

Passos a seguir per treballar amb GitHub:

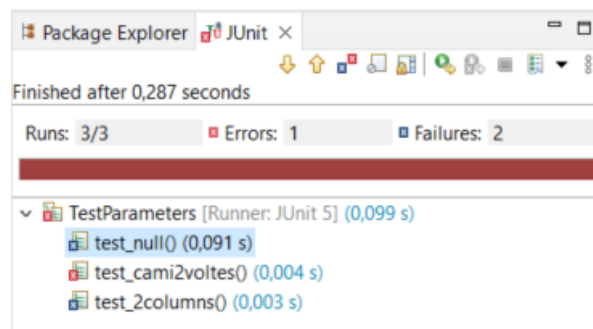
1. Seguiu la invitació a la Github Classroom que trobareu a l'aula virtual:
 - a. Tots els alumnes s'han de crear un compte, si no en teniu.
 - b. Un alumne del grup entra a la classroom i crea el grup, si us plau, intenteu posar el mateix nom que la **Parella XX** de l'aula virtual i amb els mateixos companys.
 - c. Els altres alumnes entres a la classroom i sel·leccionen el grup creat.
2. Tots els alumnes han de clonar el directori compartit al seu ordinador, podeu utilitzar el programa que vulgueu, explico com fer-ho des de Eclipse:
 - a. Primer cal afegir Egit des de Help > Eclipse Marketplace
 - b. Obrir la finestra per veure el repositori: Window > Show View > Git Repositories
 - c. Clone a Git repository
 - d. Import project
3. Feu un exemple per veure com funciona la gestió de versions:

Un estudiant fa una modificació dins del codi i després un commit and push:

 - a. Per exemple, escriu un comentari.
 - b. Obrir la finestra per gestionar el repositori: Window > Show View > Git Staging
 - c. Escriu un Commit Message "Primer modificació"
 - d. Botó Commit and Push

Els altres estudiants fan un Pull i veuran els canvis que han fet:

- e. Botó dret al repositori dins de la finestra Git Repositories
 - f. Pull
4. (Només cal fer aquest pas si no es troba la llibreria JUnit) Totes les comprovacions es realitzen amb la llibreria JUnit, segueix el pas 3 on sigui un altre estudiant qui fa el commit and push, passos previs:
 - a. Propiety > Java Bild Path > Libraries > Add library...
 - b. JUnit
5. Cal executar amb Run as.. > JUnit Test i s'obrirà la següent pantalla indicant quines de les comprovacions han fallat.



Pràctica 1 : Testing

Important: Al document README cal escriure el nom del grup Parella XX de l'aula virtual, i cada alumne ha d'escriure el seu nom complet i el seu nom d'usuari de GitHub.

Important: Caldrà adjuntar un document amb les solucions dels punts 1, 2 i 3. Si voleu ho podeu fer a mà.

Al servidor de GitHub trobareu el projecte Eclipse on hi ha la classe `CalculDescompte` amb un sol mètode estàtic anomenat `getDescompte()`. Es demana testejar el mètode aplicant la tècnica del Conjunt Bàsic descrita a classe, concretament caldrà lliurar:

```
3 public class CalculDescompte {
4
5     // la matriu productes té dos columnes: TIPUS i PREU
6     // cada fila representa un producte
7     // Si el tipus és 1 i el preu és superior a 20,00 euros aplicarem un 5% de descompte
8     // si el tipus és 2 i el preu és superior a 50,00 euros aplicarem un 10% de descompte
9     public static double getDescompte(double[][] productes)
10    {
11        double preu, percentatge, descompte = 0;
12        int tipus;
13        for( int index = 1; index <= productes.length; index++)
14        {
15            tipus = (int) productes[index][0];
16            preu = productes[index][1];
17            percentatge = 0;
18            if(tipus == 1 && preu > 20)
19                percentatge = 0.05;
20            if(tipus == 2 && preu > 50)
21                percentatge = 0.10;
22            descompte += preu * percentatge;
23        }
24        return descompte;
25    }
26 }
```

1. Trobar tots els camins lògics d'execució des de la línia 11 fins a la 24.
 - a. Dibuixa el graf seguint la següent imatge.
 - b. Calcula la complexitat ciclomàtica.
 - c. Troba totes els caminis segons el graf.
2. Escollir un Conjunt Casos de Prova amb la màxima cobertura de proves.
 - a. Indica els possibles conjunts bàsics (CB), per fer-ho selecciona el mínim de camins linealment independents que donin obertura màxima, el nombre de camins serà inferior a la complexitat ciclomàtica.
 - b. Indica tots els caminis de zero voltes (COV).

- c. Selecciona el conjunt de casos de prova (CCP), que estarà format per la unió d'un conjunt CB i el conjunt COV.
3. Dissenyar els casos de prova pel CCP: Per cada camí seleccionat indica quin són els paràmetres d'entrada i quin serà la sortida esperada segons la descripció (però no miris en detall el codi).
4. Programar els casos de prova dissenyats en el punt anterior **creant una nova classe que sigui un TestCase de JUnit 5**.
5. Executar els teus testos unitaris:
 - a. En cas d'èxit (fail = s'ha trobat alguna errada) o en cas d'excepció al codi(=error), depurar l'error trobat modificant el codi del mètode subministrat (**cal que documenteu l'error i la modificació feta, indica amb quin camí has trobat l'error**).
 - b. Tornar el punt 5 fins que no hi hagi errors.
6. També cal modificar el programa per tal que compleixi els tests unitaris de la classe TestParameters que comproven els paràmetres d'entrada. Crea un TestSuite que agrupi tots dos TestCase.
 - a. En el cas de les excepcions del paràmetre d'entrada, cal comprovar les condicions i llançar un IllegalArgumentException amb el missatge indicat.
 - b. En el cas del camí de 2 voltes, si amb les teves proves no has trobat tots els errors, cal comprovar-ho ara.