

Testeig mètodes de les classes

A continuació mostrem la documentació dels tests de tots els mètodes de les classes del sistema. Tota la documentació es troba també al codi en format comentari *Javadoc* a sobre de cada mètode.

Interpret

De la classe `Interpret` s'han realitzat n tests i tots ells passen correctament.

```
andreuorensanz src % make TestInterpret
javac -d ../../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../../lib/junit-4.12.jar:../../lib/hamcrest-core-1.3.jar -d ../../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java ./test/*.java ./test/types/*.java ./test/functions/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../../lib/junit-4.12.jar:../../lib/hamcrest-core-1.3.jar:../../EXE/out/ org.junit.runner.JUnit4 test.functions.TestInterpret
JUnit version 4.12
.....
Time: 0,078
OK (43 tests)
```

testInterpretAsSingleton()

- * Objecte de la prova: Test del mètode `testInterpretAsSingleton` de la classe `TestInterpret`.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es comprova que només existeixi una instància de `Interpret` ja que es tracta d'una classe singleton.
- * Operativa: Es defineixen dues instàncies de `Interpret` i es comprova que aquestes siguin la mateixa ja que només en pot existir una.

getTypeInt()

- * Objecte de la prova: Test del mètode `getType` de la classe `Interpret`.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es comprova que l'entrada introduïda manualment sigui de tipus `INT`.
- * Operativa: En aquest test es comprova que el string passat per paràmetre a la funció `getType` sigui de tipus `INT`, definit al enum `TType`.

getTypeDouble()

- * Objecte de la prova: Test del mètode `getType` de la classe `Interpret`.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es comprova que l'entrada introduïda manualment sigui de tipus `DOUBLE`.
- * Operativa: En aquest test es comprova que el string passat per paràmetre a la funció `getType` sigui de tipus `DOUBLE`, definit al enum `TType`.

getTypeDate()

- * Objecte de la prova: Test del mètode `getType` de la classe `Interpret`.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es comprova que l'entrada introduïda manualment sigui de tipus `DATE`.

* Operativa: En aquest test es comprova que el string passat per paràmetre a la funció getType sigui de tipus DATE, definit al enum TType.

getTypeFunction()

* Objecte de la prova: Test del mètode getType de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es comprova que l'entrada introduïda manualment sigui de tipus FUNCTION.

* Operativa: En aquest test es comprova que el string passat per paràmetre a la funció getType sigui de tipus FUNCTION, definit al enum TType.

getTypeString()

* Objecte de la prova: Test del mètode getType de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es comprova que l'entrada introduïda manualment sigui de tipus STRING.

* Operativa: En aquest test es comprova que el string passat per paràmetre a la funció getType sigui de tipus STRING, definit al enum TType.

getDouble1()

* Objecte de la prova: Test del mètode getDouble de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa a Double el String passat per paràmetre

* Operativa: En aquest test es comprova que el String es pot parsejar a tipus Double i es parseja si així és el cas.

* Es dóna un marge de 0,0001 entre el double entrat manualment i el parsejat amb la funció getDouble.

getDouble2()

* Objecte de la prova: Test del mètode getDouble de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa a Double el String passat per paràmetre

* Operativa: En aquest test es comprova que el String es no pot parsejar a tipus Double ja que és un dígit i retorna -1.

* Es dóna un marge de 0,0001 entre el double entrat manualment i el parsejat amb la funció getDouble

getInt1()

* Objecte de la prova: Test del mètode getInt de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa a Integer el String passat per paràmetre

* Operativa: En aquest test es comprova que el String es pot parsejar a tipus Integer i es parseja si així és el cas.

getInt2()

* Objecte de la prova: Test del mètode getInt de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

- * Valors estudiats: Estratègia caixa gris. Es passa a Integer el String passat per paràmetre
- * Operativa: En aquest test es comprova que el String es no pot parsejar a tipus Integer ja que és un dígit i retorna -1.

getData1()

- * Objecte de la prova: Test del mètode getData de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa a Date el String passat per paràmetre
- * Operativa: En aquest test es comprova que el String es pot parsejar a tipus LocalDate i es parseja si així és el cas.
- * Es defineix el format que utilitzem per als tipus Date amb el DateTimeFormatter i es comprova que es pot parsejar amb el mateix format

getData2()

- * Objecte de la prova: Test del mètode getData de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa a Date el String passat per paràmetre
- * Operativa: En aquest test es comprova que el String no es pot parsejar a tipus LocalDate amb un format que hem definit i retorna la data actual.

callFunctionArrodonir1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció ARRODONIR i es passa un dígit negatiu com a valor.
- * Seguidament es comprova que la funció és calculada i el valor és el correcte, que és el que introduïm manualment.

callFunctionArrodonir2()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció ARRODONIR i es passa un dígit positiu com a valor.
- * Seguidament es comprova que la funció és calculada i el valor és el correcte, que és el que introduïm manualment.

callFunctionTruncar1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció TRUNCAR i es passa un dígit amb decimals com a valor. Seguidament es comprova que la funció és calculada i el valor és el correcte, que és el que introduïm manualment.

callFunctionDecimalABinari1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció DEC2BIN i es passa un dígit com a valor. Seguidament es comprova que la funció és calculada i que es passa el dígit del paràmetre a Binari, i es comprova que el valor és el correcte.

callFunctionBinariADecimal1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció BIN2DEC i es passa un dígit en binari com a valor. Seguidament es comprova que la funció és calculada i que es passa el dígit en binari del paràmetre a decimal, i es comprova que el valor és el correcte.

callFunctionDecimalAHexadecimal1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció DEC2HEX i es passa un dígit com a valor. Seguidament es comprova que la funció és calculada i que es passa el dígit del paràmetre a Hexadecimal, i es comprova que el valor és el correcte.

callFunctionAbsolut1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció ABS i es passa un dígit negatiu com a valor. Seguidament es comprova que la funció és calculada i que es passa el dígit del paràmetre a positiu, i es comprova que el valor és el correcte.

callFunctionPotencia1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció POTENCIA i es passa un dígit que és la base i un altre que és l'exponent. Seguidament es comprova que la funció és

calculada i que es calcula la potència a partir de la base i l'exponent que es passen per paràmetre, i es comprova que el valor és el correcte.

callFunctionArrel1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció ARREL i es passa un dígit que és la base i un altre que és el grau de l'arrel. Seguidament es comprova que la funció és calculada i que es calcula l'arrel a partir de la base i el grau de l'arrel que es passen per paràmetre, i es comprova que el valor és el correcte.

callFunctionFactorial1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció FACTORIAL i es passa un dígit en integer per paràmetre. Seguidament es comprova que la funció és calculada i que es calcula el factorial del valor que es passa per paràmetre, i es comprova que el valor és el correcte.

callFunctionObteDia1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció OBTEDIA i es passa un string representant una data en format dd/MM/yyyy per paràmetre. Seguidament es comprova que la funció és calculada parsejant el string del paràmetre a LocalDate i en retorna el dia. Per últim, es comprova que el valor és el correcte.

callFunctionObteMes1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció OBTEMES i es passa un string representant una data en format dd/MM/yyyy per paràmetre. Seguidament es comprova que la funció és calculada parsejant el string del paràmetre a LocalDate i en retorna el número del mes. Per últim, es comprova que el valor és el correcte.

