RP - Module 320

NOM : VOTRE NOM ICI (suivi de 2 espaces;-)
Prénom : Votre prénom ici (suivi de 2 espaces;-)
Classe : Votre classe ici

Remarques (toute cette partie/remarque devra être supprimée avant reddition du RP) :

- **Génération automatique de table des matières**
  La table des matières ci-dessous est mise à jour automatiquement et automatiquement entretenue au fur et à mesure que vous ajouterez/supprimerez/modifierez la structure du document, et ce grâce à l'extension VSC Auto Markdown TOC
- **Tranformation du RP en PDF**
  Une fois votre RP en MD (MarkDown) terminé, on peut facilement le transformer en PDF afin de le stocker dans votre dossier de formation. Pour le moment, la "meilleure façon" pour produire un PDF avec son contenu bien présenté, le code en couleur, toutes les images utilisées, les schémas, ... reste d'utiliser le savoir-faire du navigateur. Faites simplement "Imprimer" puis "Imprimer en PDF" ou directement "Sauvegarder en PDF" sur Mac.
  Il existe également des extensions VSC diverses comme celle-ci (Markdown PDF) qui permettent de le faire directement ou à l'aide de nombreux outils externes spécialement conçus pour cela (sites web). Mais pour le moment (août 2024) aucune de ces autres solutions n'est satisfaisante.

Table des matières - 320

- Mots-clés du langage Java
  - Tous les mots-clés de Java
  - Mots-clés liés à l'héritage
    - `protected`
    - `super.`
    - `super()`
    - `abstract`
      - Représentation de `abstract` en UML
    - `extends`
      - Représentation de `extends` en UML
    - `instanceof`
  - Mots-clés liés à l'implémentation
    - `implements`
      - Représentation de `implements` en UML
    - `interface`
      - Représentation de `interface` en UML
- TDD - Test Driven Development / Extrême programming
  - Quels principes simples sont prônés par la TDD ?
  - Quels sont les avantages de faire de la TDD ?
  - À quoi sert la Javadoc ?
  - À quoi servent les TU (Tests Unitaires) ?
  - Quelle relation y a-t-il entre Javadoc et TU ?
  - Tests Unitaires
    - Qu'elle particularité une méthode doit-elle avoir pour qu'on puisse la tester ?
    - Que faut-il faire avec VSC pour réaliser les TU d'une telle méthode ?
    - Quels sont les pas de tests minimaux à prévoir pour un `int`
    - Quels sont les pas de tests minimaux à prévoir pour un `String`
    - Quels sont les pas de tests minimaux à prévoir pour un `double`
- UML
  - Les différentes relations en UML
    - Agrégation
    - Composition
    - Implémentation
    - Héritage

Introduction

Introduction sommaire sur ce qui sera vu durant ce module tiré du descriptif de module.

Structures de données

Récapitulatif des structures et utilité

Mettez ici les explications/image fournie qui récapitule tout ce qu'on a vu et qui sera un guide pour adéquatement choisir la bonne structure de données face à vos besoins.

Bean/Objet

Mini explication et exemple super simple/concret d'utilisation (bout de code Java propre et focalisé).

Enum

Mini explication et exemple super simple/concret d'utilisation (bout de code Java propre et focalisé).

Tableau

Une dimension

Mini explication et exemple super simple/concret d'utilisation (bout de code Java propre et focalisé).

Deux dimensions

Mini explication et exemple super simple/concret d'utilisation (bout de code Java propre et focalisé).

ArrayList<> et Vector<>

Comment créer un ArrayList<> vide

Le petit bout de code qui fait exactement cela

Comment créer un ArrayList<> avec le même contenu qu'une autre "liste"

Le petit bout de code qui fait exactement cela

Comment ajouter dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment insérer à un endroit dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment prendre un élément dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment rechercher un élément dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment modifier un élément dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment supprimer un élément dans un ArrayList<>

Le petit bout de code qui fait exactement cela

Comment traiter tous les éléments d'un ArrayList<> avec indice (fori)

Le petit bout de code qui fait exactement cela

Comment traiter tous les éléments d'un ArrayList<> sans indice (fore)

Le petit bout de code qui fait exactement cela

Comment vider complètement une liste ?

Le petit bout de code qui fait exactement cela

Comment trier le contenu d’une liste ?

Le petit bout de code qui fait exactement cela

Comment inverser le contenu d’une liste ?

Le petit bout de code qui fait exactement cela

Comment mélanger le contenu d'une liste ?

