

Tema: I

Introducción a Vala

Prácticas.

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

Curso 2014-2015 , Copyleft © 2011-2015 .
Reproducción permitida bajo los términos de la licencia de
documentación libre GNU.

Contenido

- 1 Entorno de desarrollo
- 2 Primeros pasos
- 3 Compila éste código
- 4 Depura éste código
- 5 Modificaciones al código anterior
- 6 Ejercicio (I)
- 7 Ejercicio (II)
- 8 Objetivos...

- Trabajaremos en GNU/Linux. Concretamente con la distribución que la EPS instala en sus laboratorios.
- Podemos usar el editor que queramos: gedit, kate, joe, vim, etc. . .
- Compilaremos desde un terminal.

- Comprueba que herramientas relacionadas con Vala tenemos instaladas.
- Abre un terminal, teclea 'vala' y sin pulsar espacio pulsa dos veces la tecla TAB.
- Una de las herramientas instaladas debe ser el compilador de Vala: 'valac'.
- Comprueba la versión instalada del mismo: `valac --version`

Compila éste código

- Crea un archivo llamado 'holamundo.vala' con el contenido siguiente:

```
1      /*
      * Hola mundo.
3      * fecha: 07/02/2012
      */
5      class HolaMundo : GLib.Object {
          public static int main(string[] args) {
7          stdout.printf("Hola Mundo\n");
              return 0;
9      }
      }
```

- Compilalo con: 'valac holamundo.vala'.
- El resultado debe ser un ejecutable llamado 'holamundo' que puedes poner en marcha desde el terminal tecleando './holamundo' y pulsando 'enter'. Genera el código 'C' que realmente se compila, usa la opción '-C' del compilador de Vala.

- Recuerda que Vala genera un ejecutable en código máquina de la arquitectura destino, no de una máquina virtual.
- Para depurar a nivel de código fuente necesitamos compilar con las opciones: `--save-temps -g`, es decir, algo así:
`valac --save-temps -g holamundo.vala`.
- Para depurar podemos emplear en modo texto `'gdb'`: `'gdb holamundo'`.
- También podemos usar `'nemiver'`: `'nemiver holamundo'`.

Modificaciones al código anterior

- El código anterior recibe como parámetros del método `main` los argumentos del programa en el array `'args'`.
- Modifícalo para que después de imprimir `'Hola Mundo'` imprima cada uno de estos argumentos.
- Modifícalo una vez más para que si no recibe ningún argumento como parámetro, te pida el nombre y después de imprimir `'Hola Mundo'` imprima tu nombre.

Ejercicio (I)

Escribe una aplicación en Vala que permita al usuario adivinar un valor numérico entero entre dos números dados, teniendo en cuenta lo siguiente:

- 1 Al comenzar pedirá los valores mínimo y máximo entre los que estará el valor a adivinar.
- 2 Generará este valor a adivinar de forma aleatoria¹.
- 3 Preguntará el nombre al jugador y comenzará el juego.

¹Puedes ver ejemplos en [Vala sección Mathematics](#) .

Ejercicio (II)

- 1 Tras cada intento fallido del jugador le preguntará si quiere seguir probando suerte o abandonar. Si abandona preguntará si termina o sigue la partida y si sigue pedirá el nombre al siguiente jugador (sin cambiar el número a adivinar) y seguirá jugando el nuevo jugador.
- 2 Tras cada acierto del número por parte de un jugador guardará junto al nombre del jugador el numero de intentos que ha necesitado para adivinar el número y a continuación preguntará si termina o sigue la partida y si sigue volverá a generar un nuevo número entre el intervalo inicial comenzando el juego de nuevo pidiendo el nombre al nuevo jugador.
- 3 Al terminar la partida mostrará las puntuaciones obtenidas por cada jugador, ordenadas de menor a mayor número de intentos necesarios.

El alumno sabe:

- ☐ Invocar el compilador de vala desde la línea de órdenes.
- ☐ Conocer la versión del compilador empleada.
- ☐ Generar código ejecutable con información para ser depurado.
- ☐ Usar un depurador en modo texto o gráfico.
- ☐ Generar código 'C' a partir de código Vala.