callFunctionObteAny1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció OBTEANY i es passa un string representant una data en format dd/MM/yyyy per paràmetre. Seguidament es comprova que la funció és calculada parsejant el string del paràmetre a LocalDate i en retorna l'any. Per últim, es comprova que el valor és el correcte.

callFunctionObteDiaSetmana1()

* Objecte de la prova: Test del mètode callFunction de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció OBTEDIASETMANA i es passa un string representant una data en format dd/MM/yyyy per paràmetre. Seguidament es comprova que la funció és calculada parsejant el string del paràmetre a LocalDate i en retorna el nom del dia de la setmana en dues lletres: dl, dm, dc, dj, dv, ds, dg. Per últim, es comprova que el valor és el correcte.

callFunctionAMajuscula1()

* Objecte de la prova: Test del mètode callFunction de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció MAJUS i es passa un string per paràmetre. Seguidament es comprova que la funció és calculada i passa tots els caràcters del string a majúscules. Per últim, es comprova que el valor és el correcte.

callFunctionReemplaca1()

* Objecte de la prova: Test del mètode callFunction de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció REEMPLACA i es passa un string d'entrada, un string amb el substring es vol substituir de l'entrada i un string amb el qual es vol substituir. Seguidament es comprova que la funció és calculada i reemplaça el substring del string entrada amb el string del 3r paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionReemplaca2()

* Objecte de la prova: Test del mètode callFunction de la classe Interpret

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.

* Operativa: En aquest test es comprova que el String es tracta de la funció REEMPLACA i es passa un string d'entrada amb més de dues paraules, un string amb el substring del que es vol substituir de l'entrada i un string amb el qual es vol substituir.

* Seguidament es comprova que la funció és calculada i reemplaça el substring del string entrada amb el string del 3r paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionTamany1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció TAMANY i es passa un string per paràmetre.
- * Seguidament es comprova que la funció és calculada i es retorna un dígit que representa el número de caràcters que té el string passat per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionParaules1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String amb l'operació que es vol calcular i amb quins valors.
- * Operativa: En aquest test es comprova que el String es tracta de la funció NUMPARAULES i es passa un string per paràmetre. Seguidament es comprova que la funció és calculada i es retorna un dígit que representa el número de substrings separat per espais (paraules) hi ha al string passat per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionResta1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos dígits.
- * Operativa: En aquest test es comprova que el String es tracta de la funció RESTA i es passen dos dígits per paràmetre. Seguidament es comprova que la funció és calculada i es retorna el valor de restar el primer dígit pel segon dels que s'han passat per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionDivisio1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos dígits.
- * Operativa: En aquest test es comprova que el String es tracta de la funció DIVISIÓ i es passen dos dígits per paràmetre. Seguidament es comprova que la funció és calculada i es retorna el valor de dividir el primer dígit entre segon dels que s'han passat per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionSuma1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígits.

* Operativa: En aquest test es comprova que el String es tracta de la funció SUMA i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és calculada i es retorna el valor de sumar tots els dígitos passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionVariancia1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígitos.
- * Operativa: En aquest test es comprova que el String es tracta de la funció VARIANCIA i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és calculada i es retorna el valor de realitzar la variància de tots els dígitos passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionDesviacioEstandar1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígitos.
- * Operativa: En aquest test es comprova que el String es tracta de la funció DESVEST i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és calculada i es retorna el valor de realitzar la desviació estàndar de tots els dígitos passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionMultiplicacio1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígitos.
- * Operativa: En aquest test es comprova que el String es tracta de la funció MULTIPLICACIO i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és calculada i es retorna el valor de multiplicar tots els dígitos passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionMitjana1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígitos.
- * Operativa: En aquest test es comprova que el String es tracta de la funció MITJANA i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és calculada i es retorna el valor de fer la mitjana de tots els dígitos passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionMediana1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada una cadena de dígitos.
- * Operativa: En aquest test es comprova que el String es tracta de la funció MEDIANA i una cadena de dígitos com a paràmetres (mínim dos). Seguidament es comprova que la funció és

calculada i es retorna el valor de fer la mediana de tots els dígit passats per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionCorrelacioPearson1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dues cadenes de dígit.
- * Operativa: En aquest test es comprova que el String es tracta de la funció PEARSON i dues cadenes de dígit com a paràmetres (amb mínim dos dígit cada cadena). Seguidament es comprova que la funció és calculada i es retorna el valor de fer la correlació de Pearson de tots els dígit de les cadenes passades per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionCorrelacioCovariancia1()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dues cadenes de dígit.
- * Operativa: En aquest test es comprova que el String es tracta de la funció COVARIANCIA i dues cadenes de dígit com a paràmetres (amb mínim dos dígit cada cadena). Seguidament es comprova que la funció és calculada i es retorna el valor de fer la covariància de tots els dígit de les cadenes passades per paràmetre. Per últim, es comprova que el valor és el correcte.

callFunctionCorrelacioCovariancia2()

- * Objecte de la prova: Test del mètode callFunction de la classe Interpret
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dues cadenes de dígit.
- * Operativa: En aquest test es comprova que el String es tracta de la funció COVARIANCIA i dues cadenes de dígit com a paràmetres (amb mínim dos dígit cada cadena). Es comproven resultats amb molts decimals. Seguidament es comprova que la funció és calculada i es retorna el valor de fer la covariància de tots els dígit de les cadenes passades per paràmetre. Per últim, es comprova que el valor és el correcte.

Operacions full de càlcul

Totes les operacions del full de càlcul es realitzen a mètodes de classes independents, és a dir, hi ha una classe per operació. Els tests de tots els mètodes de totes les classes s'executen a la vegada amb un fitxer que l'anomenem *TestSuiteFuncions*. A continuació veiem el resultat de tots els tests unitaris:

```
andreuorensanz src % make TestSuiteFuncions
javac -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java ./test/*.java ./test/types/*.java ./test/functions/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar:../EXE/out/ org.junit.runner.JUnit4 org.junit.runner.JUnit4Core test.functions.TestSuiteFuncions
JUnit version 4.12
.....
Time: 0,073

OK (57 tests)
```

Absolut

absolutNegatiu()

- * Objecte de la prova: Test del mètode absolut de la classe Absolut
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que amb un dígit negatiu, l'operació absolut es calcula correctament. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Absolut.

absolutPositiu()

- * Objecte de la prova: Test del mètode absolut de la classe Absolut
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que amb un dígit positiu, l'operació absolut es calcula correctament. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Absolut.

aMajuscula

aMajusculaMix()

- * Objecte de la prova: Test del mètode aMajuscula() de la classe aMajuscula
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String.
- * Operativa: En aquest test es comprova que amb un String on els caràcters estan barrejats entre majúscules i minúscules, l'operació aMajuscula es calcula correctament i tots els caràcters es passen a majúscules. Primer es fa un setter del paràmetre en String i seguidament es calcula l'operació d'aMajuscula.

aMajusculaMin()

- * Objecte de la prova: Test del mètode aMajuscula() de la classe aMajuscula
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String.
- * Operativa: En aquest test es comprova que amb un String on tots els caràcters estan en minúscules, l'operació aMajuscula es calcula correctament i tots els caràcters es passen a majúscules. Primer es fa un setter del paràmetre en String i seguidament es calcula l'operació d'aMajuscula.

aMajusculaMaj()

- * Objecte de la prova: Test del mètode aMajuscula() de la classe aMajuscula
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String.
- * Operativa: En aquest test es comprova que amb un String on tots els caràcters estan en majúscules, l'operació aMajuscula es calcula correctament i tots els caràcters es passen a majúscules. Primer es fa un setter del paràmetre en String i seguidament es calcula l'operació d'aMajuscula.