Le petit bout de code qui fait exactement cela

HashMap<> et HashTable<>

Comment savoir si une association/clé existe ?

Le petit bout de code qui fait exactement cela

Comment obtenir ce qui est associé à la clé ?

Le petit bout de code qui fait exactement cela

Comment créer une nouvelle association ?

Le petit bout de code qui fait exactement cela

Comment obtenir et parcourir l'ensemble des associations/clés ?

Le petit bout de code qui fait exactement cela

Comment vider complètement un HashMap ?

Le petit bout de code qui fait exactement cela

Comment transformer ce que retourne keySet() en un ArrayList ?

Le petit bout de code qui fait exactement cela

Autres méthodes utiles

Jetez un oeil et mentionnez-les et leur utilité

Boucles et collections

Boucle « fori » sur un ArrayList<>

Le petit bout de code qui fait exactement cela

Boucle « fori » sur les clés d'un HashMap<>

Le petit bout de code qui fait exactement cela

Boucle « fore » sur un ArrayList<>

Le petit bout de code qui fait exactement cela

Boucle « fore » sur les clés d'un HashMap<>

Le petit bout de code qui fait exactement cela

Formatage de nombres

Formater un `double` avec séparateur de milliers et 2 chiffres aprés la virgule ?

Exemple super simple et concret (code)

Formater un `int` avec séparateur de milliers et 3 chiffres significatifs ?

Exemple super simple et concret (code)

Formatage de dates et heures

Comment obtenir la date actuelle et heure actuelle

Exemple super simple et concret (code)

Formater la date actuelle au format JJ.MM.AAAA

Exemple super simple et concret (code)

Formater l'heure actuelle au format HH:MM:SS

Exemple super simple et concret (code)

Mots-clés du langage Java

Tous les mots-clés de Java

Faites un tableau avec tous les mots clés de Java et mettez en évidence (par exemple en gras) ceux qui ont été vus jusqu'ici

Mots-clés liés à l'héritage

`protected`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

`super.`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

`super()`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

`abstract`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

Représentation de `abstract` en UML

Comment reconnaît-on cela sur un diagramme UML de classes ?
Image/exemple

`extends`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

Représentation de `extends` en UML

Comment reconnaît-on cela sur un diagramme UML de classes ?
Image/exemple

`instanceof`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

Mots-clés liés à l'implémentation

`implements`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

Représentation de `implements` en UML

Comment reconnaît-on cela sur un diagramme UML de classes ?
Image/exemple

`interface`

Une explication très simple et directe de a) C'est quoi ? b) A quoi ça sert ? c) Comment on l'utilise ? avec un exemple super simple et concret (code).

Représentation de `interface` en UML

Comment reconnaît-on cela sur un diagramme UML de classes ?
Image/exemple

TDD - Test Driven Development / Extrême programming

Quels principes simples sont prônés par la TDD ?

Vos explications simples, concrètes, directes. Reprendre de la théorie vue ensemble.

Quels sont les avantages de faire de la TDD ?

Votre explication simple, concrète, directe. Reprendre de la théorie vue ensemble.

À quoi sert la Javadoc ?

Votre explication simple, concrète, directe.

À quoi servent les TU (Tests Unitaires) ?

Votre explication simple, concrète, directe.

Quelle relation y a-t-il entre Javadoc et TU ?

Votre explication simple, concrète, directe.

Tests Unitaires

Qu'elle particularité une méthode doit-elle avoir pour qu'on puisse la tester ?

Votre explication simple, concrète, directe.

Que faut-il faire avec VSC pour réaliser les TU d'une telle méthode ?

Votre tuto pour produire des TU d'une méthode à l'aide de VSC.

Quels sont les pas de tests minimaux à prévoir pour un `int`

Tableau des valeurs à impérativement tester, au minimum, lorsqu'un des paramètres est un entier.

Quels sont les pas de tests minimaux à prévoir pour un `String`

Tableau des valeurs à impérativement tester, au minimum, lorsqu'un des paramètres est une chaîne de caractères.

Quels sont les pas de tests minimaux à prévoir pour un `double`

Tableau des valeurs à impérativement tester, au minimum, lorsqu'un des paramètres est un nombre à virgule flottante.

UML

Les différentes relations en UML

Agrégation

Que signifie ce type de relation ? Comment le représente-t-on en UML ?

Composition

Que signifie ce type de relation ? Comment le représente-t-on en UML ?

