algorismes	Pàgines 4, 5, 6 i 7
Joc de proves	Pàgines 8, 9 i 10
Conclusió	Pàgina 11

- **Descripció de l'enunciat de la pràctica**

Introducció

Es tracta d'implementar el clàssic joc del penjat de la següent manera:

- Generem un nombre a l'atzar i amb aquest nombre triem una paraula de entre les predefinides. Aquesta és la paraula secreta que haurà d'endevinar l'usuari.
- Repetim:
 - L'usuari introdueix una lletra per teclat
 - Si aquesta lletra pertany a la paraula secreta, aleshores mostrem en quines posicions es troba.
 - Si no hi pertany aleshores li queda un intent menys i li mostrem el dibuix corresponent del penjat.
- Fins que s'esgotin els intents o encerti la paraula.

Coneixements:

Programació orientada a objectes, esquemes de tractament seqüencial i disseny descendent, tipus de dades i operacions definides pel programador, atributs, paràmetres i variables locals, tractament i propietats de taules, cadenes i caràcters (classe String i Character, i les seves propietats i mètodes), E/S estàndard, fitxers, maneig d'excepcions.

Indicacions:

La pràctica ha d'incloure:

- La definició d'una classe Penjat amb el seu constructor que contindrà la paraula secreta i la plantilla amb els guionets, i un mètode d'interfície: Jugar().
- El mètode Jugar() pot fer ús d'altres mètodes com, per exemple:
 - o *LlegirLletra()* que ha de llegir la lletra introduïda des de teclat per l'usuari (eliminant el caràcter Enter), comprovar que es tracta d'una lletra, i convertir-la a minúscula (consulteu els mètodes de la classe Character).
 - o *LletraEnParaula()* que comprova si la lletra de l'usuari pertany a la paraula secreta, modificant en tal cas la plantilla, i incrementant el nombre d'errors en cas contrari.
 - o *MostrarPlantilla()* que mostra les lletres encertades col·locades al lloc corresponent i els guionets restants.
 - o *ObtenirDibuixError()* que indica quina part del penjat s'ha de dibuixar segons el nombre d'errors.
 - o *Dibuixar()* que dibuixa l'estat del penjat.
- L'ús d'un fitxer de tet que serà el nostre diccionari de paraules. Contindrà 20 paraules en minúscules, una per línia. La primera versió del projecte, pot utilitzar un array d'Strings amb els valor assignats per codi, més endavant aquests valors hauran de ser llegits del fitxer de text.

- **Disseny descendent i algorismes**

1. **Disseny descendent aplicat per arribar a la solució del problema amb les estructures de dades.**

Una vegada plantejada la pràctica vaig veure que era molta cosa per fer-ho en un sol arxiu; així que s'havia de dividir en tres classes diferents: la principal, la classe penjat i el diccionari, com demanava la professora.

La classe del diccionari (*DiccionariAmbFitxer.java*), el que havia de fer era generar una paraula aleatòria. Per això, vaig separar el programa en diferents mètodes. Primer de tot havia de fer que es generàs una paraula aleatòria d'entre una llista de 20 d'un arxiu de text anomenat *diccionari.txt*. Com que això era massa difícil de fer vaig ficar primer de tot les 20 paraules a una taula de 20 posicions, i després vaig crear un mètode que retornàs un sencer aleatori, i aquest seria la posició de la taula on es trobava la paraula; aquest mètode retornava la paraula com a String.

Després havia de fer una classe penjat (*Penjat.java*) on hi hagués el constructor per crear un objecte de tipus Penjat i que al constructor es fes servir la paraula aleatòria i generés la plantilla de guions. Aquesta classe consta de un mètode d'interfície anomenat jugar(), i aquest fa servir tots els altres mètodes. El funcionament del programa és senzill: el programa demana a l'usuari una lletra i la llegeix, comprovant abans si es tracta d'una lletra i convertint-la a minúscula (*llegirLletra()*), comprova si aquesta pertany a la paraula secreta mitjançant un recorregut (*lletraEnParaula()*), i en cas de que hi sigui, retorna la plantilla amb la lletra encertada (*mostrarPlantilla()*) i augmenta el contador de lletres encertades; si no hi és, disminueix el contador d'errors que pots cometre i mostra una part del dibuix (*dibuixar()*).

El programa acaba quan has comés 7 errors i per tant has obtingut el dibuix amb l'home mort o quan l'usuari encerta la paraula (el contador és igual a la quantitat de lletres que té la paraula).

A la classe principal (*Principal.java*) només es creen els objectes de tipus *DiccionariAmbFitxer* i *Penjat* necessaris per jugar i es llança el mètode *jugar()*.

