

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique



Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

LICENCE ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systemes Informatiques**

Présenté par :

- **Debbab Nesrine**
- **Medellel Wafa**
- **Chekima Asma**

Thème

Conception et réalisation d'une calculatrice (Multi-Devise)

Proposé et Encadré par : M. Guia Sana Sahar

Soutenue le xx-xx- 2017 Devant le jury:

M. Saci Medileh	MCA	Président
M. Samir Othmani	MAA	Rapporteur

Année Universitaire: 2016/2017

إهداء

أهدي هذا العمل:

إلى ينبوع الصبر والتفائل والأمل

والتي وهبتني الحياة والنشأة

على الشغف والاطلاع والمعرفة

إلى كل من في الوجود بعد الله ورسوله

" أمي الغالية "

إلى القلب الكبير " والدي "

إلى من حبهم يجرى في عروقي

ويلهج بذكراهم فؤادي

، إلى أخواني " بشير علي يوسف حفناوي "

وأخواتي " نسيم نورة حياة خديجة "

إلى أبناء أخوتي وأخواتي

إلى روح " جدي " الذي لم يحضى له أن يكون معي في هذه المرحلة من حياتي

— رحمة الله عليك —

وإلى عمي " أحمد، وجدتي، وعمتي، وخالتي

وجميع أحبتي وإلى أقربيهم إلى قلبي

إلى من تذوقت معهم أجمل اللحظات عمري أصدقائي

" رانية نسرین أسماء اشراق سليمة "

وفاء مدلل.

إهداء

أهدي هذا العمل:

إلى من أحمل اسمه بكل فخر إلى من أفتقده في كل لحظة ويرتعش قلبي لذكره إلى من أودعتني لله أبي
الغالي رحمة الله عليك.

إلى الاسم الذي يسري في الدم والروح التي عاشت بها روحي إلى من ربتي وأنارت دربي وأعانتني
بالصلوات والدعوات إلى أغلى إنسان في هذا الوجود أمي الحبيبة.

إلى من أرى التفاؤل بعينه والسعادة في ضحكته في نهاية مشواري أريد أن أشكرك على مواقفك
النبيلة إلى من تطلعت لنجاحي بنظرات الأمل أخي العزيز "أسامة" وإلى كل أخواتي الأعزاء "زهية"
"وسيلة" "يسمينة" وأولادهم وبناتهم.

إلى صديقاتي الغاليات "اسماء" "وفاء" "رانية" ارجوا لكن كل التوفيق في مشواركن الدراسي وكذا
المستقبلي.

والى أحبتي وعائلتي كل باسمه وبالأخص ابن خالتي "زياد"

نسرین دباب

إهداء

أهدي هذا العمل:

إلى من جرع الكأس فارغا ليسقيني قطرة حب ومن كلت أنامله ليقدم لنا لحظة سعادة إلى من حصد الأشواك عن دربي ليمهد لي طريق العلم إلى القلب الكبير أبي العزيز.

إلى من تتسابق الكلمات لتخرج معبرة عن مكنون ذاتها من علمتني وعانت الصعاب لأصل إلى ما أنا عليه الآن إلى من أرضعتني الحب والحنان أمي الغالية.

إلى إخوتي الغاليين "فوزي" "حسين" و "إسماعيل" وأخواتي الغاليات "فاطمة" "سعيدة" "مسعودة" "خديجة" وابن أخي العزيز "محمد ريان".

إلى الأخوات التي لم تلدهن أمي... إلى من تميزوا بالوفاء وكانوا معي في طريق النجاح إلى من عرفت كيف أجدهم وعلموني أن لا أضيعهم صديقتي: "نسرین", "أميرة", "وفاء", "رانية" وإلى كل أحبتي.

أسماء شكيمة

Remerciements

Louange à Dieu, qui nous a éclairé le chemin de la science, la connaissance et nous a aidés à accomplir ce devoir et nous aider à accomplir ce travail.

*Nous tenons à remercier tous ceux qui nous ont aidés de près ou de loin à la réalisation de ce travail, Notamment le professeur superviseur « **GUIA SANA SAHAR** » Ce qui ne nous retenir avec des conseils et des directives et qui nous aident à réaliser ce projet,*

*Nous ne pouvons manquer de remercier le Président du Département de l'informatique le professeur « **SASSI Meddelih** » et tout le personnel du Département.*

ملخص:

يتلخص مشروعنا هذا في إنشاء آلة حاسبة متعددة العملات ويكمن دورها في تحويل من عملة إلى أخرى وقادرة على إجراء العمليات الحسابية التقليدية على العملة المختارة لتسهيل تبادلات العمل في ما بينها مما يؤدي إلى تسهيل العلاقات التجارية الخارجية.

الكلمات المفتاحية : العملة، آلة حاسبة البرمجة الكائنية الموجهة.

Résumé:

L'objectif de notre travail est la réalisation d'une calculatrice multi-devises, en se basant sur la conception orientée objet à l'aide du langage de modélisation UML, et en utilisant un langage de programmation orienté objet.

La calculatrice multi devises s'occupe particulièrement de devises et de conversion d'une devise à une autre, de plus elle doit pouvoir faire les opérations arithmétiques classiques dans la devise choisie.

Les mots clé: Calculatrice, Devise, UML, Programmation Orientée Objet.

Abstract:

Our project is to create a multi-currency calculator and its role is to convert from one currency to another and the ability to perform traditional calculations on the selected currency. In order to facilitate trade exchanges between them leading to the facilitation of foreign trade relations.

Keywords: Currency, Calculator, Exchange.

