

PROYECTO

1 - DESCRIPCIÓN GENERAL

El proyecto consiste en el estudio y desarrollo de técnicas de implementación de conceptos, herramientas, mecanismos o constructores provistos por los lenguajes de programación. Las tareas a desarrollar se presentarán en una serie de consignas divididas en dos partes: una de traducción a SimpleSem y la otra de análisis y explicación de la traducción realizada. Cada comisión tendrá asociado un tema, el cual determinará el código que deba ser traducido. Además del ejercicio de traducción obligatorio, cada comisión podrá optar por incorporar a su traducción elementos adicionales. Al realizar los ejercicios adicionales los alumnos pueden mejorar la nota final de la materia para aquellos alumnos que promocionen, y para aquellos alumnos que tengan que rendir el examen final, determinará si se les tomarán ejercicios de traducción en el final, o no.

CÓDIGO DE HONOR

Se espera que cada comisión se comprometa a resolver el trabajo de manera autónoma. Las partes o ideas que hayan sido tomadas de otras fuentes, no deben constituir partes esenciales de la tarea y deben estar claramente identificadas.

2 - PAUTAS DE ENTREGA

La entrega del proyecto consistirá de un archivo (*.simplON*) con la traducción correspondiente al programa Java que le fue asignado (Secciones 4 y 5), y un informe con las respuestas a las consignas de análisis y explicación (Sección 6). Todos los archivos que entreguen deben ser comprimidos dentro de un *.zip*. En caso de que el código fuente entregado haya sido modificado con el objetivo de cumplir con alguno de los puntos opcionales (Sección 7) deberá incluir en el *.zip* los nuevos archivos fuente. Además, se deberá hacer mención explícita de estas modificaciones dentro del informe.

La resolución deberá ser enviada por **Moodle** antes del **01/07/21** a las **14.00 hs.**

3 - DESAPROBACIÓN Y RE-ENTREGA

En caso de desaprobación la primera entrega tendrán posibilidad de una única re-entrega cuya fecha límite es el día **13/07/21** a las **14.00 hs.** La desaprobación de la re-entrega del proyecto implica la desaprobación del cursado de la materia.

4 - TRADUCCIÓN

Realizar la traducción completa a SimpleSem de los archivos *.java* del tema asociado, a excepción del archivo "Sistema.java". El programa SimpleSem resultante de la traducción debe ser tal que al ejecutarse en SimplON tenga el mismo comportamiento que el de compilar y ejecutar los *.java* del tema asociado. En la sección de código puede ver todas las clases que debe traducir.

Para la traducción deberán utilizar comentarios para indicar qué sentencia (o partes, en caso de involucrar llamadas) de los archivos fuente *.java* se está traduciendo, y utilizar anotaciones para decorar la memoria con información vinculada a los registros de activación, INSTs y VTs. Sean criteriosos en el uso de estas herramientas ya que la valoración del programa SimpleSem entregado estará fuertemente influenciada por la claridad de la traducción.

Cualquier decisión de diseño o aclaración necesaria deberá estar documentada en el informe a entregar. En el informe **no** debe incluirse el código SimpleSem resultante de la traducción.

5 - CÓDIGO JAVA

<pre> public class A { int v1, v2; public void init() { v1 = 3; v2 = 6; } public void m1() { int i = 0; while((v2>v1 v1+v2 >= 5) & i<50){ i = i+1; v1 = ((v1-i)*v2)/i; if(v1+v2+i > v2*i){ continue; } v2 = ((v2+3)*(5+i))/i; } } public int m2(int x, int y) { v1 = v1+x; v2 = v2+y; this.m1(); return v1+v2; } } </pre>	<pre> public class B extends A{ int max; public void init(){ super.init(); v1 = v1*5; v2 = v2*5; max = 4; } public int m2(int x){ if(x < 10 max == 1){ super.m1(); max = 4; return v1+(v2-x); } max = max - 1; return this.m2(x / 10); } } public class C extends B{ B v3; public void init(){ super.init(); v3 = new B(); v3.init(); } public void m1(){ super.m1(); v1 = v1+this.m2(5+v3.m2(v1,v2)); } } </pre>
<pre> public class Principal { public static void main(String[] args){ A oA; int x1, x2; oA = new C(); oA.init(); x1 = Sistema.read(); x2 = Sistema.read(); Sistema.print(oA.m2(x1,x2)); } } </pre>	

6 - ANÁLISIS Y EXPLICACIÓN

- Esquematice los INSTs y VTs de todas las clases del tema asociado y también los registros de activación de todos los métodos. No realice esto para los componentes de la clase Sistema.
- Explique de forma intuitiva como hizo para traducir el comportamiento del mecanismo de iteración de método `m1` de la clase A.
- Indique cuales fueron las complicaciones con las que se topó al traducir el método `m2(int)` de la clase B, y explique de forma intuitiva cómo hizo para resolver estos problemas.
- Indique cuales fueron las complicaciones con las que se topó al traducir la asignación a `v1` en el método `m1` de la clase C, y explique de forma intuitiva cómo hizo para resolver este problema.

7 - EJERCICIOS OPCIONALES

En caso de realizar algún ejercicio opcional, méncionelo de forma explícita dentro del informe, explicando en cada caso cuales fueron los pasos o estrategia a seguir para resolverlo.

- Incorpore a la traducción chequeos para controlar que las referencias a objetos no sean nulas antes de enviar un mensaje.
- Defina un constructor para alguna de las clases que reciba al menos un parámetro, y luego realice la traducción de este constructor. Tenga en cuenta que los constructores son métodos de clase.
- Agregue al código una asignación condicional y luego incorpórela a la traducción. Explique de forma intuitiva cuál fue la estrategia que siguió para realizar la traducción.
- Modifique alguna de las clases de tal forma que uno de los métodos agregue el mecanismo de selección múltiple **switch** (que solo trabaje con enteros). Luego, traduzca la nueva versión de este método. Explique intuitivamente cómo haría para comparar cualquier tipo de objetos con el constructor **switch**.
- Modifique las traducciones correspondientes a la evaluación de expresiones booleanas para que estas se evalúen por corto-circuito.
- Modifique alguna de las clases para incorporar en algún método una llamada encadenada de la forma: **this**.<getObj1()>.<getObj2()>.<metodoObj2()>. Luego, traduzca la nueva versión del código, y explique de forma intuitiva la estrategia que utilizó para realizar esta traducción.
- Modifique alguna de las clases de tal forma que uno de los métodos tenga que recorrer un arreglo. Luego, traduzca la nueva versión del código. Explique intuitivamente cuál fue la estrategia que utilizó para representar el arreglo en memoria y para acceder a los distintos elementos de la estructura.