Arrel

arrelExcept()

- * Objecte de la prova: Test del mètode arrel() de la classe Arrel
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un dígit base en negatiu i un grau de l'arrel, no es calcula correctament l'operació i es llença l'excepció NegativeRootException. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació d'Arrel.

arrel()

- * Objecte de la prova: Test del mètode arrel() de la classe Arrel
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un dígit base en positiu i un grau de l'arrel, es calcula correctament l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació d'Arrel.

Arrodonir

arrodonirPositiu7()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu acabat en un decimal major a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

arrodonirPositiu5()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu acabat en un decimal igual a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

arrodonirPositiu3()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu acabat en un decimal menor a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

arrodonirNegatiu7()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.

* Operativa: En aquest test es comprova que a partir d'un dígit negatiu acabat en un decimal major a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

arrodonirNegatiu5()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit negatiu acabat en un decimal igual a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

arrodonirNegatiu3()

- * Objecte de la prova: Test del mètode arrodonir() de la classe Arrodonir
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit negatiu acabat en un decimal menor a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació d'Arrodonir.

BinariADecimal

binariADecimalPositiu()

- * Objecte de la prova: Test del mètode binariADecimal() de la classe BinariADecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Binari.
- * Operativa: En aquest test es comprova que a partir d'un dígit Binari positiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Binari i seguidament es calcula l'operació de BinariADecimal.

binariADecimalNegatiu()

- * Objecte de la prova: Test del mètode binariADecimal() de la classe BinariADecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Binari.
- * Operativa: En aquest test es comprova que a partir d'un dígit Binari negatiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Binari i seguidament es calcula l'operació de BinariADecimal.

CorrelacioPearson

correlacioPearson1()

- * Objecte de la prova: Test del mètode correlacioPearson() de la classe CorrelacioPearson.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de CorrelacioPearson i es comprova el correcte funcionament.

correlacioPearson2()

- * Objecte de la prova: Test del mètode correlacioPearson() de la classe CorrelacioPearson.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació amb un resultat negatiu. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de CorrelacioPearson i es comprova el correcte funcionament.

Covariancia

covarianca1()

- * Objecte de la prova: Test del mètode covarianca() de la classe Covariancia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de Covariancia i es comprova el correcte funcionament.

covarianca2()

- * Objecte de la prova: Test del mètode covarianca() de la classe Covariancia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació amb un resultat amb molts decimals. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de Covariancia i es comprova el correcte funcionament.

covarianca3()

- * Objecte de la prova: Test del mètode covarianca() de la classe Covariancia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double (amb dígit positius i negatius), es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de Covariancia i es comprova el correcte funcionament.

DecimalABinari

decimalABinariPositiu()

- * Objecte de la prova: Test del mètode decimalABinari() de la classe DecimalABinari.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un dígit enter.

* Operativa: En aquest test es comprova que a partir d'un dígit enter positiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de BinariADecimal.

decimalABinariNegatiu()

- * Objecte de la prova: Test del mètode decimalABinari() de la classe DecimalABinari.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un dígit enter.
- * Operativa: En aquest test es comprova que a partir d'un dígit enter negatiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de DecimalABinari.

DecimalAHexadecimal

decimalAHexadecimalPositiu()

- * Objecte de la prova: Test del mètode decimalAHexadecimal() de la classe DecimalAHexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un dígit enter.
- * Operativa: En aquest test es comprova que a partir d'un dígit enter positiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de DecimalAHexadecimal.

decimalAHexadecimalNegatiu()

- * Objecte de la prova: Test del mètode decimalAHexadecimal() de la classe DecimalAHexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un dígit enter.
- * Operativa: En aquest test es comprova que a partir d'un dígit enter negatiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de DecimalAHexadecimal.

Divisio

divisioPosPosPeriod()

- * Objecte de la prova: Test del mètode divisio() de la classe Divisio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir de dos Doubles positius on el valor de la divisió no és exacte, es calcula correctament l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Divisio.

divisioPosNegPeriod()

- * Objecte de la prova: Test del mètode divisio() de la classe Divisio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un Double positiu com a dividend i un Double negatiu com a divisor on el valor de la divisió no és exacte, es calcula correctament

l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Divisio.

divisioNegPosPeriod()

- * Objecte de la prova: Test del mètode divisio() de la classe Divisio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un Double negatiu com a dividend i un Double positiu com a divisor on el valor de la divisió no és exacte, es calcula correctament l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Divisio.

divisioNegNegPeriod()

- * Objecte de la prova: Test del mètode divisio() de la classe Divisio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un Double negatiu com a dividend i un Double negatiu com a divisor on el valor de la divisió no és exacte, es calcula correctament l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Divisio.

divisioExacte()

- * Objecte de la prova: Test del mètode divisio() de la classe Divisio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir de dos Doubles positius on el valor de la divisió és exacte, es calcula correctament l'operació. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Divisio.

DoubleAInt

doubleAIntPosMaj()

- * Objecte de la prova: Test del mètode doubleAInt() de la classe doubleAInt.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu acabat en un decimal major a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de doubleAInt.

doubleAIntPosMen()

- * Objecte de la prova: Test del mètode doubleAInt() de la classe doubleAInt.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu acabat en un decimal menor a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de doubleAInt.

doubleAIntNegMaj()

- * Objecte de la prova: Test del mètode doubleAInt() de la classe doubleAInt.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit negatiu acabat en un decimal major a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de doubleAInt.

doubleAIntNegMen()

- * Objecte de la prova: Test del mètode doubleAInt() de la classe doubleAInt.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit negatiu acabat en un decimal menor a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de doubleAInt.

Factorial

factorialPositiu()

- * Objecte de la prova: Test del mètode factorial() de la classe Factorial.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un enter.
- * Operativa: En aquest test es comprova que a partir d'un dígit positiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de Factorial.

factorialNegatiu()

- * Objecte de la prova: Test del mètode factorial() de la classe Factorial.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un enter.
- * Operativa: En aquest test es comprova que a partir d'un dígit negatiu, no es calcula l'operació i es llença l'excepció NegativeFactorialException. Primer es fa un setter del paràmetre en Integer i seguidament es calcula l'operació de Factorial, de manera que salta l'excepció.

HexadecimalADecimal

hexadecimalADecimalPositiu()

- * Objecte de la prova: Test del mètode hexadecimalADecimal() de la classe hexadecimalADecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Hexadecimal.
- * Operativa: En aquest test es comprova que a partir d'un dígit Hexadecimal positiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Hexadecimal i seguidament es calcula l'operació de hexadecimalADecimal.

hexadecimalADecimalNegatiu()

- * Objecte de la prova: Test del mètode hexadecimalADecimal() de la classe hexadecimalADecimal.

- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Hexadecimal.
- * Operativa: En aquest test es comprova que a partir d'un dígit Hexadecimal negatiu, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Hexadecimal i seguidament es calcula l'operació de hexadecimalADecimal.

Mediana

mediana()

- * Objecte de la prova: Test del mètode mediana() de la classe Mediana.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de Mediana i es comprova el correcte funcionament.