LISTE DES FIGURES
INTRODUCTION GENERALE	1
CHAPITRE I: PRESENTATION GENERALE.....	3
1. INTRODUCTION	3
2. CALCULATRICE :.....	3
3. DEVISE:	3
4. LE TAUX DE CHANGE	4
5. DEFINITION DE PROBLEME:.....	4
6. CONCLUSION:	4
CHAPITRE II: MODELISATION	6
1. INTRODUCTION	6
2. LANGAGE DE MODELISATION UML:.....	6
3.1. <i>Définition de l'UML:</i>	6
3.2. <i>Les diagrammes de l'UML:</i>	6
3.2.1. Diagramme de cas d'utilisation:	7
3.2.1.1. Définition	7
3.2.1.2. Cas d'utilisation convertir:	7
3.2.1.3. Cas d'utilisation calculer:	8
3.2.2. <i>Diagramme d'activité:</i>	8
3.2.2.1. Définition:	8
3.2.3. <i>Diagramme de séquence:</i>	9
3.2.3.1. Définition	9
3.2.3.2. Diagramme de séquence d'une opération de calcul arithmétique:	10
3.2.3.3. Diagramme séquence d'une opération d'une conversion de devise: ...	11
3.2.4. <i>Diagramme de classe:</i>	12
3.2.4.1. Définition:	12
3.2.4.2. Classe Devise:	12
3.2.4.3. Classe calculatrice:.....	14
3. CONCLUSION:	14
CHAPITRE III : IMPLEMENTATION DE L'APPLICATION	16
1. INTRODUCTION	16
2. L'ENVIRONNEMENT DE PROGRAMMATION:	16
2.1. <i>Définition de NetBeans:</i>	16
2.2. <i>Les avantage:</i>	17
3. PRESENTATION D'APPLICATION:	17
3.1. L'interface:	18
3.2. <i>Le code :</i>	22
4. CONCLUSION	23
CONCLUSION GENERALE:.....	25
BIBLIOGRAPHIE:	26

Liste des figures

Figure 1. Diagramme de cas d'utilisation. -----	7
Figure 2. Cas d'utilisation Convertir. -----	7
Figure 3. Cas d'utilisation Calculer. -----	8
Figure 4. Diagramme d'activité. -----	9
Figure 5 .Diagramme de séquence d'une opération de calcul arithmétique. -----	10
Figure 6. Diagramme de séquence d'une opération de conversion de devise. -----	11
Figure 7. Diagramme de classe -----	13
Figure 8. L'environnement de programmation « NetBeans ».-----	16
Figure 9. Écran d'une calculatrice Multi-Devises -----	18
Figure 10 . La conversion de DZD à EUR. -----	19
Figure 11. La conversion à EUR.-----	19
Figure 12 . La conversion à TND.-----	20
Figure13 . Entrer le premier opérande et le type d'opération "Addition". -----	20
Figure 14. Entrer le deuxième opérande. -----	21
Figure 15. L'affichage du résultat. -----	21
Figure 16. Code classe Devise. -----	22
Figure 17. Code classe EUR -----	23

Introduction

Générale

Introduction

La monnaie nationale est l'unitaire monétaire d'un pays propre qui est utilisée dans les paiements internes entre les citoyens. Tandis que le devise qualifie l'unité monétaire d'un pays étranger, et qui représente une convention de chiffage d'une transaction commerciale entre des personnes morales ou physiques d'un même pays ou de pays différents. Il existe un cours des devises qui permet de fixer le taux de change entre deux devises, autrement dit qui permet de convertir la valeur d'une devise en une autre. Chaque devise est donc cotée sur un marché des devises et peut être librement échangée contre une autre. [1]

De là, se pose la nécessité pour l'échange de devises.

L'objectif de ce rapport est la réalisation d'une calculatrice (Multi -Devise) afin de faciliter le processus de l'échange de devise en passant par la conception à travers le langage de modélisation UML et en utilisant la (P.O.O) programmation orientée objet.

Ce rapport se compose de trois (3) chapitres :

Chapitre I: Dans ce chapitre nous avons donné quelques termes sont: calculatrice, devise, le taux de change, qui à liés à notre projet.

Chapitre II: Ce chapitre est destiné à la conception de notre application, en utilisant la programmation orientée objet et le langage de modélisation UML ce qui facilite la réalisation de calculatrice Multi-Devise.

Chapitre III: Ce chapitre présente l'implémentation de calculatrice Muti-Devise en se basant sur la modélisation du chapitre précédent. Nous avons commencé par la description de la plateforme utilise pour l'implémentation le «NetBeans», ensuite nous avons expliqué les différentes phases de réalisation.

Chapitre I

Présentation générale

Chapitre I: Présentation générale

1. Introduction

La méthode du commerce dans les temps anciens prend par voie l'échange de biens; mais maintenant compter sur la devise, et Chaque État possède une devise nationale spéciale, et on utilise cette devise nationale dans les paiements internes entre les personnes et on peut utiliser la devise entre les États lorsqu'il est nécessaire d'utiliser des devises.

Le changement de devises se faire entre paires deux devises différentes et en d'autres termes le prix de devise internationale ne pas déterminer sauf contre un prix d'autres devises international; donc comme ça ils se passent le processus d'échange de devises entre eux. Dans ce chapitre nous allons présenter d'un quelques concepts générale de base sur notre problème.

2. Calculatrice :

Une calculatrice est un objet électronique capable d'effectuer des calculs rapidement et sans erreur. Les premières machines à calculer étaient mécaniques mais aujourd'hui les calculatrices sont électroniques.

Il existe différents types de calculatrices qui vont des simples "4 opérations" pour effectuer des additions, soustractions, multiplications, divisions, aux calculatrices graphiques capables de représenter graphiquement des fonctions, en passant par les calculatrices scientifiques ou financières. [2]

3. Devise:

Une devise représente une unité monétaire qui est le plus souvent émise par la banque centrale d'un pays (euro, dollar, livre sterling...).

Donc, le terme devise est couramment employé pour désigner une monnaie étrangère qui est différente de celle de son pays.

La devise est émise par un gouvernement et circule dans une économie nationale. Utilisée comme instrument d'échange contre des biens ou des services.

Chaque pays ou chaque zone géographique possède sa devise officielle. Chaque devise se subdivise généralement en une autre unité qui équivaut à une fraction de la monnaie en vigueur. Un code de trois lettres est utilisé pour identifier chaque devise, USD, EUR,.... [3]

4. Le taux de change

Nombre d'unités de monnaie étrangère que l'on peut obtenir avec une unité de monnaie nationale. [4]

Exemple : 1 euro = 1,25 US dollars, noté EUR/USD=1,25.

