CURSO DE PROGRAMACIÓN FULL STACK

INTEGRADOR JAVA

INTEGRADOR JAVA

Llegó el momento de poner a prueba nuestros conocimientos de Java, para ello nos han propuesto el siguiente desafío:

En el siguiente link vamos a encontrar un proyecto de Java, que consta de tres problemas(funciones) a resolver. Cada función tendrá una consigna diferente y deberemos lograr que devuelvan diferentes resultados según parámetros que se nos van a especificar.

Link Proyecto: IntegradorJava

El proyecto consta de las siguientes clases:

Una clase Main, que tiene los llamados a las funciones a resolver, estos llamados nos van a servir para probar nuestras resoluciones . Estos llamados van a tener que ser descomentados para poder usarlos.

```java
package integrador;

import java.util.ArrayList;
import java.util.Arrays;

public class Integrador {

    public static void main(String[] args) {

        Practica practica = new Practica();

        //Generar las variables necesarias para probar
//        ArrayList<String> medias = new ArrayList(Arrays.asList(new String[]{"A", "B", "C", "D","A", "C", "D", "A"}));
//
//        System.out.println(practica.mediasAmigas(medias));
//
//        System.out.println(practica.numeroPalindromo(2L));

//        System.out.println(practica.prisioneroDulce(0, 10, 6));
    }
}
```

Una clase Practica donde encontraremos nuestros tres problemas.

```java
public class Practica {

    /**
     * El programa debera tomar un numero x y determinar si es palindromo o no
     * **Contemplar que el num que llega puede ser negativo. **Comtemplar que el
     * num que llega puede ser de un digito **Contemplar que el num que llega
     * puede ser null,en caso de que sea null, retornar false, en caso que sea
     * palindromo retornar true.
     *
     * @param num
     * @return esPalindromo
     */
    public Boolean numeroPalindromo(Long num) {
        //Aca escribir la logica necesaria
        return null;
    }
```

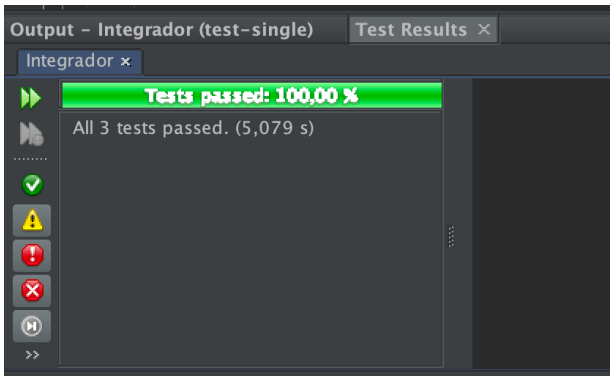
Y por ultimo este integrador cuenta con una clase Test, esta clase Test va a ser la encargada de corregir nuestro integrador. Esta clase consta de una serie de llamados a las funciones con resultados esperados.

```java
public class EggTest {

    private Practica pr;

    public EggTest() {
        this.pr = new Practica();
    }

    @Test
    public void prisioneroDulceTest() {
        assertEquals(3, pr.prisioneroDulce(0, 10, 6));
        assertEquals(2, pr.prisioneroDulce(7, 4, 8));
        assertEquals(1, pr.prisioneroDulce(1, 5, 2));
        assertEquals(0, pr.prisioneroDulce(0, 10, 1));
        assertEquals(39, pr.prisioneroDulce(455, 1585, 500));
    }
```

Nuestra tarea es hacer los problemas(funciones) y lograr que cuando corramos la clase EggTest y esta llame a nuestras funciones, se devuelvan los resultados esperados. La manear que funcionan estos llamados, es le pasa a la función ciertos valores y espera ciertos resultados.

Para saber el resultado de nuestros problemas vamos a revisar nuestra clase EggTest, esta se encuentra en los Test Packages. Para correr la clase Test, deberemos, entrar a la clase y dentro de ella hacer click derecho y darle a la opción Test File.

Esto nos mostrará una pantallita con el porcentaje de Tests pasados y nos mostrará en donde nos equivocamos si es que tenemos un resultado que la clase no esperaba.