He optat per anar disminuint el nombre d'errors en lloc d'augmentar-los perquè al principi del programa s'informa a l'usuari de la quantitat d'errors que pot cometre, i donat que aquest comença amb 7, he preferit anar restant unitats a aquest valor. El programa es capaç de detectar si el que s'ha introduït és una lletra o no, i si s'introdueixen més d'una lletra, només accepta la primera, i ignora la resta.

2. Algorismes associats.

CLASSE PRINCIPAL.JAVA

```
public class Principal {  
  
    declaracions  
    static final DiccionariAmbFitxer p = new DiccionariAmbFitxer();  
    static Penjat a = new Penjat(p.obtenirParaula());  
  
    public static void main(String[] args) {  
        imprimeix("JOC DEL PENJAT");  
        imprimeix salt de línia;  
        a.jugar();  
    }  
}
```

CLASSE PENJAT.JAVA

```
public class Penjat {  
  
    declaracions  
    public static enter errors = 7;  
    public static enter contador = 0;  
    public static character lletra = ' ';  
    public static String objectiu;  
    public static characters[] guions;  
    public static final String BLANC = " ";  
  
    public Penjat(String str) {  
        objectiu = str;  
        guions = new char[objectiu.length()];  
        per(int idx = 0; idx < objectiu.length(); idx++) {  
            guions[idx] = '_';  
        }  
    }  
  
    public void jugar() {  
        imprimeix("El màxim d'errors permesos és: " + errors);  
        imprimeix salt de línia;  
        mostrarPlantilla(guions);  
        fes {  
            lletra = llegirLletra();  
            imprimeix salt de línia;  
            si (!lletraEnParaula(lletra)) {  
                errors--;  
                dibuixar(obtenirDibuixError(errors));  
            }  
            mostrarPlantilla(guions);  
        } mentre ((errors > 0) i (contador != objectiu.length()));  
        si(contador == objectiu.length()) {  
            imprimeix("Enhorabona!!!");  
            imprimeix("La paraula secreta era: " + objectiu);  
        } sino si (errors == 0) {  
            imprimeix("Has perdut!");  
            imprimeix("La paraula secreta era: " + objectiu);  
        }  
    }  
  
    private static char llegirLletra() {  
        intenta {  
            fes {  
                imprimeix("Introdueix una lletra: ");  
                lletra = (char) llegir;  
                botarLletres();  
            }  
        }  
    }  
}
```

```

        si (comprova si el llegir es una lletra) {
            lletra = transforma a minúscula;
        } sinó si (comprova si lletra es una número) {
            imprimeix("Això és un número!");
        } sinó {
            imprimeix("Això no és una lletra!");
        }
    } mentre (!Character.isLetter(lletra));
} atrapa (excepció) {
}
retorna lletra;
}

private boolean lletraEnParaula(char c) {
    enter i;
    booleà trobada = fals;
    fes {
        per (i = 0; i < objectiu.length(); i++) {
            si ((c == (objectiu.charAt(i))) i (c != guions[i])) {
                guions[i] = objectiu.charAt(i);
                contador++;
                trobada = true;
            }
        }
    } mentre (i < objectiu.length());
    retorna trobada;
}

private void mostrarPlantilla(caracters[] guionets) {
    imprimeix ("Paraula: ");
    per (enter j = 0; j < objectiu.length(); j++) {
        imprimeix (guionets[j] + BLANC);
    }
    imprimeix dos salts de línia;
}

private static String obtenirDibuixError(sencer error) {
    String dibuix = "";
    switch (error) {
        case 6:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   \n" + "|"|\n" + "|"|\n" + "|"|\n" + "+-----\n");
            break;
        case 5:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   \n" + "|"|\n" + "|"|\n" + "+-----\n");
            break;
        case 4:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   |\n" + "|"|\n" + "|"|\n" + "+-----\n");
            break;
        case 3:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   /|\n" + "|"|\n" + "|"|\n" + "+-----\n");
            break;
        case 2:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   /|\n" + "|"|\n" + "|"|\n" + "+-----\n");
            break;
        case 1:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   /|\n" + "|"   / \n" + "|"|\n" + "+-----\n");
            break;
        case 0:
            dibuix = ("-----\n" + "|" /   |\n" + "|"   O\n" + "|"   /|\n" + "|"   / \n" + "|"|\n" + "+-----\n");
            break;
    }
    retorna dibuix;
}

```

```

}

private static void dibuixar(String dibuix2) {
    imprimeix(dibuix2);
}

private static void botarLletres() {
    caràcter lletra2 = ' ';
    intenta {
        mentre (lletra2 != '\n') {
            lletra2 = (char) llegir;
        }
    } atrapa (excepció) {
    }
}
}