5. Définition de problème:

L'absence d'une unité monétaire commune adoptée par les commerçants comme base pour le compte, ce qui provoque le problème des transactions économiques internationales de sorte que vous devez connaître la valeur de chaque monnaie par rapport aux autres monnaies.

6. Conclusion:

Dans ce chapitre nous avons donnée la présentation générale de quelque concept du problème. Dans le chapitre suivant nous allons voir comment utiliser le modèle calculatrice Multi-Devise pour le langage de modélisation UML.

Chapitre II

Modélisation

Chapitre II: Modélisation

1. Introduction

Dans ce chapitre nous allons modéliser le problème de calculatrice de multi - devise qui s'occupe particulièrement de devises et de conversion d'une devise à une autre. A fin d'arriver à une bonne modélisation et puisque le problème est résolu en utilisant la programmation oriente objet nous avons choisi le langage de modélisation UML.

2. Langage de modélisation UML:

3.1. Définition de l'UML:

UML (Unified Modeling Language) est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancées par l'OMG (Objet management groupe) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. Elle est héritée de plusieurs autres méthodes telles qu'OMT (Objet Modeling Technique) et OOSE (Objet Oriented Software engineering) et Booch. Les principaux auteurs de la notation UML son Grady Booch, Ivar Jacobson et Jim Rumbaugh. [5]

Dans UML, nous allons utiliser :

- Diagramme de cas d'utilisation.
- Diagramme d'activité.
- Diagramme de séquence.
- Diagramme de classe.

3.2. Les diagrammes de l'UML:

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. C'est une perspective du modèle, pas "le modèle". Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis). Un type de diagramme UML véhicule

une sémantique précise (un type de diagramme offre toujours la même vue d'un système). [6]

3.2.1. Diagramme de cas d'utilisation:

3.2.1.1. Définition

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. [7]

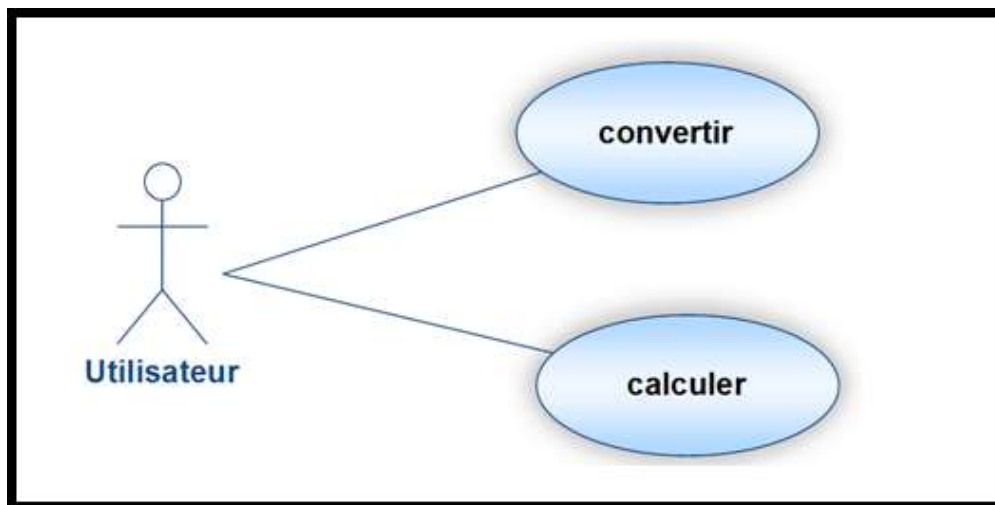


Figure 1. Diagramme de cas d'utilisation.

- Ce schéma décrit les principales tâches de notre programme, ce qui est de la conversion des devises et des calculs de performance sur la devise choisie.

3.2.1.2. Cas d'utilisation convertir:

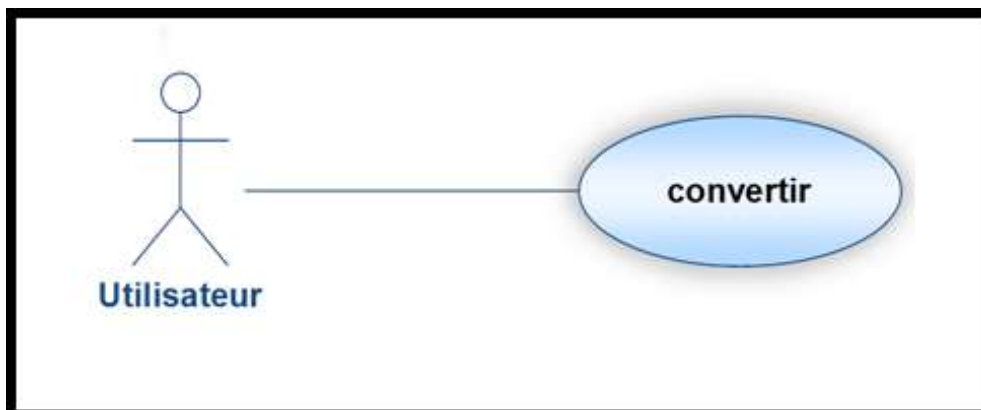


Figure 2. Cas d'utilisation Convertir.

Titre	Convertir
But	-convertir devise
Description de l'enchaînement	- Sélectionner la devise. -Entrer la somme d'argent. -Sélectionner la devise cible.
Acteur	L'utilisateur.

3.2.1.3. Cas d'utilisation calculer:

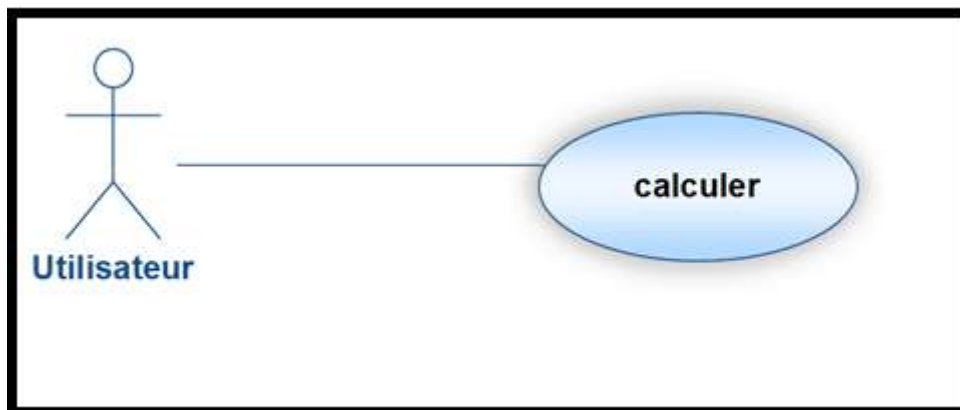


Figure 3. Cas d'utilisation Calculer.

