



UNIVERSIDAD SIMÓN BOLÍVAR

Dept. Computación y Tecnología de la Información

Laboratorio de Algoritmos y Estructuras II

Abr-Jul 2016

## Laboratorio 1

1. Los estudiantes del Laboratorio de Algoritmos y Estructuras II decidieron hacer su propia librería de interfaz gráfica en Python debido a los inconvenientes que tuvieron el trimestre pasado usando la librería *Pygame*. Como es una ardua tarea, los estudiantes deberán realizar sólo una pequeña parte de esta usando clases. Para ello se deberá crear:
  - Una clase *Canvas* la cuál tendrá como atributos la altura, el ancho del espacio de trabajo y la información de la matriz inicializada en cada posición con " – ". La dimensión de la matriz viene dada por la altura y el ancho del canvas. Dicha clase tendrá los siguientes métodos:
    - `setpixel(row, col)`: Pintar un pixel. Pintar un píxel será colocar un \* en esa posición de la matriz. Este método no debe de imprimir en pantalla.
    - `getpixel(row, col)`: Devolverá el pixel pintado (" \* ") o no pintado(" – ") de una posición.
    - `deletepixel(self,row,col)`: Borrará ese pixel.
    - `display()`: Imprimirá en consola la matriz con los píxeles pintados(" \* ") o no pintados(" – ").
  - Una clase *Rectangulo* la cual debe tener como atributos las 4 coordenadas que se necesitan para saber cuáles son los puntos que forman un rectángulo y un método `paint(canvas)` que se encarga de pintar todos los píxeles (asignarle a esos píxeles un " \* ") para formar el rectángulo. Este método no debe de imprimir en pantalla.
  - Una clase *Cuadrado* que debe heredar todos los métodos la clase Rectángulo. No se pueden copiar y pegar los métodos de la clase Rectángulo.

El archivo donde se encuentran implementadas las clases se llamará `lab01Ejer1.py` y allí mismo deberá crear un pequeño programa donde pruebe que la clase y cada uno de sus métodos funcionan.

2. El trimestre pasado a la profesora Catherine se le perdió unos componentes electrónicos en el laboratorio que necesitaba para construir su robot. Amablemente sus estudiantes la ayudaron pero cuando terminó el proyecto notó que se le habían vuelto a perder pero esta vez su lista de componentes que se encuentran en el laboratorio está desordenada y no tiene tiempo de arreglarla. Necesita para hoy antes de las 4:30pm encontrar los componentes importantes para poder armar su robot entre los que estan en el laboratorio.

Las listas referidas anteriormente se muestran a continuación:

Los componentes que se encuentran en el laboratorio = [4, 20, 5, 5, 10, 6, 7, 8, 16, 9, 11, 14, 5]

Los componentes que la profesora necesita saber si se encuentran en el laboratorio = [4, 7, 8, 9, 14]

Como la profesora está muy apurada, deberán implementar el algoritmo de búsqueda que, en general, sea más eficiente para esta tarea. Por otro lado la profesora desea comparar el funcionamiento de los 3 algoritmos de ordenamiento (inserción, burbuja y selección) en dicho problema, por lo que sus estudiantes deberán implementar los 3 algoritmos.

Deberán existir entonces tres archivos:

- En el archivo lab01Ejer2B.py debe de estar implementada la solución del problema con el Ordenamiento Burbuja.
- En el archivo lab01Ejer2I.py debe de estar implementada la solución del problema con el Ordenamiento por Inserción.
- En el archivo lab01Ejer2S.py debe de estar implementada la solución del problema con el Ordenamiento Selección.

Como la profesora se guía por el libro del curso, dichas implementaciones deberán de ser según el pseudocódigo del Kaldewaij encontrado en este link: <http://chimo.ldc.usb.ve/ProgrammingTheDerivative>. Es decir, no se podrá usar ningún otro pseudocódigo o código encontrado en otros libros o internet para dicha implementación.

La profesora necesita que sus estudiantes le resuelvan una última duda y su respuesta la deberán dejar comentada al final del código implementado como un comentario (en el archivo lab01Ejer2B.py):

¿En el caso que sólo desee buscar el componente 4 cuál algoritmo de búsqueda es más eficiente: búsqueda binaria o búsqueda lineal? ¿Y en el caso que busque sólo el elemento 14?

### Condiciones de la entrega

Cree un archivo comprimido del tipo “tgz” llamado Lab1-X-Y.tgz, donde X es su número de carné y Y el de su compañero. Su contenido serán los programas lab01Ejer1.py, lab01Ejer2B.py, lab01Ejer2I.py y lab01Ejer2S.py. Debe subir el archivo en el aula virtual, en la sección de Semana 2, el día 21 de abril de 2016 antes de las 4:30 p.m.