**Alumno:** Olivarez Alexis Emanuel  **DNI**: 44.542.230 **Laboratorio**: 2

**Profesores:** Medina – Airaldi **TP**: 4 **Fecha de Entrega**: 24/09/23

En la guía de ejercicios del TP4, implementamos la reutilización de clases predefinidas, Estructuras Estáticas (Array), Estructuras Dinámicas (ArrayList), Mapas (HashMap) y JavaDocs.

**Array** es un contenedor que almacena datos primitivos o referencias de objetos de forma estática, del mismo tipo, es decir homogéneos, también el tamaño esta fijo y no puede variar durante la ejecución. Ejemplo:

***Punto*[] arrayPunto = *new* *Punto*[6];**

**ArrayList** es un contenedor que almacena elementos de forma dinámica. El tamaño de esta es dinámico, es decir puede variar durante la ejecución. Se puede agregar y eliminar estos elementos. Ejemplos:

***ArrayList***

***private* *ArrayList* empleados; // *Necesita ser casteado para recuperar los objetos de tipo Empleado.***

***this*.getEmpleados().add(p\_empleado); // *Agrega un empleado a la colección.***

***this*.getEmpleados().remove(p\_empleado); // *Quita un empleado de tipo Empleado de la colección.***

**ArrayList (Genéricos)**

***private* *ArrayList*<*Empleado*> empleados; // *Definimos que tipo de objetos recibe la colección.***

***private* *ArrayList*<*CuentaBancaria*> cuentaBancaria; // *No hace falta castear para trabajar con los elementos, en todo método o constructor se debe especificar al recibir como parámetro de que tipo es el ArrayList. ArrayList<Empleado>***

**Tipos de Recorridos (ArrayList y ArrayList Genéricos)**

***for* (*int* i = 0; i < arrayPunto.length; i*++*) {} // *For Tradicional.***

***for* (*Punto* punto *:* arrayPunto) {} // *For-each.***

Un HashMap es otro tipo de contenedor­ que almacena y manipula pares de tipo <clave, valor>. Ejemplo:

***private* *HashMap*<*Long*, *Empleado*> empleados; // *Es recomendable utilizar genéricos en lugar de hacer casting.***

Los objetos contenidos estarán identificados mediante algún atributo que lo caracterice (Algo que no se repita que sea único), por ejemplo, libreta universitaria, DNI, CUIL, número de producto, etc. Así se asocia cada objeto insertado (valor) a otro objeto identificador (clave). Esto facilita encontrar un elemento sin necesidad de recorrer por todos los elementos contenidos.

En este ejemplo la clave es el CUIL del empleado el método .get() le pasamos el CUIL y busca en la colección algún CUIL (clave) que sea igual a la que le pasamos para encontrar el valor asociado (Objeto Empleado) a esta si no lo encuenta devuelve null.

***public* *Empleado* buscarEmpleado(*long* *p\_cuil*) {**

***return* *this*.getEmpleados().get(p\_cuil);**

**}**

**Tipos de Recorridos (HashMap y HashMap Genéricos)**

***for* (*Object* object *:* *this*.getAlumnos().entrySet()) { // *Necesita ser casteado para recuperar los objetos de tipo Alumno de la colección.***

***Map*.*Entry* entry = (Map.Entry) object;**

***Alumno* alumno = (Alumno) entry.getValue();**

**}**

***for* (*Empleado* empleado *:* *this*.getEmpleados().values()) {} // *For-Each con el .values() recorre solo los valores de la colección (Genéricos).***