

Projet Scala : 07/06/2022

Etape 01 : Dépôt git

Créer un nouveau projet sur github :

name : scalaProjectGS

description : yourName

allez dans le workspace depuis le terminal :

```
git init
```

```
echo "# scalaProjectGS" >> README.md
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git remote add origin https://github.com/\[yourId\]/scalaProjectGS.git
```

```
git push -u origin master
```

Etape 02 : Projet SBT

Dans IntelliJ :

File → New → Project → sbt

dans le build.sbt rajoutez la ligne suivante :

```
libraryDependencies += "org.scalatest" %% "scalatest" % "3.2.12" % "test"
```

Etape 03 : Développement

partie 01 : implémenter la class MyList fournis en PJ

(**hint** : Pattern matching et récursivité)

partie 02 :

package cars

abstract class Voiture : vitesse, nom, acceleration, moteur, prix, nombreDeCourse, marque

classe VoitureClassic (sous classe de Voiture)

classe VoitureSport (sous classe de Voiture)

classe VoitureSuperSport (sous classe de Voiture)

package pilote

classe Pilote : Map[course, classement] , nom, prénom, âge, voiture, experience

fonction:

dejaChampion

nombre de victoire

nombre de fois en podium

total des gain

information palmares

classe ListPilotes : MyList[Pilote]

fonction:

filtrez les pilotes selon les types de voiture, ajoutez des pilotes, supprimez des pilotes

package competition

classe Course : ListPilote, champion, date, circuit, typeVoiture, gain

fonction :

podium : 2 fonctions une random et une calculés selon les attributs : vitesse, accélération, experience et difficulté

classe Circuit : nom, lieu, longueur, difficulté

N.B : implémentez des fonctions d'affichage en String pour chaque classe et attribut.

N.B : ajoutez d'autres attributs et règles si besoin !

Partie 03 : exception

lors de la création des instances 4 règles doivent être respectée :

- 1) les pilotes ne doivent pas être mineur (age > 18 ans),
- 2) les types de voiture ont la classification de vitesses suivante :
VoitureClassic : 0 et 260 km/h
VoitureSport : 0 et 330 km/h
VoitureSuperSport : 0 et 400 km/h
- 3) les pilotes doivent avoir un classement compris entre 1 et 12 durant une course
- 4) une course ne peut avoir qu'un seul type de voiture

créez 3 classes d'exception :

InvalideAgeException

SpeedRangeException

RankRaceException

TypeCarException

gérez chaque exception de 3 manière différente : try..catch..finally, Option, Try

Partie 04 : Main

instancier les classes vu précédemment et simulez 2 courses.

- 1) vérification d'âge, de type de voiture et de vitesse
- 2) utilisation des fonctions créées précédemment !
- 3) génération d'un vainqueur avec un random et en comparant les attributs pour la deuxième course.

Etape 04 : Test unitaire

- avoir la même hiérarchie niveau package et classes
- avoir la même appellation des classes avec le suffixe Test
- pour chaque fonction prévoir 2 tests différents (dans un seul describe)

Etape 05 : Build

- modifier le fichier build.sbt pour permettre de lancer le main automatiquement et lors du packaging :
- lancer la compilation sur le sbt shell : compile
- lancer les tests sur le sbt shell : test
- lancer le run sur le sbt shell : run

```
// set the main class for the main 'sbt run' task
```

```
mainClass in (Compile, run) := Some("mainPackage.Main")
```

Etape 06 : Push

- créer un fichier .gitignore et rajoutez :

```
# sbt
```

```
project/project/
```

```
project/target/
```

```
target/
```

```
# intellij idea
```

```
*.iml
```

```
.idea
```

- allez dans le workspace depuis le terminal :
 - `git add *`
 - `git commit -m "final commit"`
 - `git push origin master`