**INFORME: POGRAMACION II**

Luraschi, Roberto Gino

**Ejercicio 1:**

**Cashflow**

El metodo, cuando encuentra un saldo negativo, busca el proximo que lo supere, tal que lo pueda volver positivo o neutro.

**for**(**int** x = 0; x < registros.size(); x++){

**if**(saldoR + registros.get(x).getSaldo() < 0){

**int** y;

**for**(y = x+1 ; registros.get(y).getSaldo() < 0 ; y++){}

**final** Registro registroN = registros.get(y);

registroN.setDistancia(y-x);

registros.get(x).setDistancia(y-x);

registros.set(y, registros.get(x));

registros.set(x, registroN);

}

saldoR += registros.get(x).getSaldo();

}

La complejidad es O(*n²*) debido al primer for que implica que lo recorrera n veces, y el segundo for a lo sumo lo recorrera *n²/2* veces.

A su vez el ejercicio cuenta con 3 clases:

* **CashFlow**: Que es la clase principal con el flujo y el orden de los registros.
* **Registro**: Que contiene el registro con el saldo de el mes, y las posiciones que se movio para dar calculos exactos.
* Test:

**Ejercicio 2:**

**PPT:**

Para el PPT se utilizaron 2 clases(Ademas de la clase Test). Las clases son:

* **Elemento**: Tiene el nombre del elemento y el elemento a cual le gana.
* **PPT**: Flujo principal del juego, el cual contiene los metodos requeridos en el ejercicio(Agregar elemento, agregar regla y jugar).

Los metodos mas importantes son los siguientes:

**public** **static** **void** agregarRegla(String leGanaA) {

Elemento seleccionado = *extraerPrimerElemento*(leGanaA);

leGanaA = leGanaA.replace(seleccionado.getName(), "");

seleccionado.setLeGanaA(*extraerSegundoElemento*(leGanaA));

}

La funcion agregarRegla, recibe un string, y buscando dentro del metodo extraerPrimerElemento el primer Elemento, teniendo en cuenta que recibira por parametro una cadena “Elemento1 le gana a Elemento2”. Esto devolvera un elemento ya ingresado anteriormente. Lo que hace extraerSegundoElemento, devuelve solo el nombre de un elemento al que le ganara el elemento ingresado.

El siguiente metodo importante es:

**public** **static** Integer jugar(String jug1, String jug2) {

**for** (Elemento e : *elementos*){

**if**(jug1.indexOf(e.getName()) == jug2.indexOf(e.getName())) **return** 0;

**if** (jug1.indexOf(e.getName()) > -1) {

**if** (jug2.indexOf(e.getLeGanaA()) > -1)

**return** 1;

**return** 2;

}

}

**return** 0;

}

Este metodo, lo que recibe es lo que eligio el jugador 1 y lo que eligio el jugador 2. Primero veria si los elementos elegido por ambos son iguales y devolveria 0. Y en caso de no serlo, veria si el jugador 2 eligio el elemento al que le gana y en ese caso devuelve 1, de lo contrario, el jugador 1 perderia por el elemento elegido por el jugador 2.

**PPTLS:**

Para el PPTLS se utilizaron 2 clases(Ademas de la clase Test). Las clases son:

* **Elemento**: Tiene el nombre del elemento y los elementos a cual les gana.
* **PPTLS**: Flujo principal del juego, el cual contiene los metodos requeridos en el ejercicio(Agregar elemento, agregar regla y jugar).

Los metodos principales dentro de la clase PPTLS son:

**public** **static** **void** agregarRegla(String leGanaA) {

Elemento seleccionado = *extraerPrimerElemento*(leGanaA);

leGanaA = leGanaA.replace(seleccionado.getName(), "");

seleccionado.agregarLeGanaA(*extraerSegundoElemento*(leGanaA));

}

Esto agegaria una regla, extrayendo el primer elemento por el nombre y extrae el segundo elemento mediante el siguiente metodo:

**private** **static** String extraerSegundoElemento(String leGanaA) {

**for** (Elemento e : *elementos*)

**if** (leGanaA.indexOf(e.getName()) > -1) {

leGanaA.replace(e.getName(), "");

**return** e.getName();

}

**return** "";

}

El metodo principal seria jugar y esta hecho de esta forma:

**public** **static** Integer jugar(String jug1, String jug2) {

**for** (Elemento e : *elementos*){

**if**(jug1.indexOf(e.getName()) == jug2.indexOf(e.getName())) **return** 0;

**if** (jug1.indexOf(e.getName()) > -1) {

**for** (String e1 : e.getLeGanaA()) {

**if** (jug2.indexOf(e1) > -1)

**return** 1;

}

**return** 2;

}

}

**return** 0;

}

El metodo incluye recibir 2 parametros con lo elegido por los jugadores, busca si elemento por elemento, si el elemento recibido por el jugador 1 es igual al del 2 devuelve 0, si encuentra el elemento recorre los elementos a los que le gana el elegido por el jugador 1, si lo encuentra devuelve que gano el jugador 1, si no lo encuentra, retora 2.

**Ivariante de Representacion**

* Los elementos pierden con diferentes elementos que con los que ganan
* Los elementos son mas de uno