

Examen 2

En este documento se responderán algunas de las preguntas del examen1 de lenguajes de programación, los códigos se pueden encontrar en el siguiente link de github:

<https://github.com/AngelVzla99/materia-lenguajes-de-programacion.git>

1. Pregunta 1

Como mi nombre empieza con A, seleccione de lenguaje a Ada

a) Descripción del lenguaje

1) Ada tiene alcance estático y asociación profunda. Esto permite identificar fácilmente al momento de estar codeando a cual variable se refiere (en el caso de las variables, las mas externa en el código) sin embargo no es tan flexible para generalizar funciones que usen variable redefinidas antes de llamar a al funcion.

2) Ada tiene diferentes módulos como:

a' module-text_file

b' module-html

c' module-xy

d' modulelatex

e' module_interface

f' module-console

Para importar un modulo se puede hacer:

with Ada.Text_IO; use Ada.Text_IO;

Tambien se les puede renombrar

with Outer;

with Outer.Inner_1;

package New_Package is

OI_1 renames Outer.Inner_1;

type New_type is new OI_1.A_Type;

end New_Package;

Para exportar un modulo de puede hacer

package example is

```
:  
:  
end example;
```

- 3) Ada permite Alias ya que posee punteros, si se quiere crear un alias simplemente se usa **rename** en la declaración de la variable

https://rosettacode.org/wiki/Pointers_and_references#Pointers_4

Ada permite la sobrecarga https://www.adaic.org/resources/add_content/standards/05rat/html/Rat-2-7.html

Ada tambien permite polimorfismo <http://computer-programming-forum.com/44-ada/f7a12d26596d3f68.htm>

- 4) Ada posee varios compiladores como: 'GNAT Ada 95', 'Aonix Object Ada compiler' y 'Ada + SQL compiler'. Ada + SQL compiler ofrece opciones de debugger tambien esta AdaCAPS que ofrece un debugger. Ademas como profiler para Ada se tiene a 'gprof'

b) Implementación

Para probar los algoritmos use el compilador de ada onlie:

<https://www.jdoodle.com/execute-ada-online/>

- 1) Se encuentra en la carpeta de github **examen1/pregunta1/pow.adb**
- 2) Se encuentra en la carpeta de github **examen1/pregunta1/matrix.adb**

2. Pregunta 2

Esta pregunta decidir hacerla en una presentacion de google la cual se puede encontrar con el siguiente link:

https://docs.google.com/presentation/d/1DS269w4j09xA_S8I6GNuF4fy0mqdHhtVFEY6DJcLtfM/edit?usp=sharing

3. Pregunta 3

Decidí utilizar para esta pregunta C++. El programa se encuentra en la carpeta de github **examen1/pregunta3** para ejecutar el programa solos se debe ejecutar el comando **make** seguido de **./main <CANTIDAD_DE_BLOQUES>**. Los test tienen una cobertura del 86.67 %

Parar correr únicamente los test unitarios se corre el comando **make test_unit** seguido de **./test_unit**

Para correr el codigo de cobertura de los testcases **make test_unit_coverage**

4. Pregunta 4

Decidí utilizar para esta pregunta C++. Hay un ejemplo de como usar el modulo en **main.cpp** o directamente en **test_unit.cpp** donde hay un ejemplo de cada funcionalidad. El modulo se encuentra en la carpeta de github **examen1/pregunta4** específicamente en el archivo **moduloVec.h**. Con los test unitarios se logro una cobertura del 95.19 %

Parar correr únicamente los test unitarios se corre el comando **make test_unit** seguido de **./test_unit**

Para correr el código de cobertura de los testcases **make test_unit_coverage**