Titre	Calculer
But	Effectuer des opérations arithmétiques.
Description de l'enchaînement	- Sélectionner la devise. -effectuer des calculs. - le choix de devise peut être modifié à la fin de l'opération arithmétique
Acteur	L'utilisateur

3.2.2. Diagramme d'activité:

3.2.2.1. Définition:

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter

Le diagramme de séquence montre les interactions entre les objets, agencés en séquence dans le temps ; il montre en particulier les objets participant à l'interaction par leurs lignes de vie et les messages qu'ils s'échangent ordonnés dans le temps mais il ne montre pas les relations entre les objets. [9]

Le diagramme de séquence est la représentation graphique des interactions entre l'utilisateur et le système selon un ordre chronologique dans la formulation UML.

-Ci-dessous une figure représentant le diagramme de séquence entre les différents composants du système :

3.2.3.2. Diagramme de séquence d'une opération de calcul arithmétique:

Le diagramme suivant présente le cas d'utilisation «calculer» :

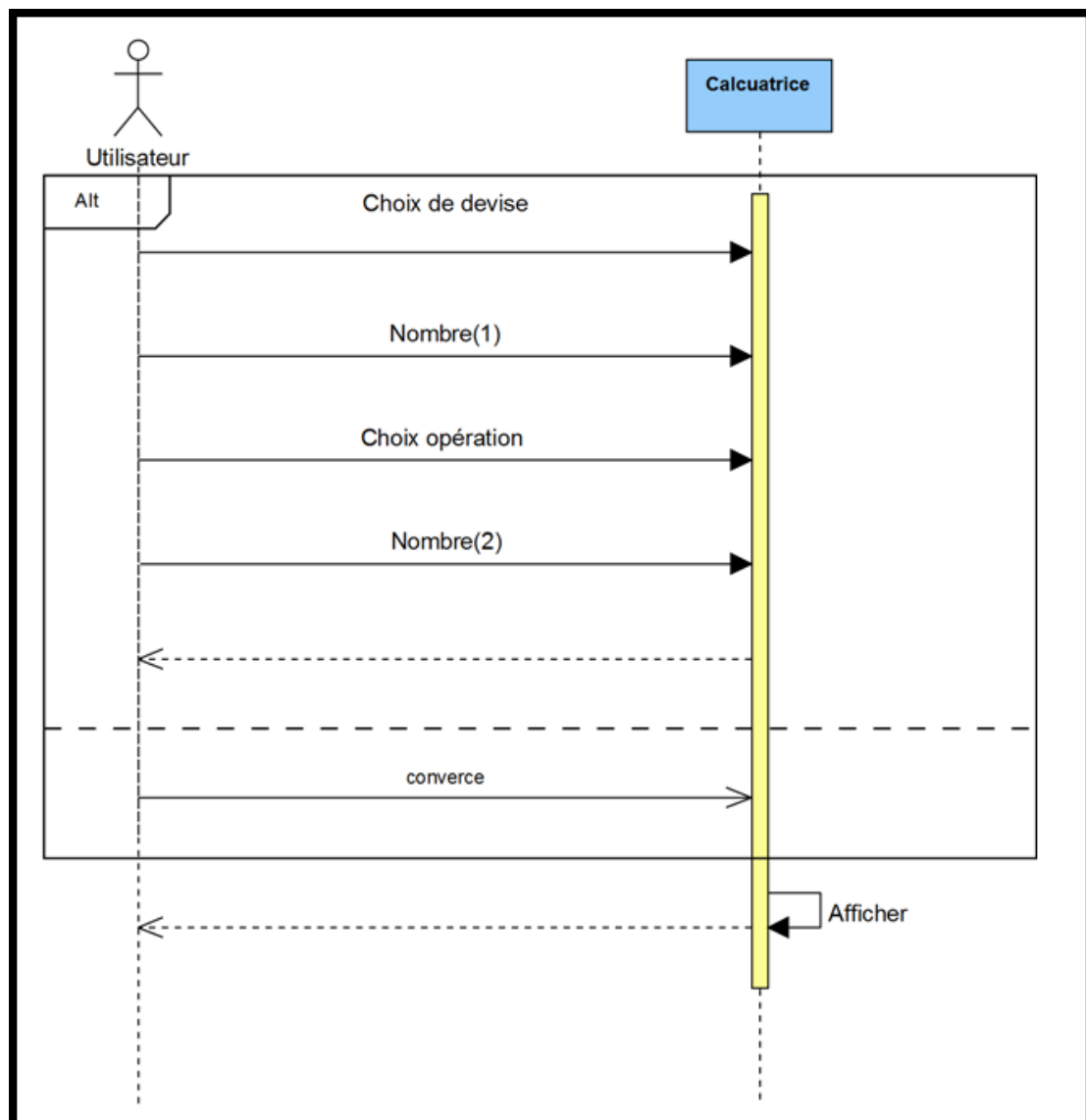


Figure 5 .Diagramme de séquence d'une opération de calcul arithmétique.

- ✓ **Intention:** l'opération de conversion de devise.
- ✓ **Actions:**
 - L'utilisateur il est choisi la devise (DZD, EUR ou TND).
 - Entrer le nombre 1.
 - Choix l'opération.
 - Entrer le nombre 2.
 - Converse le résultat.