Implémentation

Que signifie ce type de relation ? Comment le représente-t-on en UML ?

Héritage

Que signifie ce type de relation ? Comment le représente-t-on en UML ?

Utilisation

Que signifie ce type de relation ? Comment le représente-t-on en UML ?

Diagrammes UML

Diagramme des cas d’utilisation (savoir que ça existe, comprendre)

C’est quoi ?

A quoi ça sert ?

Un exemple concret simple (reprenez ce que le prof vous a donné)

Diagramme d’activité (savoir que ça existe, comprendre)

C’est quoi ?

A quoi ça sert ?

Un exemple concret simple (reprenez ce que le prof vous a donné)

Diagramme de classe (savoir lire, savoir produire, maîtriser)

C'est quoi ?

A quoi ça sert ?

Un exemple concret simple avec au moins :

- 2 classes ayant une relation entre-elles (avec cardinalités, nom de relation, type de relation), et une méthode publique, une privée et une statique
- 1 énumération

Diagramme de séquence (savoir lire, savoir produire, maîtriser)

C'est quoi ?

A quoi ça sert ?

Un exemple concret simple

MVC

C'est quoi ?

Vos explications claires, précises, simples.

Ça sert à quoi ?

Vos explications claires, précises, simples.

Le rôle des 3 parties d'un découpage MVC

Vos explications claires, précises, simples.

Exemple simple

Sous forme de diagramme UML

Réalisé en couleurs avec Entreprise Architect

Diagramme MVC complet avec les couleurs EMF correctes

Réalisé avec Mermaid

Diagramme MVC complet à l'aide de 'mermaid'

Sous forme de code Java

Les classes impliquées et leur code (sans ce qui est superflu, juste pour le MVC).

Conclusion et auto-évaluation

Ce que j'ai appris

Ce que j'ai aimé

Ce que je n'ai pas aimé

Mon auto-évaluation

Conclusions

Exemple de choses diverses en notation MarkDown (à supprimer à la fin)

Ceci est du texte, **du texte gras**, du *texte italique*, du texte, du texte, du `texte qui représente du code` mis en évidence, qui se termine ici.
Pour passer à la ligne suivante, il faut ajouter 2 espaces à la fin, ici il n'y en a pas. Du coup ce texte se poursuit normalement. Mais ici il y a 2 espaces à la fin.
Du coup ce texte est sur une nouvelle ligne !

Et celui-ci séparé du précédent car il y a 2x ENTREE.

Voici une liste de choses :

- la première
- la deuxième
    - la sous-deuxième 1
    - la sous-deuxième 2
- la troisième

Voici une liste numérotée :

la première

la deuxième

```
# la sous-deuxième 1
# la sous-deuxième 2
```

la troisième (<== notez que la numérotation générée est juste !)

Et voici comment présenter du code Java (attention à ne pas oublier de préciser le langage utilisé juste après les 3x` sinon il ne sera pas mis en forme en couleur)

```java
package javaapplication249;

import java.math.RoundingMode;
import java.text.DecimalFormat;
import java.text.DecimalFormatSymbols;
import java.util.Locale;

public class NumberFormatterDemo {

    public static void main( String[] args ) {

        double value = 12345678#123456;

        System.out.println( niceNumberFormatter( value )
        System.out.println( specialNumberFormatter( valu
    }

    public static String niceNumberFormatter( double val

        // Version simple et rapide
        DecimalFormat df = new DecimalFormat( "#,##0.000
        df.setDecimalFormatSymbols( DecimalFormatSymbols
        return df.format( value );
    }

    public static String specialNumberFormatter( double

        // Version 'on définit tous les détails'
        DecimalFormatSymbols dfSymbols = new DecimalForm
        dfSymbols.setDecimalSeparator( '.' );
        dfSymbols.setGroupingSeparator( '\'' );

        DecimalFormat df = new DecimalFormat();
        df.setDecimalFormatSymbols( dfSymbols );
        df.setMaximumFractionDigits( 4 );
        df.setGroupingUsed( true );
        df.setRoundingMode( RoundingMode.DOWN );

        return df.format( value );
    }

}
```

qui produira ce résultat :

```
123’456’78#1235
123'456'78#1234
```

Voici un simple tableau :

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

Et pour toute autre chose voici un lien vers de la documentation markdown : Documentation markdown

A final, vous pourrez supprimer ce chapitre et même produire un PDF de votre fichier (cherchez 'markdown to pdf' sous Google il y a des tas de solutions).