Mitjana

mitjana()

- * Objecte de la prova: Test del mètode mitjana() de la classe Mitjana.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors i finalment es calcula l'operació de Mitjana i es comprova el correcte funcionament.

Multiplacacio

multiplicacio()

- * Objecte de la prova: Test del mètode multiplicacio() de la classe Multiplicacio.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors. Finalment es calcula l'operació de Multiplicacio i es comprova el correcte funcionament.

Variancia

variancia1()

- * Objecte de la prova: Test del mètode variancia() de la classe Variancia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir d'un vector de tipus Double, es calcula correctament l'operació amb un resultat amb decimals. Primer es crea un vector de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter del vector. Finalment es calcula l'operació de Variancia i es comprova el correcte funcionament.

variancia2()

- * Objecte de la prova: Test del mètode variancia() de la classe Variancia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació amb un resultat amb decimals. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors. Finalment es calcula l'operació de Variancia i es comprova el correcte funcionament.

DesviacioEstandar

desviacioEstandar1()

- * Objecte de la prova: Test del mètode desviacioEstandar() de la classe DesviacioEstandar.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un vector de tipus Double.
- * Operativa: En aquest test es comprova que a partir d'un vector de tipus Double, es calcula correctament l'operació amb un resultat amb decimals. Primer es crea un vector de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter del vector. Finalment es calcula l'operació de DesviacioEstandar i es comprova el correcte funcionament.

desviacioEstandar2()

- * Objecte de la prova: Test del mètode desviacioEstandar() de la classe DesviacioEstandar.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un vector de tipus Double.
- * Operativa: En aquest test es comprova que a partir d'un vector de tipus Double, es calcula correctament l'operació amb un resultat amb decimals. Primer es crea un vector de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter del vector. Finalment es calcula l'operació de DesviacioEstandar i es comprova el correcte funcionament.

NumeroParaules

numeroParaules()

- * Objecte de la prova: Test del mètode numeroParaules() de la classe numeroParaules
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String.
- * Operativa: En aquest test es comprova quantes paraules (substrings separats per un espai) hi ha a un String. Primer es fa un setter del paràmetre en String i seguidament es calcula l'operació d'aMajuscula i se'n comprova el correcte funcionament.

ObteAny

obteAny()

- * Objecte de la prova: Test del mètode obteAny() de la classe obteAny.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un LocalDate.

* Operativa: En aquest test es calcula l'any d'una data (en LocalDate) entrada per paràmetre i es comprova que el resultat és el correcte. Primer es fa un setter del paràmetre en LocalDate i seguidament es calcula l'operació d'obteAny.

ObteDia

obteDia()

- * Objecte de la prova: Test del mètode obteDia() de la classe obteDia.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un LocalDate.
- * Operativa: En aquest test es calcula el dia d'una data (en LocalDate) entrada per paràmetre i es comprova que el resultat és el correcte. Primer es fa un setter del paràmetre en LocalDate i seguidament es calcula l'operació d'obteDia.

ObteDiaSetmana

obteDiaSetmana()

- * Objecte de la prova: Test del mètode obteDiaSetmana() de la classe obteDiaSetmana.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un LocalDate.
- * Operativa: En aquest test es calcula el dia de la setmana d'una data (en LocalDate) entrada per paràmetre i es comprova que el dia de la setmana és el correcte. Primer es fa un setter del paràmetre en LocalDate i seguidament es calcula l'operació d'obteDiaSetmana.

ObteMes

obteMes()

- * Objecte de la prova: Test del mètode obteMes() de la classe obteMes.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un LocalDate.
- * Operativa: En aquest test es calcula el mes d'una data (en LocalDate) entrada per paràmetre i es comprova que el resultat és el correcte. Primer es fa un setter del paràmetre en LocalDate i seguidament es calcula l'operació d'obteMes.

Potencia

potencia()

- * Objecte de la prova: Test del mètode potencia() de la classe Potencia
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir d'un dígit que representa la base i un que representa l'exponent, es calcula correctament l'operació potencia. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Potencia.

potenciaGran()

- * Objecte de la prova: Test del mètode potencia() de la classe Potencia
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.

* Operativa: En aquest test es comprova que a partir d'un dígit que representa la base i un que representa un exponent relativament gran, es calcula correctament l'operació potencia. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Potencia.

Reemplaca

reemplaca()

- * Objecte de la prova: Test del mètode reemplaca() de la classe Reemplaca
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada tres Strings.
- * Operativa: En aquest test es comprova que amb un String d'entrada, un String que representa un substring de l'entrada que es vol reemplaçar i un String amb el qual es vol reemplaçar aquest substring. Primer es fa un setter dels tres Strings passats per paràmetre i seguidament es calcula l'operació de Reemplaca.

Resta

resta()

- * Objecte de la prova: Test del mètode resta() de la classe Resta.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos Doubles.
- * Operativa: En aquest test es comprova que a partir de dos Doubles positius, es calcula correctament l'operació resta entre els dos dígit. Primer es fa un setter dels paràmetres en Double i seguidament es calcula l'operació de Resta i s'en comprova el correcte resultat.

Suma

suma()

- * Objecte de la prova: Test del mètode suma() de la classe Suma.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada dos vectors de tipus Double.
- * Operativa: En aquest test es comprova que a partir de dos vectors de tipus Double, es calcula correctament l'operació. Primer es creen dos vectors de Double i s'afegeixen els valors amb el qual es vol fer el test de l'operació i, seguidament, es fa un setter dels dos vectors. Finalment es calcula l'operació de Suma i es comprova el correcte funcionament.

TamanyText

tamanyText()

- * Objecte de la prova: Test del mètode tamanyText() de la classe tamanyText.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un String.
- * Operativa: En aquest test es comprova el número de caràcters que conté el String passat per paràmetre. Primer es fa un setter del paràmetre en String i seguidament es calcula l'operació de tamanyText i se'n comprova el correcte funcionament.

Truncar

truncarMenor()

- * Objecte de la prova: Test del mètode truncar() de la classe Truncar

- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit acabat en un decimal menor a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de Truncar.

truncarMajor()

- * Objecte de la prova: Test del mètode truncar() de la classe Truncar
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es passa per entrada un Double.
- * Operativa: En aquest test es comprova que a partir d'un dígit acabat en un decimal major a .5, es calcula correctament l'operació. Primer es fa un setter del paràmetre en Double i seguidament es calcula l'operació de Truncar.

Tipus

Les classes que defineixen nous tipus de dades es comproven també amb un fitxer *TestSuite*.

```

> make TestSuiteTipus
javac -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java ./test/*.java ./test/types/*.java ./test/functions/*.java
Note: ./test/types/TestPair.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar:../EXE/out/ org.junit.runner.JUnit4 org.junit.runner.RunWith org.junit.runners.JUnit4 TestSuiteTipus
JUnit version 4.12
.....
Time: 0.028
OK (31 tests)

```

Binari

Binari()

- * Objecte de la prova: Test del mètode Binari() de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Binari amb valor "0" predeterminadament. Primer es crea un objecte binari sense paràmetres, es crida al mètode getValor() i es comprova que el resultat és el correcte.