3.2.3.3. Diagramme séquence d'une opération d'une conversion de devise:

Le diagramme suivant présente le cas d'utilisation «Convertir» :

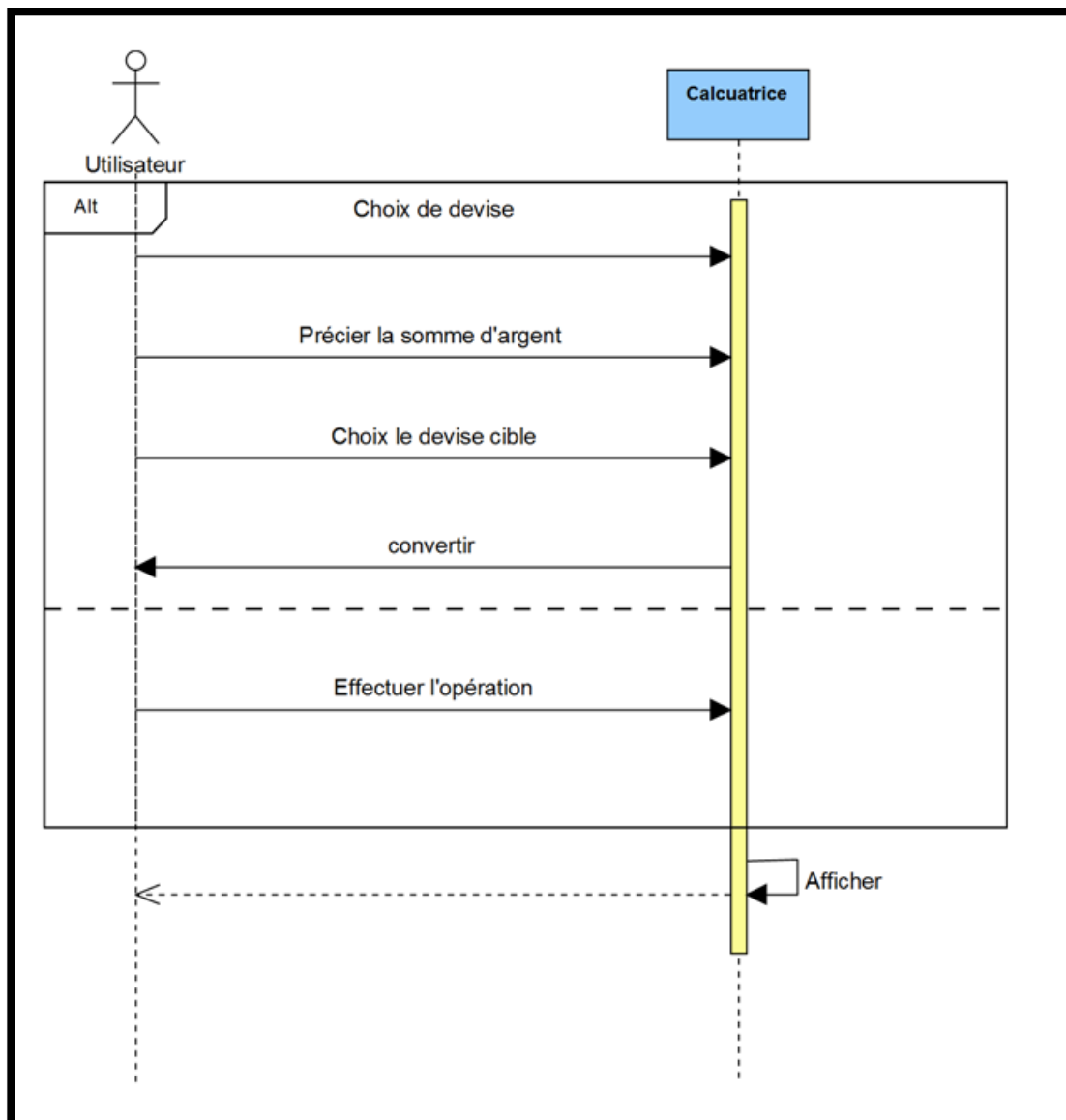


Figure 6. Diagramme de séquence d'une opération de conversion de devise.

- ✓ **Intention:** l'opération de conversion de devise.
- ✓ **Actions:**

- L'utilisateur il est choisi la devise (DZD, EUR ou TND).
- Entrer la somme d'argent.
- Sélectionner la devise cible.
- Effectuer les opérations.

3.2.4. Diagramme de classe:

3.2.4.1. Définition:

Le diagramme de classes permet de représenter des classes et leurs relations. Aussi, outre des classes, il peut contenir des types, des paquetages, des relations, voire des instances (objets et liens). Différents types de relations peuvent être exprimées dans un diagramme de classes, et notamment l'association (relation bidirectionnelle entre plusieurs classes). Le diagramme de classes exprime la traçabilité directe entre un processus, ses travailleurs et ses entités (i.e. entre le cas d'utilisation et ses classes). En ce sens, il s'agit d'un diagramme structurel statique. [9]

