Universidad Veracruzana

Facultad de Estadística e Informática

Curso PROFA: Desarrollo de recursos educativos abiertos (REA)

Autor: Adriana Cervantes Castillo

adcervantes@uv.mx

Actividad 3: Desarrollo de un REA individual

Contenido

Introducción al lenguaje de programación java

Introducción

A través de este REA se pretende introducir al lector al lenguaje de programación java. Para lograrlo se presentarán los elementos básicos del lenguaje, la estructura general de un programa en java, los tipos de datos que soporta, el manejo de variables y constantes, el uso correcto de estructuras de selección y control y por último la propuesta y solución de algunos ejercicios básicos.

El objetivo final de este Rea es fortalecer los saberes teóricos de la experiencia educativa Programación agregando este nuevo saber al inicio del programa. Esto permitirá tener alumnos con un conocimiento uniforme del lenguaje de programación java, lo que les permitirá avanzar con mayor confianza y seguridad en el resto del curso.

Elementos básicos de java

Antes de iniciar a programar es necesario conocer los elementos básicos que conforman un programa en java.

Clases: Un programa en java consiste en la declaración de una clase.

Constructores: Métodos que se nombran igual que la clase. No tienen tipo de retorno y pueden o no recibir parámetros. Sirven para instanciar la clase y para inicializar los atributos o variables de la clase.

Nombre del programa o archivo: Debe coincidir con el nombre de la clase principal, es decir aquella clase que contiene el método main.

public static void main(String[] arg){}: Método por el cual java inicia la ejecución del programa.

Métodos de la clase: Métodos generados por el usuario dentro de la clase.

Declaración de variables y constantes: identificadores creados por el usuario que describen las características o propiedades de la clase.

Un programa general en java

Un programa general en java consta de las características anteriores. La imagen 1 muestra la estructura general de un programa en java. En la imagen1 se puede observar que el primer paso para crear un programa en java es declarar una clase con la sentencia:

public class nombre_de_la_clase{}.

Dentro de la clase encontramos la definición de variables, para este ejemplo se crean tres variables **var1** del tipo de dato doble, **var2** del tipo de dato entero y **var3** del tipo de dato cadena.

Seguido de la declaración de variables encontramos la declaración del **constructor sobrecargado**, es decir, al constructor que recibe los parámetros necesarios para inicializar las variables antes declaradas.

Seguido del constructor se encuentra la declaración del método **main.** A partir de este método es que inicia la ejecución del programa. En el cuerpo de este método debe codificarse o invocarse todo lo que se necesite ejecutar.

Por último seguido del método main se encuentran los métodos propios del usuario.

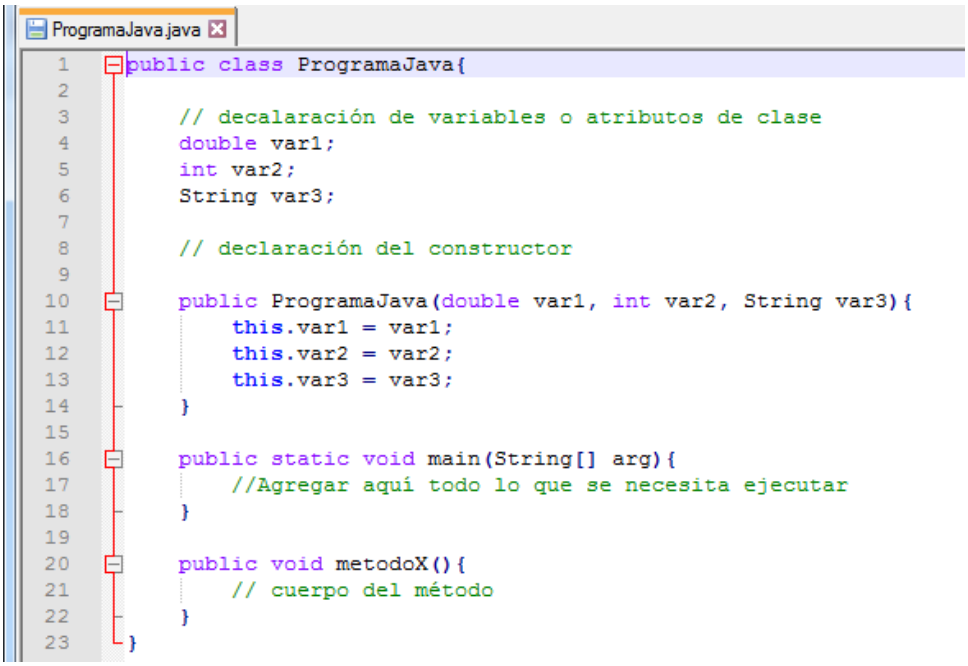

Imagen 1: Estructura general de un programa en java

Tipos de datos

Tipos de datos primitivos

double, int, char, boolean, float, short, long. Los tipos de datos primitives de inicializan a su valor por default, los de tipo numérico a 0, los de tipo boolean a true y los de tipo char a ‘’.

Tipos de datos referenciados

Todas las variables que apuntan algún tipo de clase en particular se conocen como tipos de datos referenciados ya que hacen referencia a alguna clase en particular. Por ejemplo:

String var3; var3 esta haciendo referencia a la clase String de java

Persona varPer; varPer hace referencia a la clase Persona. A través de la variable se puede acceder a los métodos y variables de la clase Persona.

Este tipo de datos se inicializa por default a null.