BinariStringCorrecte()

- * Objecte de la prova: Test del mètode Binari(String x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari i es passa un String per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Binari amb un String per paràmetre. Primer es crea un objecte binari amb un String per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

BinariStringIncorrecte()

- * Objecte de la prova: Test del mètode Binari(String x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari i es passa un String per paràmetre.
- * Operativa: En aquest test es comprova que no es crea correctament un objecte Binari amb un String per paràmetre i es llença l'excepció NumberFormatException. Primer es crea un objecte binari amb un String per paràmetre, es crida al mètode getValor() i es comprova que salta l'excepció

BinariIntPositiu()

- * Objecte de la prova: Test del mètode Binari(int x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari i es passa un enter per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Binari amb un Integer positiu per paràmetre. Primer es crea un objecte binari amb un Integer per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

BinariIntNegatiu()

- * Objecte de la prova: Test del mètode Binari(int x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari i es passa un enter per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Binari amb un Integer negatiu per paràmetre. Primer es crea un objecte binari amb un Integer per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

setValorStringCorrecte()

- * Objecte de la prova: Test del mètode setValor(String x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Binari el valor String passat per paràmetre. Es comprova que el valor és Binari.

setValorStringIncorrecte()

- * Objecte de la prova: Test del mètode setValor(String x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter no guarda correctament a l'atribut String intern de la classe Binari el valor String passat per paràmetre. Es comprova que el valor no és Binari.

setValorIntPositiu()

- * Objecte de la prova: Test del mètode setValor(int x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari sense paràmetres.

* Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Binari el valor Integer passat per paràmetre. Es passa el valor int a String i es comprova que el valor és binari.

setValorIntNegatiu()

- * Objecte de la prova: Test del mètode setValor(int x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Binari el valor Integer negatiu passat per paràmetre. Es passa el valor int a String i es comprova que el valor és binari.

aDecimalPositiu()

- * Objecte de la prova: Test del mètode aDecimal() de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari amb enter com a paràmetre.
- * Operativa: En aquest test es comprova que la funció passa a integer el nombre binari que s'ha creat. Es crea un objecte Binari amb un enter positiu com a paràmetre i es comprova que la funció el parseja correctament a decimal.

aDecimalNegatiu()

- * Objecte de la prova: Test del mètode aDecimal() de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari amb enter com a paràmetre.
- * Operativa: En aquest test es comprova que la funció passa a integer el nombre binari que s'ha creat. Es crea un objecte Binari amb un enter negatiu com a paràmetre i es comprova que la funció el parseja correctament a decimal.

aHexadecimal()

- * Objecte de la prova: Test del mètode aHexadecimal(int x) de la classe Binari
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Binari amb enter com a paràmetre.
- * Operativa: En aquest test es comprova que la funció passa a Hexadecimal el nombre binari que s'ha creat. Es crea un objecte Binari amb un enter com a paràmetre i es comprova que la funció el parseja correctament a hexadecimal.

Hexadecimal

Hexadecimal()

- * Objecte de la prova: Test del mètode Hexadecimal() de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Hexadecimal amb valor "0" predeterminadament. Primer es crea un objecte hexadecimal sense paràmetres, es crida al mètode getValor() i es comprova que el resultat és el correcte.

HexadecimalStringCorrecte()

- * Objecte de la prova: Test del mètode Hexadecimal(String x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal i es passa un String per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Hexadecimal amb un String per paràmetre. Primer es crea un objecte hexadecimal amb un String per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

HexadecimalStringInorrecte()

- * Objecte de la prova: Test del mètode Hexadecimal(String x) de la classe Hexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal i es passa un String per paràmetre.
- * Operativa: En aquest test es comprova que no es crea correctament un objecte Hexadecimal amb un String per paràmetre i es llença l'excepció NumberFormatException. Primer es crea un objecte hexadecimal amb un String per paràmetre, es crida al mètode getValor() i es comprova que salta l'excepció.

HexadecimalIntPositiu()

- * Objecte de la prova: Test del mètode Hexadecimal(int x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal i es passa un enter per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Hexadecimal amb un Integer positiu per paràmetre. Primer es crea un objecte hexadecimal amb un Integer per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

HexadecimalIntNegatiu()

- * Objecte de la prova: Test del mètode Hexadecimal(int x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal i es passa un enter per paràmetre.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Hexadecimal amb un Integer negatiu per paràmetre. Primer es crea un objecte hexadecimal amb un Integer per paràmetre, es crida al mètode getValor() i es comprova que el resultat concorda amb el que s'ha passat per paràmetre.

setValorStringCorrecte()

- * Objecte de la prova: Test del mètode setValor(String x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Hexadecimal el valor String passat per paràmetre. Es comprova que el valor és Hexadecimal.

setValorIncorrecte()

- * Objecte de la prova: Test del mètode setValor(String x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter no guarda correctament a l'atribut String intern de la classe Hexadecimal el valor String passat per paràmetre. Es comprova que el valor no és Hexadecimal.

setValorIntPositiu()

- * Objecte de la prova: Test del mètode setValor(int x) de la classe Hexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Hexadecimal el valor Integer passat per paràmetre. Es passa el valor int a String i es comprova que el valor és hexadecimal.

setValorIntNegatiu()

- * Objecte de la prova: Test del mètode setValor(int x) de la classe Hexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal sense paràmetres.
- * Operativa: En aquest test es comprova que la funció setter guarda correctament a l'atribut String intern de la classe Hexadecimal el valor Integer negatiu passat per paràmetre. Es passa el valor int a String i es comprova que el valor és hexadecimal.

aDecimalPositiu()

- * Objecte de la prova: Test del mètode aDecimal() de la classe Hexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal amb enter com a paràmetre.
- * Operativa: En aquest test es comprova que la funció passa a integer el nombre hexadecimal que s'ha creat. Es crea un objecte Hexadecimal amb un enter positiu com a paràmetre i es comprova que la funció el parseja correctament a decimal.

aDecimalNegatiu()

- * Objecte de la prova: Test del mètode aDecimal() de la classe Hexadecimal.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal amb enter com a paràmetre.
- * Operativa: En aquest test es comprova que la funció passa a integer el nombre hexadecimal que s'ha creat. Es crea un objecte Hexadecimal amb un enter negatiu com a paràmetre i es comprova que la funció el parseja correctament a decimal.

aBinari()

- * Objecte de la prova: Test del mètode aBinari(int x) de la classe Hexadecimal
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Hexadecimal amb enter com a paràmetre.

* Operativa: En aquest test es comprova que la funció passa a Binari el nombre hexadecimal que s'ha creat. Es crea un objecte Hexadecimal amb un enter com a paràmetre i es comprova que la funció el parseja correctament a binari.

Pair

Per comprovar el funcionament de la classe Pair, hem efectuat 7 tests.

TestConstructora()

- * Objecte de la prova: Test de la constructora de la classe Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Pair.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Pair amb valors 12, "12" predeterminadament. Primer es crea un objecte Pair i després es fa un assertEquals() dels dos elements amb first() i second()

TestSetFirst()

- * Objecte de la prova: Test del mètode setFirst() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Pair i se li canvia el primer element.
- * Operativa: En aquest test es comprova que després de fer setFirst() de 123, el .first() retornara el nou valor.

TestSetSecond()

- * Objecte de la prova: Test del mètode setSecond() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Pair i se li canvia el segon element.
- * Operativa: En aquest test es comprova que després de fer setSecond() de "123", el .second() retorna el nou valor.

TestFirst()

- * Objecte de la prova: Test del mètode first() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Pair i es comprova el primer element.
- * Operativa: En aquest test es comprova que després de de crear un nou Pair el .first() retornara el assignat a la creadora.