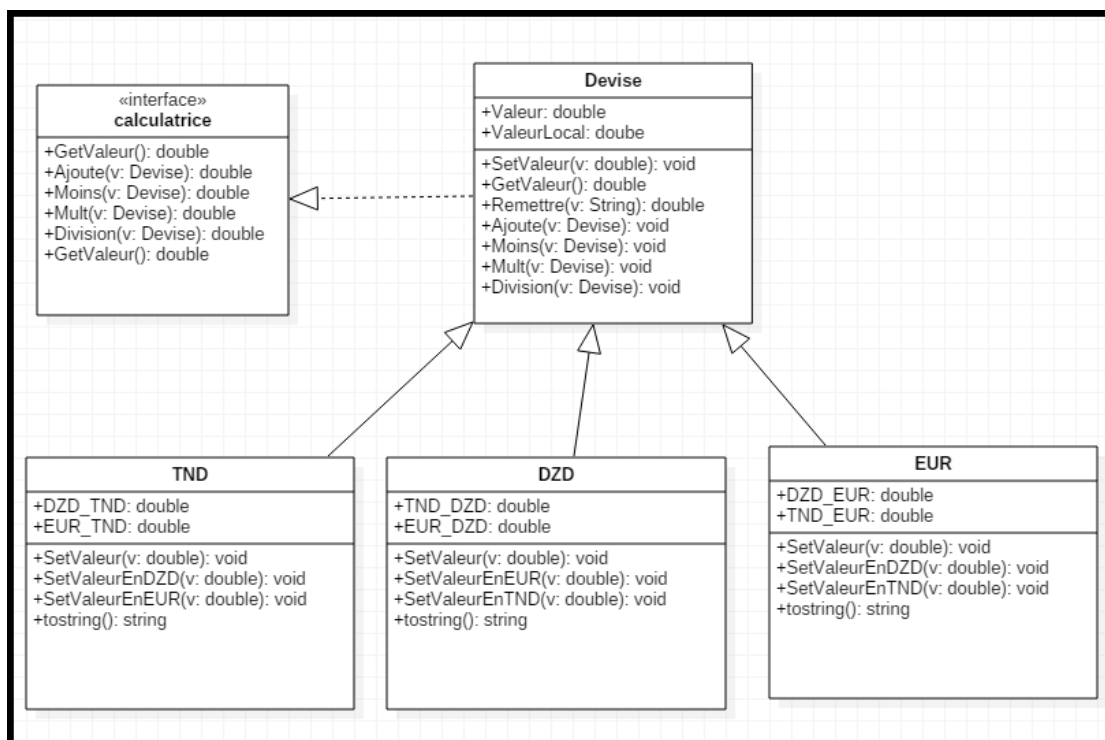


Figure 4. Diagramme de classe.

- La figure ci-dessus représente le diagramme de classe de la calculatrice multi - devises.

3.2.4.2. Classe Devise:

Le concept principal de notre problème est le concept de devise. Une devise représente une convention de chiffrage d'une transaction commerciale entre les

personnes d'un même pays ou de pays différents donc la classe Devise est introduit pour représenter ce concept.

Chaque type de devise considéré peut être introduit par une sous classe de la classe devise. Dans ce cas, aucun objet ne peut être créé à partir de la classe Devise. La classe Devise sera donc définie comme une classe abstraite. Une classe abstraite est une classe non-instanciable dont une partie des méthodes sont abstraites et restent à implémenter. [10]

Cette classe contient un attribut de type Double qui représente une valeur de devise transformée par l'opération convertir. Aussi il ya deux opération nécessaire dans cette classe sont: setValeur et getValeur.

Les sous classe de la classe Devise sont les classes : EUR, DZD et TND.

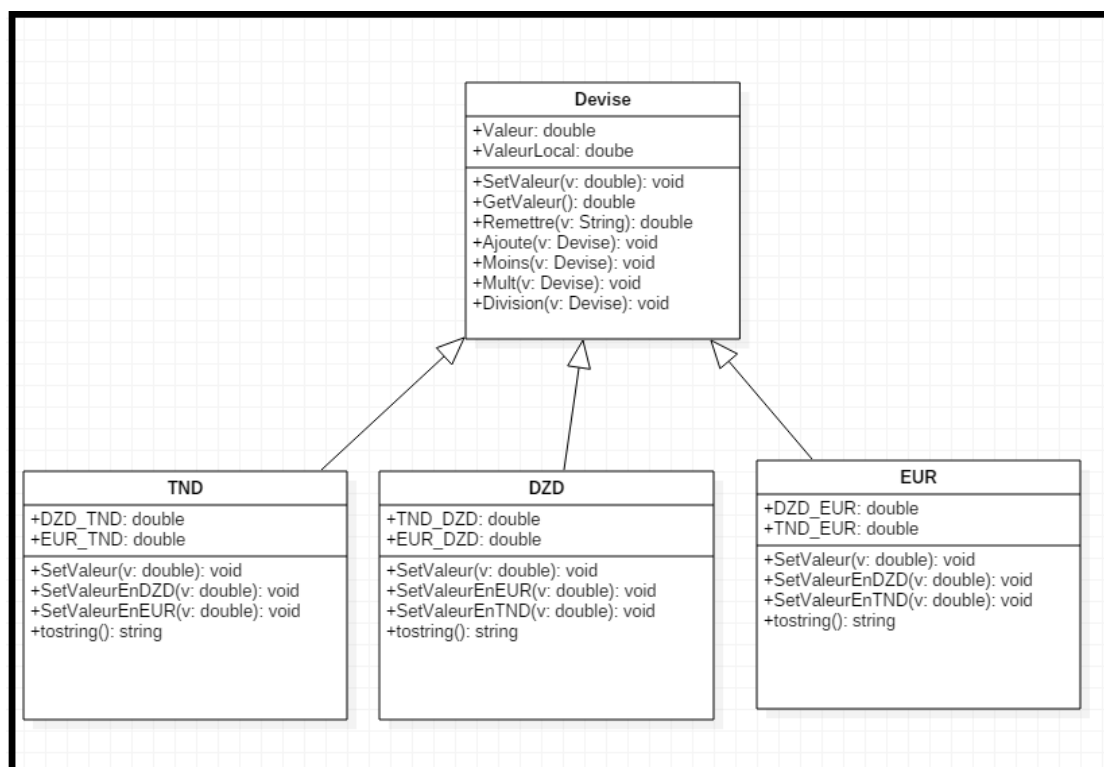


Figure 7. Diagramme de classe

Chaque classe (EUR, DZD, TND) contient deux variables, à savoir les coefficients de conversion.

EUR: la devise européenne unifiée.

DZD: la devise de l'Algérie

TND: la devise de Tunisienne

3.2.4.3. Classe calculatrice:

Pour utiliser correctement les sous classes de la classe Devise, il va falloir passer à travers un objet façade qui va recevoir uniquement toutes les opérations déclenchées par un écran de la calculatrice. Pour garder la généralité de notre solution nous passerons par la définition d'une interface qui définit le rôle joué par une calculatrice multi devises

La classe devise implémente cette interface

3. Conclusion:

Dans ce chapitre nous avons donné les détails de modélisation du calculatrice Multi-Devise, dans ce contexte nous avons utilisé le langage de modélisation UML, en commençant par le diagramme de cas d'utilisation, ensuite le diagramme d'activité, après cela le diagramme de séquence, en fin le diagramme de classe. L'utilité de la modélisation de notre système est de faciliter l'implémentation qui sera le sujet du chapitre suivant.