```

CLASSE DICCIONARIAMBFITXER.JAVA

```

import java.io.*;
import java.util.*;

public class DiccionariAmbFitxer {

    declaracions
    public static String DiccionariAmbFitxer[] = new String[20];
    public static FileReader fin;
    public static BufferedReader entrada;
    public static enter i = 1;
    public static Random aleatori = new Random();

    public DiccionariAmbFitxer() {
        intenta{
            obrirDiccionari();
            String linia;
            linia = entrada.llegir;
            mentre (linia no buida) {
                per (i = 1; i < 21; i++) {
                    linia = entrada.llegir();
                    DiccionariAmbFitxer[i] = linia;
                }
            }
            tancarDiccionari();
        } atrapa (excepció) {
        }
    }

    public static void obrirDiccionari() {
        intenta {
            fin = new FileReader("diccionari.txt");
            entrada = new BufferedReader(fin);
        } atrapa (FileNotFoundException fnfex) {
            imprimeix("Fitxer no trobat.");
        } atrapa (excepció) {
            imprimeix("ERROR: " + e.getMessage());
        }
    }

    public static void tancarDiccionari(){
        intenta{
            fin.close();
        } atrapa (excepcio){}
    }
}

```

```
}  
public String obtenirParaula() {  
    String paraula;  
    enter nombre = obtenir nombre aleatori;  
    paraula = DiccioniariAmbFitxer[nombre];  
    retorna paraula;  
}  
}
```


- **Joc de proves que mostra per diferents entrades les respostes de l'aplicació**
- Lectura d'una lletra i comprovació de si és a la paraula (hi és):

```
Salida - JocDelPenjat (run) %
run:
JOC DEL PENJAT
El màxim d'errors permesos és: 7
Paraula: _ _ _ _
Introdueix una lletra: a
Paraula: _ a _ _
Introdueix una lletra: |
```

- Lectura d'una lletra i comprovació de si és a la paraula (no hi és):

```
Salida - JocDelPenjat (run) %
Introdueix una lletra: w
+-----
| /   |
|
|
|
+-----
Paraula: _ a _ _
Introdueix una lletra:
```

- Introducció de més d'una lletra:

```
Salida - JocDelPenjat (run) ✖
Introdueix una lletra: bfhdft
+-----+
| /      |
|        | O
|         |
|         |
|         |
+-----+

Paraula: _ a _ _

Introdueix una lletra:
```

- Introducció de números:

```
Salida - JocDelPenjat (run) ✖
Introdueix una lletra: 7
Això és un número!
Introdueix una lletra:
```

- Introducció d'un espai en blanc:

```
Salida - JocDelPenjat (run) ✖
Introdueix una lletra:
Això no és una lletra!
Introdueix una lletra: |
```

- Final (si has guanyat):

```
Salida - JocDelPenjat (run) ✖
Paraula: j a _ _
Introdueix una lletra: z
Paraula: j a z z

Enhorabona!!!
La paraula secreta era: jazz
GENERACIÓN CORRECTA (total time: 9 minutes 10 seconds)
```

- Final (si has perdut):

```
Salida - JocDelPenjat (run) ✖
▶ Introduueix una lletra: x
▶
■ +-----
  | /      |
  |      O
  |      /|\
  |      / \
  |
  +-----

Paraula: _ r _ n _ _ _ r _ n _

Has perdut!
La paraula secreta era: transparent
GENERACIÓN CORRECTA (total time: 16 seconds)
```

- **Conclusions que sintetitzen els conceptes apresos i la problemàtica a la que s'ha fet front**

En un principi, quan ens varen comunicar que havíem de fer el joc del penjat en java tothom va pensar que era massa complicat, però una vegada que comences a pensar-ho una mica, amb els coneixements que hem après al llarg del curs tant a les classes teòriques com a les pràctiques, no resulta tan difícil. Un avantatge molt important que hem tingut es que els dibuixos s'havien de fer amb els caràcters del teclat i no amb objectes de la classe Gobject (ja que durant el curs no hem vist gràfics).

Crec que aquesta pràctica obliga a l'estudiant a tenir clars els conceptes més bàsics i com operar amb ells, ja que per fer-la, he hagut d'emprar instruccions que hem fet servir cada dia a classe. Un problema que crec que pot haver sigut general és el fet de generar un nombre aleatori, ja que és una cosa que mai havíem fet a classe, i tots hem hagut de cercar diferents maneres de fer-ho, però principalment hi ha hagut dues maneres: amb la classe Random (que es com ho he fet jo) i amb la classe Math.

En aquesta pràctica s'ha vist clarament la reutilització de variables, l'ús de taules i tècniques de recorregut, a més de la lectura des de fitxers o l'ús de Strings. A més, crec que el desenvolupament d'aquest programa ha augmentat les meves habilitats per programar i ha fet possible una comprensió adequada dels temes i elements vists a les classes teòriques.

Finalment vull remarcar que els problemes que van sorgir varen ser pocs i esporàdics, i que amb l'ajuda d'Internet i de les classes impartides per *Mehran Sahami* a la *Stanford University* s'han pogut solucionar sense gaire complicació.