TestSecond()

- * Objecte de la prova: Test del mètode second() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Pair i es comprova el primer element.
- * Operativa: En aquest test es comprova que després de crear un nou Pair, el second() retornara el valor assignat a la creadora.

TestEquals()

- * Objecte de la prova: Test del mètode equals() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es creen 6 pairs per efectuar diferents proves sobre equals().
- * Operativa: En aquest test es comprova que el mètode equals() només sigui cert quan els Pairs son idèntics en tipus i valors, o si son el mateix element (pair)

TestToString()

- * Objecte de la prova: Test del mètode toString() de Pair
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Pair i es comprova el String resultatnt de toString().
- * Operativa: En aquest test es comprova que després d'executar el mètode toString(), l'string resultatnt estigui en el format adequat

Cela

Per a la classe Cela s'han efectuat 9 tests i tots d'ells han sortit correctes

```
andreuorensanz src % make TestCela
javac -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/
domain/classes/types/*.java ./main/domain/controllers/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/fu
nctions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java ./test/*.java .
/test/types/*.java ./test/functions/*.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar:../EXE/out/ org.junit.runner.JUnit4 test.TestCela
JUnit version 4.12
.....
Time: 0,012
OK (9 tests)
```

TestConstructora()

- * Objecte de la prova: Test de la constructora de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela i es comprova que el valorUsuari sigui el mateix.
- * Operativa: Creem una nova Cela amb el paràmetre "test" i comprovem que el valorUsuari sigui el mateix.

TestEsFuncio()

- * Objecte de la prova: Test del mètode EsFuncio() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela i si el que conté és o no funció.
- * Operativa: Creem una nova Cela amb el paràmetre "test" i comprovem que el contingut es o no una funció i retorna false si no ho és.

TestGetValorUsuari()

- * Objecte de la prova: Test del mètode GetValorUsuari() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela amb un String com a paràmetre
- * Operativa: Creem una nova Cela amb un paràmetre String i comprovem que el contingut del valor usuari és el mateix que l'introduït.

TestGetValorReal()

- * Objecte de la prova: Test del mètode GetValorReal() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela amb un String com a paràmetre
- * Operativa: Creem una nova Cela amb un paràmetre String i comprovem que el contingut del valor real és el mateix que l'introduït.

TestModificaValor()

- * Objecte de la prova: Test del mètode ModificaValor() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela amb un String com a paràmetre.
- * Operativa: Creem una nova Cela amb un paràmetre String i es crida al mètode ModificaValor amb un altre String. Seguidament comprovem que efectivament s'ha modificat amb el String que hem introduït.

TestSetValorReal()

- * Objecte de la prova: Test del mètode SetValorReal() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela amb un String com a paràmetre.
- * Operativa: Creem una nova Cela amb un paràmetre String i es crida al setter del valor real amb aquesta nova Cela creada. Seguidament comprovem que efectivament s'ha col·locat a el valor introduït l'atribut _valorReal i es comprova que sigui el mateix.

TestEquals()

- * Objecte de la prova: Test del mètode equals() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es creen tres objectes Cela amb un String com a paràmetre.
- * Operativa: Creem tres noves Cel·les, dues amb el mateix paràmetre String i una tercera que serà igual que la primera i comprovem que quan cridem a equals() entre la primera i la segona surt false (ja que tot i tenir el mateix valor les cel·les són diferents) i quan cridem a equals() entre la primera i la tercera surt true (perquè són la mateixa cel·la exactament).

TestToString()

- * Objecte de la prova: Test del mètode toString() de Cela
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou objecte Cela amb un String com a paràmetre

* Operativa: Creem una nova Cella amb un paràmetre String i comprovem que el string introduït és el mateix que quan cridem al mètode toString(), que simplement imprimeix el contingut de la cel·la

TestCompareTo()

- * Objecte de la prova: Test del mètode compareTo() de Cella.
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es creen 10 cel·les amb diferents valors i es comprova que el compareTo de les dues cel·les sigui l'esperat (ordenació descendent) i que si es comparen les mateixes cel·les en ordre invers, el resultat s'invertirà.
- * S'espera que torni -1 si la cel·la que executa compareTo(celComparada) es anterior a la comparada, 0 si son iguals i 1 si aquesta es posterior.
- * Operativa: El primer assertEquals() comprova que una parella de cel·les son iguals independentment de si son majúscules.
- * El segon assertEquals() comprova entre números i lletres es faci correctament, és a dir, primer les lletres i després els números
- * El tercer assertEquals() comprova que entre una parella de cel·les, els números enters es comparin adequadament, 12 anterior a 11 (descendent).
- * El quart assertEquals() comprova que una parella de cel·les que tenen numeros i lletres s'ordenin adequadament.
- * El cinquè assertEquals() comprova que una parella de cel·les tenen funcioni correctament amb números decimals.
- * El cinquè assertEquals() comprova que una parella de cel·les tenen funcioni correctament amb números negatius.

Full

Per a la classe Full s'han efectuat 24 tests. Tots ells executats a la vegada amb el fitxer *TestFull*. Com bé es veu a la següent imatge cada un dels test imprimeix com quedaria visualment el full i les cel·les que conté després de realitzar el mètode:

```

andreuprensanz src % make TestFull
javac -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar -d ../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java ./main/domain/controllers/*.java ./main/domain/classes/types/*.java ./main/domain/controllers/*.java ./test/*.java ./test/types/*.java ./test/functions/*.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../lib/junit-4.12.jar:../lib/hamcrest-core-1.3.jar:../EXE/out/ org.junit.runner.JUnit4 test.TestFull
JUnit version 4.12
.Test Afegir Fitx Principi

|A1| | | | | | | | |
|B2| | | | | | | | |
|B3| | | | | | | | |
| | | | | | | | |
|C5| | | | | | | | |
| | | | | | | | |
|F6| | | | | | | | |
| | | | | | | | |

.Test Ordenar Columnes de forma descendent
Full abans d'ordenar la columna 'A'
|aaa| | | | | | | | | |
|123 A4|B2| | | | | | | |
|122 A5|B3| | | | | | | |
|zyx A2| | | | | | | | |
|abc A3| |C5| | | | | | | |
| | | | | | | | |
|F6| | | | | | | | |
|zzz| | | | | | | | |
| | | | | | | | |
Full despres d'ordenar la columna 'A'
|aaa| | | | | | | | | |
|zyx A2|B2| | | | | | | |
|abc A3|B3| | | | | | | |
|123 A4| |C5| | | | | | | |
|122 A5| | | | | | | | |
| | | | | | | | |
|F6| | | | | | | | |
|zzz| | | | | | | | |
| | | | | | | | |

```

```

Time: 0,151
OK (24 tests)

```

TestConstructora()

- * Objecte de la prova: Test de les constructores de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es creen 3 fulls per provar les 3 constructores diferents.
- * Operativa: Creem els fulls amb les constructores i comprovem que els valors obtinguts per els getters siguin els mateixos que els introduïts.

TestConstructoraExcepcio()

- * Objecte de la prova: Test de les constructores de Full amb errors
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es creen un full amb un número de columnes negatiu
- * Operativa: Quan es crea un full amb el numero de files o de columnes inferior a 0 (negatiu) llença una excepció de tipus NegativeArraySizeException

TestGetID()

- * Objecte de la prova: Test del mètode GetID() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un nou
- * Operativa: Creem una nova Cella amb el parametre "test" i comprovem que el valorUsuari sigui el mateix

TestGetCela()

- * Objecte de la prova: Test del mètode GetCela() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es consulten dues cel·les, una amb valor (0,0) i l'altre buida/no definida (9,9).