Chapitre III
Implémentation de
l'application

Chapitre III : Implémentation de l'application

1. Introduction

Dans ce chapitre, nous parlons de la mise en œuvre de la calculatrice multi devise. Tout d'abord, nous commençons par la description de l'environnement de programmation, Ensuite nous présentons notre application et le déroulement de son exécution.

2. L'environnement de programmation:

2.1. Définition de NetBeans:

NetBeans est un environnement de développement intégré (EDI), placé en Open Source par Sun. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, PHP, HTML ... Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). [7]

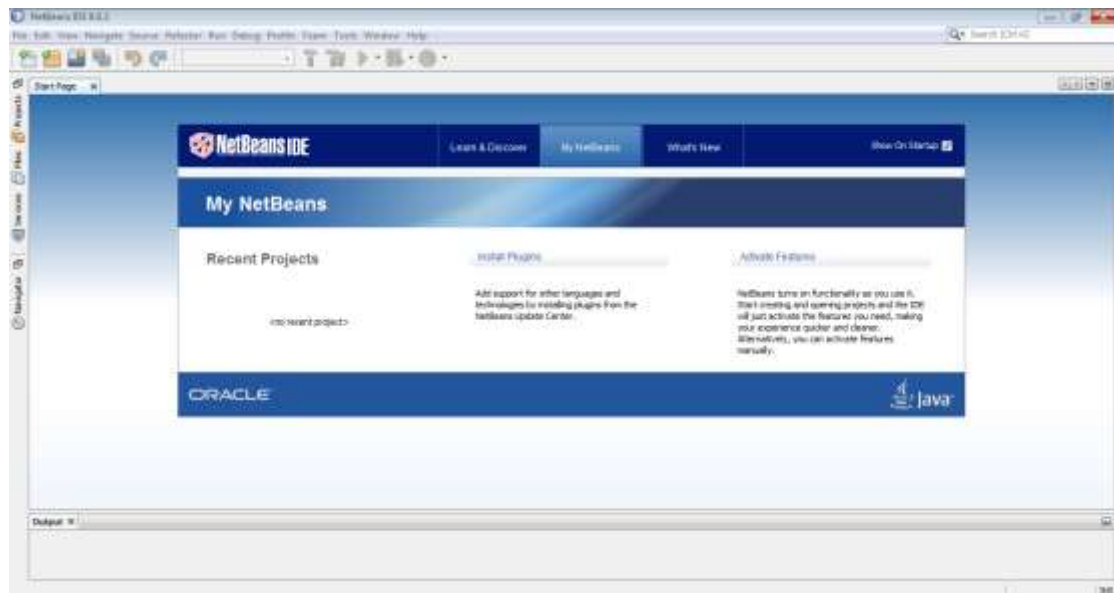


Figure 8. L'environnement de programmation « NetBeans ».

2.2. Les avantage:

Il y a plusieurs des environnements de programmation qui permet de développer, compiler et exécuter notre programme mais notre choix a porté sur NetBeans parce qu'il est:

- Un contexte de déploiement runtime pour des fonctionnalités arbitraires qui simplifient le développement
- Une boîte à outils qui permet de gagner beaucoup de temps en développement et d'effort
- Un ensemble d'abstractions qui permet aux développeurs de se concentrer sur la business logique, et non de réécrire de la logique de routine et des composants requis par la plupart des applications
- Un ensemble de Standards pour réhausser et renforcer la consistencet et l'interopérabilité entre les applications et les systèmes d'exploitation
- En tirant avantage de cette trousse à outils gratuite, basée sur des standards, les développeurs peuvent concevoir des applications complexes plus rapidement, avec une plus grande assurance de robustesse et de concevoir des applications qui résisteront à l'épreuve du temps. [11]

3. Présentation d'Application:

Cette application est développée en NetBeans permettant la conversion d'une devise à une autre, de plus elle fait les opérations arithmétiques classiques dans la devise choisie.

Généralement le changement de devises s'effectue sur des nombres et par des calculs de multiplication par des coefficients. L'ajout d'une nouvelle devise nécessitait des modifications sur des instructions, alors que dans une modélisation objet, on devrait se ramener au choix d'une classe pour construire de nouveaux objets.

La calculatrice doit pouvoir faire les quatre opérations arithmétiques classiques dans la devise choisie. A tout moment, le choix de la devise peut être modifié. Par exemple, si on a entré un nombre alors que le choix se trouve sur EUR, on peut successivement appuyer sur les autres boutons des devises pour voir s'afficher les sommes converties correspondantes.

Dans notre application nous allons traiter que trois devises (DZD, EUR, TND), mais il est possible d'ajouter de nouvelles devises en modifiant l'écran, avec un minimum de modifications dans la partie applicative.

3.1. L'interface:

L'interface de la calculatrice multi devises est composée de trois boutons de base qui représentent les devises: DZD, EUR, TND et chacun d'entre eux a deux fonctions:

- Déterminer la valeur entrée par type (de 0 à 9).
- Transférer le montant à un autre type en changeant le bouton sélectionné.

Les autres boutons (+, -, *, /) effectuent des calculs sur la devise choisie.

Comme nous savons que le taux de change est en train de changer à chaque instant que nous résultat dans le développement de cette calculatrice et connecté à Internet.



Figure 9. Écran d'une calculatrice Multi-Devises

Exemple 1: La conversion de Dinars Algérien à l'Euro puis au Dinars Tunisien

$$14 \text{ DZD} = 0.11840598 \text{ EUR} = 0.292622618673 \text{ TND.}$$



Figure 10 . La conversion de DZD à EUR.



Figure 11. La conversion à EUR.



Figure 12 . La conversion à TND.