Variables y constantes

Constantes

Una constante es una variable cuyo valor no puede ser modificado dentro del programa. Se definen utilizando la palabra reservada final. Ejemplo

final tipo nombre = valor

final int var2 = 5;

Variables

Una variable es un identificador declarado por el usuario que puede tomar distintos valores dentro del programa. Antes de utilizarse deben ser declaradas e inicializadas. Una variable puede ser declarada fuera de los métodos, en este caso, su alcance es global a todos los elementos de la clase. Una variable puede ser declarada dentro de un método o dentro de un bloque, en este caso su alcance es local ya sea dentro del método o dentro del bloque. Para declarar una variable basta con definir e tipo y darle un nombre a la variable. Por Ejemplo:

double var1;

Se le puede asignar un valor por default al momento de su definición o asignarle un valor antes de ser accedida.

Entradas y salidas

Entradas

Existen distintas clases que java ofrece para capturar entradas ya sea desde la entrada estándar que es el teclado o desde algún archivo en particular. Para capturar información a través del teclado se puede utilizar la clase Scanner proporcionada por java en el paquete java.util. Basta con crear un objeto de esta clase y a partir de este objeto invocar al método apropiado de acuerdo a lo que se espera recibir desde el teclado, por ejemplo si es un entero se captura con el método nextInt(), si es un doble con el método nextDouble(), si es un String con el método nextLine.

El objeto de la clase Scanner se crea pasando al constructor de la clase la sentencia System.in para indicar que se va a leer información desde la consola. A continuación se muestra un ejemplo

Scanner objScanner = new Scanner(System.in);
String cadenaLeida = objScanner.nextLine();

Lo referente a archivos esta fuera del alcance de este REA

Salidas

Para poder mostrar información en pantalla se hace uso del método print() o println de la Clase System.

Por ejemplo si se necesita sacar por la pantalla la cadena "Hola amigos", basta con escribir la siguiente sentencia.

System.out.println ("Hola amigos");

El método print/ln recibe como parámetro una cadena, así que, todo lo que sea colocado dentro del método, será tomado como cadena.

Estructuras de Selección

Sentencia if-else

La estructura de la sentencia if es la siguiente:

```
If(expresion) {
//cuerpo
}
```

La expresión es una comparación (<,>,==,!=,…) entre dos elementos a,b.

La sentencia if es utilizada para evaluar una expresión, si la expresión evaluada se cumple, es decir es verdadera, entonces se ejecuta lo que se encuentra dentro del cuerpo de la sentencia if. Si la expresión no se cumple entonces es posible ejecutar el código de una sentencia else . Ejemplo:

```
If(){
//cuerpo
}else{
//cuerpo
}
```

Sentencia switch

A partir de esta sentencia es posible evaluar un conjunto de casos posibles asociados a la expresión. Si algún valor asociado a la expresión coincide, entonces se ejecuta el código correspondiente a dicho caso. Ejemplo:

switch(expresion){
case valor1:
// codigo

break;
case valor2:
//código
break;
….
default:
}
La sentencia break permite que se evalué cada valor de la expresión. Si esta no esta presente al final de cada case, entonces se evaluarán los case en cascada. Si ningún case es evaluado, entonces se ejecuta el código de la sentencia default.

Estructuras de control

Sentencia while

La estructura de un while es la siguiente:

while(expresion){
//ejecutar codigo
}
La sentencia while permite ejecutar un bloque de código mientras se cumpla expresión evaluada en el while. Si la expresión no se cumple, entonces el código dentro del while no se ejecutará. Se puede utilizar la sentencia break dentro del while para parar la ejecución de éste.

Sentencia for

for(expresion){
//cuerpo for
}
Donde: expresión se compone de tres elementos:

Int i: contador inicial del que partirá el for. El cual se incrementará en cada iteración.

i<10; La condición de paro del for. En el momento que i tenga un valor igual o superior a 10, el for terminará se ejecución.

i++: incrementar contador

Al igual que la sentencia while, la sentencia for permite la ejecución del código dentro del for mientras la evaluación de la expresión se cumpla. La diferencia entre ambos es que en el for el operador de incremento y de paro está dado en la expresión, mientras que en el while la condición de paro está dada por la expresión y el operador de incremento se encuentra dentro del cuerpo.

Ejercicios

Ejercicio 1: Escribe un programa en java que imprima la cadena “hola”.

```
public class Ejercicio1{
        public static void main(String[] arg){
                System.out.println(“Hola”);
        }
}
```

Ejercicio 2. Escribe un programa en java que declare una variable de tipo entera con un valor inicial de 5, un una variable doble con un valor de 2. El programa debe guardar en una tercer variable suma de tipo double, la imprimir el valor de la variable suma.

```
public class Ejercicio2{

        public static void main(String[] arg){
                int var1 = 5;
                double  var2 = 2;
                double suma =0;
                suma = var1 + var2;
                System.out.println(suma);
        }
}
```

Ejercicio 3. Escribir un programa en java que imprima la suma de los primeros 10 números si se cumple que a es menor a 10 utilizando un ciclo for, de lo contrario utiliza un ciclo while para imprimir la suma requerida.

```
public class Ejercicio3{

        public static void main(String[] arg){
                int a = 5;
                int suma = 0;
                if(a < 10){

                        for(int i=1; i<=10; i++)
                                suma = suma + i;
                }else{
                        Int i=1;
                        while(i<=10){
                                suma = suma + i;
                                i++;
                        }
                }
        }
}
```

Referencias

L. Joyanes-Aguilar, I. Zahonero-Martinez; Programación en java 6: Algoritmos y programación orientada a objetos.McGraw-Hill, 2008.
C.S Hortsmann,G.Cornell; Core java:volumeI fundamentals, Prentice-Hall 2013.