* Operativa: Comprovem que la cel·la definida a 0,0 ens retorni el valor assignat a setUp() i que la cel·la a la posició 9,9, que no esta definida, ens retorni null

TestGetCelaExcepcio()

* Objecte de la prova: Test del mètode GetCela() de Full amb errors

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es consulta una cel·la fora del full

* Operativa: Comprovem que si accedim a una cel·la fora del full, en aquest cas la posició 10,10 que correspondria a la 11a columna i la 11 fila, d'un full de 10x10 ens llençi l'excepció IndexOutOfBoundsException

TestSetNomFull()

* Objecte de la prova: Test del mètode SetNomFull() de Full

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es modifica el nom del full creat a setUp(), que té el nom per defecte de "Full 1" i que un cop canviat, el nou nom es el especificat i que el que retorna es l'anterior

* Operativa: Canviem el nom del full creat al SetUp() per "NouNom" i comprovem ue el retornat es "Full 1" amb un assertEquals()

TestModificaCela()

* Objecte de la prova: Test del mètode ModificaCela() de Full

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es modifiquen dues cel·les del full creat a setUp(), i es comprova que el boolea que retorna correspongui amb si la cel·la s'ha modificat o creat.

* Operativa: Modifiquem la cel·la a 0,0, que tenia anteriorment el valor de "A1" i fem un assertTrue() per comprovar que la cel·la s'ha modificat, despres modifiquem la cel·la a 0,1 i fem un assertFalse() per comprovar que la cel·la era buida. El metode printFull() ens permet veure el canvi efectuat pel test.

TestModificaCelaEliminar()

* Objecte de la prova: Test del mètode ModificaCela() de Full

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

* Valors estudiats: Estratègia caixa gris. Es modifiquen dues cel·les del full creat a setUp(), i es comprova que el boolea que retorna correspongui amb si la cel·la s'ha modificat o crea, a més, es comprova que al fer GetCela(), ens retorni null, ja que la posició es buidarà si el string es "".

* Operativa: Modifiquem la cel·la a 0,0, que tenia anteriorment el valor de "A1" i fem un assertTrue() per comprovar que la cel·la s'ha modificat seguit d'un assertNull de la mateixa cel·la per determinar que s'ha "eliminat"

* Despres modifiquem la cel·la a 0,1 i fem un assertFalse() per comprovar que la cel·la era buida seguit d'un assertNull() per comprovar que al fer get de dita cel·la segueix estant buida.

* El mètode printFull() ens permet veure els canvi efectuat pel test.

TestModificaCelaExcepcio()

* Objecte de la prova: Test del mètode ModificaCela() de Full amb errors

* Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.

- * Valors estudiats: Estratègia caixa gris. Es consulta modifica una cel·la amb índexs negatius.
- * Operativa: Comprovem que si modifiquem a una cel·la fora del full, en aquest cas la posició -1,-1 ens llença l'excepció `IndexOutOfBoundsException`.

TestAfegirFilaFinal()

- * Objecte de la prova: Test del mètode `AfegirFila()` de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'afegeix una fila al final del full i es comprova que el tamany s'hagi actualitzat i que hi puguem accedir mitjançant el `GetCela()` i `ModificaCela()`.
- * Operativa: Afegim una fila al final del full (últim index + 1 o directament el tamany) i comprovem que podem modificar una Cella a la posició 0,10 i que dita Cella es nova amb `assertFalse()`. Comprovem el tamany amb un `assertEquals()` de `GetNumFil()`.

TestAfegirFilaPrincipi()

- * Objecte de la prova: Test del mètode `AfegirFila()` de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'afegeix una fila al principi del full i es comprova que el tamany s'hagi actualitzat i que les cel·les presents pel `setUp()` s'hagin mogut adequadament..
- * Operativa: Afegim una fila al index 0, la primera fila i comprovem que la cel·la que abans estava a 0,0 segueix sent la mateixa amb `assertSame()`
- * El mètode `printFull()` ens permet veure els canvi efectuat pel test a les altres columnes.

TestAfegirFilaExcepcio()

- * Objecte de la prova: Test del mètode `AfegirFila()` de Full amb errors
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'afegeix una fila amb índex negatiu.
- * Operativa: Comprovem que si afegim una fila a un índex negatiu, llençi l'excepció `IndexOutOfBoundsException`

TestEliminarFilaFinal()

- * Objecte de la prova: Test del mètode `EliminarFila()` de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina una fila al final del full i es comprova que el tamany s'hagi actualitzat
- * Operativa: Eliminem la ultima fila del full (últim index + 1 o directament el tamany) i comprovem el tamany amb un `assertEquals()` de `GetNumFilal()`.

TestEliminarFilaPrincipi()

- * Objecte de la prova: Test del mètode `EliminarFila()` de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina una fila al index 0 del full i es comprova que el tamany s'hagi actualitzat i que una cel·la s'hagi desplaçat.
- * Operativa: Eliminem la primera fila del full i comprovem el tamany amb un `assertEquals()` de `GetNumFil()`.
- * Comprovem que la cel·la que es troba a 0,1 ara està a 0,0 amb un `assertSame()`

TestEliminarFilaExcepcio()

- * Objecte de la prova: Test del mètode EliminarFila() de Full amb errors
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina una fila fora del full.
- * Operativa: Comprovem que si eliminem una fila a un índex fora del full, llençi la excepció IndexOutOfBoundsException

TestAfegirColPrincipi()

- * Objecte de la prova: Test del mètode AfegirCol() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'afegeix una columna al principi del full i es comprova que el tamany s'hagi actualitzat i que les cel·les presents pel setUp() s'hagin mogut adequadament.
- * Operativa: Afegim una columna al index 0, la primera columna i comprovem que la cel·la que abans estava a 0,0 ara esta a 1,0 i segueix sent la mateixa amb assertSame()
- * El mètode printFull() ens permet veure els canvi efectuat pel test a les altres columnes.

TestAfegirColFinal()

- * Objecte de la prova: Test del mètode AfegirCol() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'afegeix una columna al final del full i es comprova que el tamany s'hagi actualitzat.
- * Operativa: Afegim una columna al final del full (el tamany) i comprovem que el tamany ha canviat amb un assertEquals() de GetNumCol().

TestAfegirColExcepcio()

- * Objecte de la prova: Test del mètode AfegirCol() de Full amb errors
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina una columna fora del full.
- * Operativa: Comprovem que si eliminem una columna a un índex fora del full, llençi la excepció IndexOutOfBoundsException

TestEliminarColPrincipi()

- * Objecte de la prova: Test del mètode EliminarColumna() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina la columna al index 0 del full i es comprova que el tamany s'hagi actualitzat i que una cel·la s'hagi desplaçat.
- * Operativa: Eliminem la primera columna del full i comprovem el tamany amb un assertEquals() de GetNumCol().
- * Comprovem que la cel·la que es troba a 1,1 ara està a 0,1 amb un assertSame().