Exemple 2: l'opération arithmétique d'Addition "+"

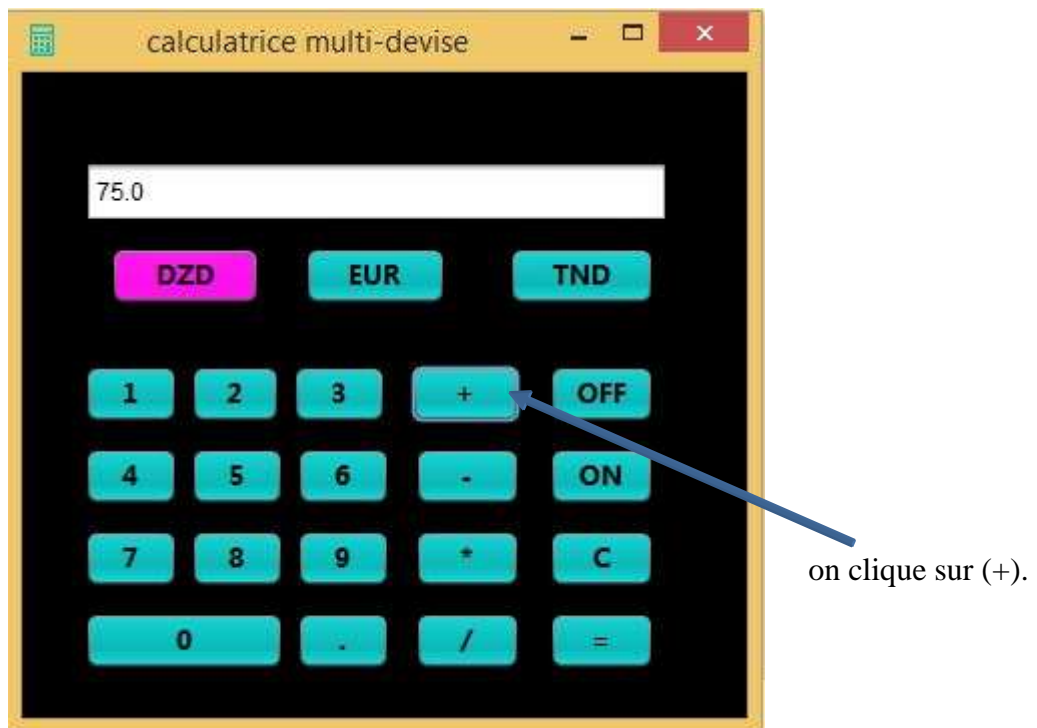


Figure13 . Entrer le premier opérande et le type d'opération "Addition".



Figure 14. Entrer le deuxième opérande.



Figure 15. L'affichage du résultat.

3.2. Le code :

```
public abstract class Devise implements calculatrice{

    public double Valeur = 0;
    public double ValeurLocal = 0;

    public abstract void SetValeur(double v);
    public abstract String toString();

    public double GetValeur() {
        return ValeurLocal;
    }

    public void Ajoute(Devise v)
    {
        SetValeur(ValeurLocal + v.GetValeur());
    }

    public void Moins(Devise v)
    {...10 lines }

    public void Division(Devise v)
    {...10 lines }

    public void Mult(Devise v)
    {...10 lines }

    public double Remettre(String v) {
        ValeurLocal =0;
        return ValeurLocal;
    }
}
```

Figure 16. Code classe Devise.

Cette classe représente le concept de base de notre application de plus elle implémente l'interface "Calculatrice".

Les méthodes (Ajoute, Moins, Division, Multi) calculent les opérations arithmétique sur un objet de type Devise.

De la classe Devise hérite les 3 classes : EUR, DZD et TND.

Exemple de la sous classe EUR héritée de la classe Devise

```
public class EUR extends Devise{
    public double EUR_DZD = 116.87;
    public double DZD_EUR = 0.00845757;
    public double TND_EUR = 0.404739;

    @Override
    public void SetValeur(double v)
    {
        super.ValeurLocal = v;
        super.Valeur = v * DZD_EUR;
    }
    public void SetValeurEndZD(double v)
    {
        super.Valeur = v;
        super.ValeurLocal = v * DZD_EUR;
    }
    public void SetValeurEntND(double v)
    {
        super.Valeur = v;
        super.ValeurLocal = v * TND_EUR;
    }
    @Override
    public String toString()
    {
        return Double.toString(GetValeur());
    }
}
```

Figure 17. Code classe EUR

4. Conclusion

L'implémentation est appliquée en utilisant l'environnement de développement **NetBeans** en suivant la modélisation UML vue au chapitre 2, dans le but de la réalisation souple productive et réutilisable de la calculatrice multi devises.

Conclusion Générale

Conclusion Générale:

Dans ce rapport nous avons implémenté la calculatrice multi devises permettant la conversion d'une devise à une autre, et de faire les opérations arithmétiques classiques dans la devise choisie, en se basant sur la conception orientée objet à l'aide du langage de modélisation UML, et en utilisant un langage de programmation orienté objet NetBeans.

L'utilisation de la programmation orientée objet fait pour une organisation spéciale de l'application en termes de connectivité et l'héritage qui se produit entre classes.

Beaucoup d'extensions sont encore possible, afin de faciliter et de simplifier les besoins de ressortissants par:

- Ajouter d'autres devises, en ajoutant des classes sans avoir à toucher au code déjà écrit vu le critère d'adaptabilité et de souplesse de la programmation orientée objet.
- La distribution via internet pour une utilisation en ligne à travers un navigateur compatible Java.

Bibliographie:

- [1]: "JND" / Définition du mot Devises/
<http://www.journaldunet.com/business/pratique/dictionnaire-economique-et-financier/14505/devises-definition.html/> Fernando Navarro/ juin 2017
- [2]: " dicodunet":
<http://www.dicodunet.com/definitions/sciences/calculatrice.htm/> Calculatrice
- [3]: <http://www.glossaire-international.com/pages/tous-les-termes/devise.html>
- [4]: Christian BIALÈS/ Coure Taux de change /02.12.13
- [5]: Bruno Bouzy/ UNIFIED MODELING LANGUAGE (UML)/ 14/06/01
- [6]: Laurent Piechocki / COURS UML / Document : version 13 / janvier 2003.
- [7]: E. K. H. Gorrab Firas" «Intermédiaire UML: Gestion de Clinique.2015/2014 .
- [8]: " developpez"[En ligne].Available:
<http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites#L6>
- [9]: Olivier Guibert/Le langage de modélisation objet UML/Édité le 6/4/10
- [10]: "Michel Lai" /Penser Objet avec UML et JAVA /Edition DUNOD
- [11]: "nblocalization":
https://nblocalization.netbeans.org/www/about/platform/index4_fr.html