Si no aparece esta pestaña deberemos irnos a la opción en Netbeans Window -> IDE Tools y dentro de IDE Tools la pestaña que dice Tests Results. Ahí veremos si nuestros tests dieron los resultados esperados.

Debajo de este pdf, están las consignas y todos los resultados esperados según cada ejercicio, entonces se recomienda que prueben en el main, pasando a la función los valores que dicen los resultados esperados y ver si les dio el resultado que pide dicho inciso. Cuando tengamos todo hecho, ahí corremos nuestro archivo test para corroborar.

Por ejemplo si nosotros queremos que una función devuelva true cuando ingresamos el numero 10, le pasamos el numero 10 a la función y vemos que nos devolvió. Si nos devolvió true, tomamos ese inciso como terminado y pasamos al siguiente.

CONSIGNAS

Función numeroCapicua():

La función recibirá un numero x y deberá determinar si es capicúa o no. Este deberá devolver verdadero(true) si es capicúa y falso(false) si no lo es. Además deberemos contemplar los siguientes escenarios:

- Contemplar que el numero que llega puede ser negativo.
- Contemplar que el numero que llega puede ser de un digito, si es así debe devolver true.
- Contemplar que el numero que llega puede ser null, si es así debe devolver false.

Resultados esperados:

Si se ingresa el numero 123454321 deberá devolver true.
Si se ingresa el numero -123454321 deberá devolver true.
Si se ingresa el numero 2 deberá devolver true.
Si se ingresa el numero 0 deberá devolver true.
Si se ingresa null deberá devolver false.
Si se ingresa el numero 231 deberá devolver false.
Si se ingresa el numero 123 deberá devolver true.

Función prisioneroDulce():

Estamos en caramelolandia, donde estan los peores ladrones de dulces. Una vez al mes, se sienta una n cantidad de presos en ronda, contemplando al preso que inicia la ronda, como el preso 0.

A los presos se les da una m cantidad de caramelos, estos caramelos se repartirán de uno en uno a cada preso, contemplando que se comenzaran a repartir los caramelos desde el primer preso (inicio). Se repartirán los caramelos hasta que no queden más y el ultimo caramelo en repartirse estará podrido, determinar a que preso, según su posición en la ronda le tocara el caramelo podrido.

Esta función recibe tres variables:

- inicio: está variable será el numero del preso que inicia la ronda.
- cantidadCaramelos: está variable será el numero de caramelos que se les da a los presos.
- cantidadPresos: está variable será el numero de presos que componen a la ronda.

Resultados esperados:

- Sí se inicia con el preso 0, con 10 caramelos y 6 presos el resultado esperado es 3.
- Sí se inicia con el preso 7, con 4 caramelos y 8 presos el resultado esperado es 2.
- Sí se inicia con el preso 1, con 5 caramelos y 2 presos el resultado esperado es 1.
- Sí se inicia con el preso 0, con 10 caramelos y 1 preso el resultado esperado es 0.
- Sí se inicia con el preso 455, con 1585 caramelos y 500 presos el resultado esperado es 39.

Función mediasAmigas():

En un universo paralelo, donde los habitantes son medias, existe un crucero de medias donde se sube una lista de medias. Esta lista de tripulantes del crucero es una Collection de letras.

Lo que se deberá hacer, es filtrar la lista de medias que se suben al crucero y retornar una lista que contenga los grupos de medias que si tenían pares. Esta lista final de medias pares se mostraran ordenadas de menor a mayor.

A continuación un ejemplo:

List de medias que llegan : A,B,A,B,C,A,F,Z,C,H. A,B y C tiene pares, mientras que F,Z y H no. Entonces la List que se debería retornar sería: A,B,C.

Resultados esperados:

- Si se ingresa la lista "A", "B", "C", "D", "A", "C", "D", "A". El resultado sería "A", "C", "D".
- Si se ingresa la lista "R", "R", "A", "A", "S", "S", "G", "H". El resultado sería "A", "R", "S"
- Si se ingresa la lista "R", "E", "T", "A", "P", "S", "G", "H". El resultado sería vacío porque no hay pares.

Nota: Recordemos que tenemos que correr el archivo EggTest para probar nuestros ejercicios, igualmente recomendamos ir probando con el main y viendo que resultado nos arroja.