TestEliminarColFinal()

- * Objecte de la prova: Test del mètode EliminarColumna() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina la ultima columna del full, al index 9 (nColumnes - 1) i es comprova que el tamany s'hagi actualitzat.
- * Operativa: Eliminem la ultima columna del full i comprovem el tamany amb un assertEquals() de GetNumCol().

TestEliminarColExcepcio()

- * Objecte de la prova: Test del mètode EliminarColumna() de Full amb errors
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'elimina una columna a un índex negatiu.
- * Operativa: Comprovem que si eliminem una columna a un índex negatiu, llençi l'excepció IndexOutOfBoundsException

TestOrdenaPerColumnesDescendent()

- * Objecte de la prova: Test del mètode OrdenarPerColumnes() de Full amb el boolea ascendent = false
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es modifiquen un conjunt de cel·les de la primera columna i després es crida sobre la primera columna, desde la segona fila (index 1) fins la 4a fila.
- * Operativa: Un cop cridada la operació d'ordenar, comprovem que el primer element es l'esperat
- * El mètode printFull() ens permet veure els canvi efectuat pel test a les altres files.

TestOrdenaPerColumnesAscendentBloc()

- * Objecte de la prova: Test del mètode OrdenarPerColumnes() de Full amb el boolea ascendent = true
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es modifiquen un conjunt de cel·les de la primera columna i després es crida sobre les primeres tres columnes, desde la segona fila (index 1) fins la 4a fila.
- * Operativa: Un cop cridada la operació d'ordenar, comprovem que el primer element es l'esperat
- * El mètode printFull() ens permet veure els canvi efectuat pel test a les altres files.

TestCercaValor()

- * Objecte de la prova: Test del mètode CercarValor() de Full
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. S'executa el primer cop per buscar una cel·la que apareix només un cop.
- * El segon cop es busca un substring que apareix més d'un cop al full
- * Per cada crida, recorre el Set que retorna comprovant que l'objecte Cella es el mateix que el que s'obté amb el GetCella() de full mitjançant un assertSame()
- * Operativa: Primer es crida CercarValor("A1"), i només s'espera un sol resultat, per la segona crida, busca l'string "B" i per a cada cel·la amb un caràcter 'B', es comprova que la cel·la retornada per CercarValor() i el GetCella() de les posicions retornades, són la mateixa Cella amb un assertSame()

Document

De la classe Document s'han realitzat 10 tests i tots ells passen correctament.

```
user@user-HP:~/Escriptorio/PROP/subgrup/FONTS/src$ make TestDocument
javac -d ../../EXE/out/ ./main/domain/classes/*.java ./main/domain/classes/functions/*.java ./main/domain/classes/exceptions/*.java
Note: ./main/domain/classes/Full.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -cp ../../lib/junit-4.12.jar:../../lib/hamcrest-core-1.3.jar -d ../../EXE/out/ ./main/domain/classes/*.java ./main/domain/
java ./main/domain/controllers/*.java ./test/*.java ./test/types/*.java ./test/functions/*.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
java -cp ../../lib/junit-4.12.jar:../../lib/hamcrest-core-1.3.jar:../../EXE/out/ org.junit.runner.JUnit4 test.TestDocument
JUnit version 4.12
...
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | |
| | | |
| | | |
| | | |

.....
Time: 0,021

OK (10 tests)
```

Test Constructores()

- * Objecte de la prova: Test del mètode Document(path) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Es crea un objecte Document.
- * Operativa: En aquest test es comprova que es crea correctament un objecte Document amb localització = path, títol = sense_titol, id = 0, doc = null
- * Primer es crea un objecte Document, es crida als mètodes GetLocal(), GetNom(), GetId() i GetDocument().size() i es comprova que el resultat és
- * el correcte.

testPrintDocument()

- * Objecte de la prova: Test del mètode printDocument() de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode per imprimir el contingut de Document, per veure visualment els canvis
- * Operativa: En aquest test es comprova que s'imprimeix un objecte Document correctament. Primer es crea un objecte Document, es creen dos Fulls amb els mètodes CrearFull(...) i s'imprimeix per pantalla el document.

testCreadoresFulls()

- * Objecte de la prova: Test dels mètodes CrearFull(), CrearFull(Nom), CrearFull(numFiles, numCols) i CrearFull(nom, numFiles, numCols) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode per crear fulls al document.

* Operativa: En aquest test es comprova que es creen correctament un els fulls en el document. Primer es crea un objecte Document, es crida als mètodes CrearFull(...) i es comprova que els resultats de GetID(), GetNomFull() i el GetNumFulls() siguin els correctes.

testCrearFullAmbNomExistent()

- * Objecte de la prova: Test del mètode CrearFull(Nom) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode per crear fulls al document.
- * Operativa: En aquest test es comprova que salta correctament l'Excepció.
- * Primer es crea un objecte Document, es crida al mètode CrearFull(nom) dues vegades i es comprova que salta l'Excepció.

testCanviarNomDoc()

- * Objecte de la prova: Test del mètode canviarNomDoc(nom) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. No calen fitxers addicionals.
- * Valors estudiats: Estratègia caixa gris. Mètode per canviar el nom del Document.
- * Operativa: En aquest test es comprova que es canvia el nom del Document correctament
- * Primer es crea un objecte Document, es crida al mètode canviarNomDoc(nom) i es comprova que GetNom() retorni el nom passat per paràmetre.

testExisteixFull()

- * Objecte de la prova: Test del mètode ExisteixFull(id) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode saber si Existeix el Full amb clau primària id.
- * Operativa: En aquest test es comprova si el mètode ExisteixFull(id) retorna true al passar-li una id existent.
- * Primer es crea un objecte Document, es crida al mètode crearFull() i es comprova que ExisteixFull(id) retorni true.

testNoExisteixFull()

- * Objecte de la prova: Test del mètode ExisteixFull(id) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode saber si Existeix el Full amb clau primària id.
- * Operativa: En aquest test es comprova si el mètode ExisteixFull(id) retorna false al passar-li una id no existent.
- * Primer es crea un objecte Document, es crida al mètode crearFull() i es comprova que ExisteixFull(id) retorni false.

testEliminarFull()

- * Objecte de la prova: Test del mètode EliminarFull(id) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode saber Eliminar el Full amb clau primària id.
- * Operativa: En aquest test es comprova si el mètode EliminarFull(id) elimina del Document el Full amb clau id.
- * Primer es crea un objecte Document, es crida al mètode crearFull() i al EliminarFull(id) i es comprova que ExisteixFull(id) retorni false.

testGetFull()

- * Objecte de la prova: Test del mètode GetFull(id) de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode per què ens retorni el Full amb clau primària id.
- * Operativa: En aquest test es comprova si el mètode GetFull(id) retorna el Full amb clau id.
- * Primer es crea un objecte Document, es crida al mètode crearFull() i es comprova que GetFull(id) retorni el full creat.

testGetTotsIDs()

- * Objecte de la prova: Test del mètode GetTotsIDs() de la classe Document
- * Fitxers de dades necessaris: Dades introduïdes manualment. Necessitem la classe Full.
- * Valors estudiats: Estratègia caixa gris. Mètode per què ens retorni una llista amb totes les id dels Fulls del Document.
- * Operativa: En aquest test es comprova si el mètode GetTotsIds() retorna totes les ids dels fulls del Document en una llista.
- * Primer es crea un objecte Document, es crida al mètode crearFull() dues vegades i es comprova que les id dels Fulls creats apareixin a la llista que retorna getTotsIDs().