|  |
| --- |
| Document de suivi  de l’unité logicielle Automate  ~  Spécification de l’unité de programme  ~  Fiche de test pour validation de l’unité  ~  Bilan de validation  ~  Annexes  (sources, données de test, résultats…)  ~  Code Interne : SPEC-FDT-BV\_ULx |

Résumé

|  |  |  |  |
| --- | --- | --- | --- |
| * Présentation des spécifications fonctionnelles du composant * Présentation des spécifications d’interface du composant * Plan de validation du composant * Fiche de test type * Compte rendu de validation et Visa de validation |  | **Niveau de diffusion** | |
|  | Contrôlée |  |
|  | Interne | X |
|  | Libre |  |

Validation

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nom | Entité | Date |
| **Auteur(s)** | Jérémy Guimberteau | **L3 Miage –** Groupe 5 | 27/05/2009 |
| **Vérificateur(s)** | Anais Dubourg | **L3 Miage –** Groupe 5 | 29/05/2009 |
|  |  |  |
| **Validation(s)** | dddddddddddd | **L3 Miage - Enseignant** | … |

Historique de mise à jour

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Rédacteur | Date | Objet | Paragraphe(s)  Concerné(s) |
| **S0F0** | Jérémy Guimberteau | 27/05/2009 | Création du document | De §‎1 à §‎5 |
| **S0F1** | Anaïs Dubourg | 29/05/2009 | Bilan – Visa de validation | De §‎1 à §‎5 |
|  |  |  |  |  |

Documents associés

|  |  |
| --- | --- |
| Titre | Référence du fichier |
|  |  |

**Table des matières**

[1. Spécifications de l’unité de programme 4](#_Toc201064016)

[1.1 Contexte 4](#_Toc201064017)

[1.2 Spécifications fonctionnelles de l’unité logicielle 4](#_Toc201064018)

[1.2.1 Exigences fonctionnelles 4](#_Toc201064019)

[1.2.2 Performances et exigences qualité 4](#_Toc201064020)

[1.3 Spécifications d’interface de l’unité logicielle 5](#_Toc201064021)

[1.3.1 Description détaillée des interfaces 5](#_Toc201064022)

[1.3.2 Schéma de la solution retenue 5](#_Toc201064023)

[1.4 Contraintes de réalisation 5](#_Toc201064024)

[1.4.1 Conditions d’activation 5](#_Toc201064025)

[1.4.2 Conditions d’exception et d’arrêt 5](#_Toc201064026)

[2. Plan de validation de l’unité logicielle 6](#_Toc201064027)

[2.1 Stratégie 6](#_Toc201064028)

[2.2 Exemple de scénario valide 6](#_Toc201064029)

[3. Fiche de test pour la validation de l’unité 7](#_Toc201064030)

[4. Bilan de validation 8](#_Toc201064031)

[4.1 Validation par le programmeur 8](#_Toc201064032)

[4.2 Validation par des testeurs indépendants 8](#_Toc201064033)

[4.3 Visa de validation 8](#_Toc201064034)

[5. Annexes : 9](#_Toc201064035)

[5.1 Les sources du programme 9](#_Toc201064036)

[5.1.1 chemin de l’arborescence\UL1.java 9](#_Toc201064037)

[5.1.4 chemin de l’arborescence\UL0Stub.java 13](#_Toc201064040)

[5.1.2 chemin de l’arborescence\TestUL1.java 10](#_Toc201064038)

[5.2 Les fichiers d’entrée et de sortie 14](#_Toc201064041)

[5.2.1 [IN] - chemin de l’arborescence\\JDT.stub 14](#_Toc201064042)

[5.2.2 [OUT] - chemin de l’arborescence\JDT.run 14](#_Toc201064043)

# Spécifications de l’unité de programme

## Contexte

L’unité logicielle Automate s’inscrit dans le cycle de développement du service d’autotest du composant de vérification syntaxique.

## Spécifications fonctionnelles de l’unité logicielle

### Exigences fonctionnelles

L’unité logicielle Automate a pour objectif de vérifier la syntaxe d’une chaine hexadécimale.

Les spécifications de l’unité logicielle peuvent se résumer de la manière suivante :

1. Une chaine est passé en paramètre d’entré
2. Le composant vérifie cette chaine
3. Il retourne un objet de type format contenant les caractéristiques de la chaine passé en paramètre

Il doit être possible de prendre en entrée un **fichier de jeu de données** et de produire en sortie un **fichier résultat** mémorisant les traitements effectués.

### Performances et exigences qualité

RAS

## Spécifications d’interface de l’unité logicielle

### Description détaillée des interfaces

L’unité logicielle **Automate** a une unique méthode publique : hexa(*String chaine*) et une unique méthode privée : carToEvent(char c).

