

1. Cahier des charges du client

Une entreprise fait appel à vos services pour réaliser une calculatrice en java.

Le développement va s'effectuer en deux lots :

- Lot 1 : interface en ligne de commande
- Lot 2 : interface graphique

a) Lot 1 → Interface console

Les règles de gestion à appliquer sont les suivantes :

Numéro	Description
CONSOLE_001	Au démarrage de l'application, afficher le message : « La calculatrice JCertif vous souhaite la bienvenue »
CONSOLE_002	La touche entrée entraine l'évaluation de l'expression saisie et l'affichage du résultat
CONSOLE_003	L'expression « exit » entraine l'arrêt du programme
CONSOLE_004	En cas de saisie d'une expression incohérente, afficher un message « L'expression saisie est incohérente »

b) Lot 2 → Interface graphique

Elle voudrait d'un produit qui aura la forme suivante :



Voici les règles de gestion à appliquer :

Numéro	Description
GRAPHIQUE_001	Un clic sur les touches de 0 à 9 entraîne l'affichage de la touche correspondante dans la zone d'affichage
GRAPHIQUE_002	Un clic sur la touche EFFACER entraîne l'effacement de la dernière saisie dans la zone d'affichage
GRAPHIQUE_003	Un clic sur la touche TOUT EFFACER entraîne l'effacement du contenu de la zone d'affichage
GRAPHIQUE_004	Un clic sur la touche = remplace la saisie effectuée par le résultat de l'opération
GRAPHIQUE_005	La saisie consécutive de deux opérateurs n'est pas autorisée
GRAPHIQUE_006	Le premier caractère ne doit pas être un opérateur
GRAPHIQUE_007	Chaque calcul effectué entraîne l'ajout de celui-ci dans la zone historique
GRAPHIQUE_008	Un clic le bouton SAUVEGARDER entraîne l'enregistrement de l'historique dans un fichier.

2. Développement & itérations proposées au client

c) Lot 1

Itération 1 : Implémenter la règle CONSOLE_001 (slide 1 à 9)

Pré-requis présentation :

- *Notion de package, visibilité entre les classes*
 - *Notion de classe*
 - *Main()*
 - *Afficher un message dans la console*
-
- Créer un projet Java sous Eclipse
 - Créer un package « principal »
 - Créer une classe « CalculatriceConsole » dans le package « principal »
 - Ajouter la méthode main()
 - Afficher la phrase « La calculatrice JCertif vous souhaite la bienvenue » dans la console

Pré-requis présentation :

- *Types primitifs*
 - *Déclarer une variable*
 - *Notion de constructeur*
 - *Notion de méthode & paramètres*
 - *Cast double -> entier*
-
- Créer un package « com.jcertif.university.calculatrice.service »
 - Créer une classe « CalculatriceEntierServiceImpl » dans le package « com.jcertif.university.calculatrice.service »
 - Implémenter les méthodes :
 - public int plus(int var1, int var2)
 - public int moins(int var1, int var2)
 - public int multiplier(int var1, int var2)
 - public int diviser(int var1, int var2)
 - Tester toutes ces méthodes dans la classe « CalculatriceConsoleMain », en prenant des exemples de paramètres d'entrées et afficher les résultats

Itération 3 : Lecture de deux entiers saisis au clavier et affichage des résultats de toutes les opérations

Pré-requis présentation :

- *La classe Scanner et System.in*
-
- Récupérer deux entrées successives du clavier et afficher les résultats de toutes les opérations à partir ces valeurs

Itération 4 : Construction d'un service plus complet sachant faire en plus des opérations sur des décimales

Pré-requis présentation :

- *Héritage*
 - *Polymorphisme*
 - *Override*
 - *Interface*
-
- Créer une classe « CalculatriceCompletServiceImpl » étendant la classe « CalculatriceEntierServiceImpl » dans le package « com.jcertif.university.calculatrice.service »
 - Implémenter les méthodes :
 - public double plus(double var1, double var2)

- public double moins(double var1, double var2)
 - public double multiplier(double var1, double var2)
 - public double diviser(double var1, double var2)
- Créer une interface « CalculatriceEntierService » dans le package « com.jcertif.university.calculatrice.service » ayant les méthodes :
 - public int plus(int var1, int var2)
 - public int moins(int var1, int var2)
 - public int multiplier(int var1, int var2)
 - public int diviser(int var1, int var2)
- Modifier les classes « CalculatriceEntierServiceImpl » et « CalculatriceCompletServiceImpl » pour qu'elles implémentent l'interface « CalculatriceEntierService »
- Utiliser l'interface dans la classe « CalculatriceConsoleMain »

Itération 5 : L'utilisateur peut désormais saisir directement une expression sur un entier et seul l'opération demandée est effectuée. Le programme tourne en continue, il ne s'arrête que qu'à la saisie de la chaîne « exit ».

Pré-requis présentation :

- La classe String
 - Structure conditionnel if/else
 - Boucle while/for
-
- Créer une boucle permettant de répéter les actions « saisie utilisateur » et « affichage de résultat ». La sortie de la boucle ne s'effectue que si l'utilisateur saisit « exit »
 - Récupérer la saisie de l'utilisateur et retrouver les 2 variables et l'opérateur pour pouvoir utiliser les services déjà développés. On pourra créer une nouvelle méthode int evaleurExpression(String expression) dans la classe CalculatriceCompletServiceImpl et ajouter cette méthode dans l'interface « CalculatriceEntierService »

Itération 6 : Implémenter la règle CONSOLE_004

Pré-requis présentation :

- Try/catch
-
- Récupérer toutes les erreurs inattendues et afficher un message d'erreur.

b) Lot 2



La calculatrice JCertif

Iteration 7 : Construire l'interface graphique de la calculatrice et implémenter toutes les règles de gestion

Pré-requis :

- Présentation Swing et des principaux composants
- Créer la classe CalculatriceSwingMain étendant JFrame.
- Implémenter les actions sur l'interface graphique