

TP2 : Prise en main Scala, Scribble

Partage de code avec votre binôme Pour chaque TP, vous devrez vous assurer que chaque membre du binôme dispose bien du code produit. Pour partager votre TP en fin de séance, réalisez un export du projet correspondant (voir Notice ScalaIDE). **Attention** : vérifiez que l'export a bien été réalisé en essayant de l'importer dans ScalaIDE.

Auto-évaluation des TP Pour vous aider à auto-évaluer votre travail, nous vous fournissons quelques tests unitaires pour chaque TP. La Notice ScalaIDE décrit la procédure à suivre pour exécuter ces tests.

Exercice 1 (Import du TP2). *En vous aidant de la Notice ScalaIDE, importez le projet ScalaIDE TP2.zip disponible sur la page Moodle.*

1 Echauffement : package `fr.istic.si2.tp2.exercices`

Les exercices suivants sont à réaliser dans le fichier `echauffement.scala`.

Exercice 2 (Double, Quadruple). *On vous fournit une fonction `double`, qui calcule le double d'un entier passé en paramètre. En utilisant la fonction `double`, programmez une fonction `quadruple`. Il y a bien sûr plusieurs solutions, mais on impose d'utiliser la fonction `double` au moins 2 fois.*

Testez votre fonction sur quelques exemples en affichant le résultat qu'elle produit (à l'aide d'un simple `println` dans le code de votre application).

Exercice 3 (Renverser). *On vous fournit deux fonctions, `unite` et `dizaine` qui étant donné un entier, calculent respectivement son chiffre des unités, et son chiffre des dizaines. Programmez la fonction `renverser` qui calcule l'entier miroir d'un entier à deux chiffres. Son signe est préservé.*

Testez votre fonction sur quelques exemples en affichant le résultat qu'elle produit (à l'aide d'un simple `println` dans le code de votre application).

Exercice 4 (Tests unitaires). *Vérifiez que vos fonctions passent les tests fournis pour ces exercices. Aidez-vous de la notice ScalaIDE au besoin. Si nécessaire, corrigez vos fonctions et re-lancez les tests.*

2 La bibliothèque Scribble : package `fr.istic.si2.tp2.images`

Pour les exercices qui suivent, et pour le reste du module SI2, reportez-vous à la documentation en ligne de la bibliothèque Scribble (voir page Moodle). Demandez de l'aide à vos encadrants pour lire cette documentation.

Pour tester vos images, on rappelle que la fonction `draw(i:Image): Unit` permet d'afficher une image dans une fenêtre. Il suffit d'appeler cette fonction dans votre application Scala.

Les exercices suivants sont à réaliser dans le fichier `motifs.scala`, et traitent de la composition d'images simples.

Exercice 5 (Images simples). À l'aide des deux valeurs `sun` et `cloud` de type `Image`, définissez une troisième image `val cloudy: Image`, dans laquelle le nuage cache partiellement le soleil. Un exemple est donné ci-contre. Vous pouvez utiliser la fonction `onBackAt` de `Scribble`.



Exercice 6 (Formes géométriques simples). À l'aide de la bibliothèque `Scribble`, définir des valeurs de type `Image` qui représentent les symboles des contrôles multi-média courants suivants :



Les images seront des `val` de type `Image` nommées respectivement `previous`, `play`, `pause`, `stop`, `next`. Pour cela, vous pourrez utiliser les fonctionnalités suivantes de `Scribble` :

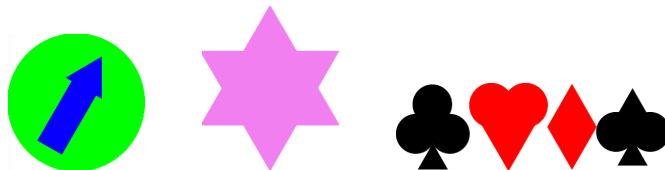
- les images de bases `Rectangle`, `Triangle`
- les fonctions de composition d'images suivantes : `on`, `beside`, `rotate`
- les fonctions de coloration d'images suivantes : `fillColor`, `lineColor`
- certaines couleurs sont définies comme des constantes dans `Scribble` : `BLACK`, `RED`, `GREEN`... Vous pouvez aussi définir vos propres couleurs (voir `Color`), en utilisant le schéma `RGB-α`.

On impose de **ne pas** utiliser la construction d'image `FromFile`, qui rendrait l'exercice inintéressant.

3 Exercices complémentaires

Ces exercices peuvent être réalisés en temps libre si vous n'avez pas le temps de les traiter en séance de TP2.

Exercice 7 (Autres motifs - optionnel). Dans le fichier `motifs.scala`, à l'aide de la bibliothèque `Scribble`, définissez des valeurs de type `Image` pour représenter des motifs ressemblant aux exemples ci-dessous : une flèche qui a subi une rotation, une étoile à 6 branches, des motifs utilisés pour un jeu de cartes...



Aucun code ne vous est fourni, ni suggéré, ni imposé ici. Amusez-vous, et découvrez la bibliothèque `Scribble` librement. Nous l'utiliserons dans les TPs et projets suivants.

Exercice 8 (Illustrer la météo). Dans le fichier `fichier illustration.scala`, en réutilisant les `val sun`, `cloud`, et `cloudy` de l'Exercice 5, programmez l'application `ImageMeteo`. On vous fournit une boucle d'interaction avec l'utilisateur, qui affiche l'image correspondant à la description du temps actuel sous forme de `String`. Vous devez programmer les fonctions `temps` et `tempsPossible`. Vous utiliserez pour ces fonctions un `pattern-matching`, plutôt qu'un test d'égalité sur les chaînes.

Exercice 9 (Mini projets bonus). Si vous avez terminé **correctement** les exercices ci-dessus, demandez les sujets *Bonus*. On vous y propose d'implémenter les applications *Sentinelle* et *Pixel Secret*, deux jeux simples avec interface graphique. Ils utilisent aussi la bibliothèque `Scribble`.