



## Programación I

Parcial (turno mañana) – 12/06/2019

### Ejercicio 1 (20 pts. - 5 pts. cada item.)

Discutir la veracidad de las siguientes afirmaciones, justificando su respuesta.

- a) El método `CharAt` de la clase `String` es un método de clase que sólo puede recibir como parámetro un entero entre 0 y el largo del `String`.
- b) Se produce aliasing cuando dos variables referencian a objetos con los mismos valores en sus variables de instancia.
- c) La función de complejidad mide cuánto tiempo tarda un programa en ejecutar.
- d) La concatenación de dos listas enlazadas sólo es posible si las listas guardan una referencia tanto al primero como al último nodo.

### Ejercicio 2 (20 pts.)

Escribir una función recursiva `String letrasMayores(String s1, String s2)` que toma dos `Strings` `s1` y `s2` y retorna una nueva cadena que en cada posición tiene el caracter de esa posición de `s1` o de `s2` que sea mayor en el orden del diccionario. Si uno de los `Strings` es más largo que el otro, al final aparecerán los caracteres del `String` más largo. Por ejemplo:

- `letrasMayores("aazz", "bubu")` debe devolver `"buzz"`.
- `letrasMayores("camas", "bus")` debe devolver `"cusas"`.
- `letrasMayores("ala", "ajo")` debe devolver `"alo"`.
- `letrasMayores("luz", "")` debe devolver `"luz"`.

Se pide resolver **utilizando recursión**. Se pueden dar por hecha la función `String resto(String s)` que devuelve una cadena igual a `s` pero sin su primer caracter.

### Ejercicio 3 (35 pts. - 15 pts. item a.- y 20 pts. item b)

Consideremos las siguientes clases:

```
public class ClinicaVeterinaria {
    public Atencion[] atenciones;
    ...
}

public class Atencion {
    public Veterinario[] veterinarios;
    public Mascota[] mascotas;
    public Medicamento[] medicamentos;
    ...
}

public class Mascota {
    public String tipoDeAnimal;
    public String raza;
    public int edad;
    ...
}

public class Veterinario {
    public String nombre;
    public String apellido;
    public int legajo;
    public int antigüedad;
    ...
}

public class Medicamento {
    public String tipoDeAnimal;
    public double dosis;
    public double precio;
    ...
}
```

Estas clases modelan una Clínica Veterinaria. Cada `Atencion` representa una atención médica que involucra un arreglo de uno o más `Veterinarios` que atendieron a un arreglo de una o más `Mascotas`, y un arreglo de `Medicamentos` que se administraron en dicha atención. Para la clase `ClinicaVeterinaria`, se pide:

- a) Escribir el método **public Set<Atencion> atencionesMasCostosas(double costoMin)** que devuelve el conjunto de atenciones que costaron más de **costoMin** de pesos. Para el *costo total* de una atención se cobran \$ 500 por veterinario involucrado en la atención, más la suma de los *precios* de los medicamentos utilizados.
- b) Las atenciones que involucran a los veterinarios con mayor experiencia son las más críticas. Escribir el método **public int atencionesCriticas(int antiguedadMinima)** que calcula la cantidad de atenciones que contaron con al menos un veterinario con una *antigüedad* de **antiguedadMinima** años o más.
- c) (*bonus track* 20 pts.) Algunos medicamentos para un tipo de animal se pueden usar para tratar a un tipo de animal diferente (por ejemplo: un desparasitante para perros pequeños se puede usar para gatos). Escribir el método **public String medicamentoEspecificoMasUsado()** que devuelve el tipo de animal del medicamento más utilizado en la veterinaria. Es decir, el tipo de animal del medicamento que se suministró en la mayor cantidad de las atenciones. En caso de existir más de un tipo de animal del medicamento con estas características se puede devolver cualquiera de ellos.

#### Ejercicio 4

Dadas las clases:

```
public class NodoInt
{
    int elemento;
    NodoInt siguiente;
}
```

```
public class ListaInt
{
    NodoInt primero;
    ...
}
```

Se pide para esta clase escribir el método de instancia **void filtrarElementosEnRango(int k, int m)** que modifica la lista dejando únicamente los elementos que son mayores a **k** y menores a **m**.

Por ejemplo

- Si la lista es [10,8,5,6,0,2,1], **k=5** y **m=9** la lista debe quedar [8,6].
- Si la lista es [2,1,2,3,0], **k=1** y **m=3** la lista debe quedar [2,2].
- Si la lista es [4,1,5,1,2], **k=5** y **m=0** la lista debe quedar [].

Se pide además que el método implementado **sea de orden lineal**, es decir,  $O(n)$  donde  $n$  es la cantidad de elementos de la lista. Justificar la complejidad del mismo.