### Schéma de la solution retenue

Voici le schéma UML représentant la solution retenue pour l’unité logicielle Automate :

|  |
| --- |
| TraitFormat |
|  |
| public static Format hexa(String chaine)  private static TEvent carToEvent(char c) |

|  |
| --- |
| Format |
| private String hexa;  private int noFormat;  private int tabOctetB10[]; |
| public Format(String hexa, int NumFormat, int[] tabOctetB10)  public String getHexa()  public int getNoFormat()  public int[] getTabOctetB10()  public String toString() |

## Contraintes de réalisation

L’unité logicielle **Automate** a besoin d’un paramètre d’entrée de type String, ce paramètre d’entrée est la chaine hexadécimal à tester.

# 

# Plan de validation de l’unité logicielle

## Stratégie

Pour pouvoir affirmer que l’unité logicielle UL1 est 0-défaut, il est nécessaire que le résultat des tests effectués soit positif.

L’unité logicielle sera valide si le fichier de sortie (Oracle.run) est identique au fichier d’entrée (TraitFormat.jdt).

Voici un exemple de scénario permettant de valider l’unité logicielle.

## Exemple de scénario valide

**Soit le JDT suivant présenté en entrée :**

1 A0

2 17 B4

3 42EF

4 C2G3

**Le résultat attendu en sortie devra être :**

1 A0

2 17 B4

3 42EF

4 C2G3

Maintenant que nous avons élaboré la stratégie de validation, nous pouvons définir une fiche de test type.

# Fiche de test pour la validation de l’unité

|  |  |  |  |
| --- | --- | --- | --- |
| **FICHE DE TEST** | | | |
| **Date de création de la FDT :** | 31/05/2009 | **Rédacteur :** | Jérémy Guimberteau |
| **Contact :** | jeremy.guimberteau@gmail.com | **Entité :** | **L3 Miage Paris Descartes** |
| **Version de la FDT :** | 1 | **Code interne** | **FDT\_Automate** |
| **Intitulé :** | **Fiche de test unitaire de l’unité logicielle Automate** | | |

|  |  |  |
| --- | --- | --- |
| **DESCRIPTIF** | | |
| **Choisir la typologie du test** | **Test Unitaire** | Test d’intégration |
| **Description du test  - Données en entrée - Données en sortie** | | |
| L’unité logicielle UL1 permet ….  Ce programme va prendre en entrée un **fichier « jeu de données »** et va produire, en sortie, un **fichier résultat** mémorisant les traitements effectués.  1 A0  2 17 B4  3 42EF  4 C2G3  **Données en entrée**  1 A0  2 17 B4  3 42EF  4 C2G3  Les données en entrées : exemple de JDT que l’on pourra tester :  **Données en sortie**  Les données en sortie : exemple de fichier *run* que l’on pourrait obtenir : | | |

|  |  |  |
| --- | --- | --- |
| **PROCEDURES DE TESTS ET D’EVALUTAION** | | |
| **Mode d’évaluation des résultats :** | **Humain** | Automatique |
| **Procédures de tests :** | | |
| Pour tester la méthode **testTraitFormat(String fichierEntree, String fichierSortie),** …  Les données de tests sont présentes dans le répertoire :  Data tests\TraitFormat.jdt (ex : fichier TraitFormat.jdt)  Les résultats de sorties seront écrits dans le répertoire :  Data tests\Oracle.run (ex : fichier Oracle.run)  Pour l’évaluation des résultats et la validation de l’unité logicielle. | | |

# Bilan de validation

## Validation par le programmeur

|  |  |  |  |
| --- | --- | --- | --- |
| **Date de passage du test :** | 30/05/2009 | **Rédacteur :** | Jérémy Guimberteau |
| **Testeur :** | Jérémy Guimberteau | jeremy.guimberteau@gmail.com | |
| **Compte Rendu de validation** | **La validation de l’unité logicielle Automate est acceptée.** | | |
|  | | | |

## Validation par des testeurs indépendants

|  |  |  |  |
| --- | --- | --- | --- |
| **Date de passage du test :** | 31/05/2009 | **Rédacteur :** | A. MUHLEMANN |
| **Testeur(s) :**  (indépendant du programmeur) | Anaïs Dubourg | anais.dubourg@gmail.com | |
| **Compte Rendu de validation** | **L’unité logicielle est donc 0-Défaut.** | | |
|  | | | |

## Visa de validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Validation par le programmeur** | | | |
| **Visa Programmeur** | OK | **Date de validation** | 30/05/2009 |
| **Visa Testeur(s) indépendants** | OK | **Date de validation** | 31/05/2009 |
| **Conclusion** | **0-Défaut** | Bug Constaté | |
|  | | | |

# Annexes :

## Les sources du programme

### Code\automate\TraitFormat.java

**public** **class** TraitFormat {

**public** **static** Format hexa(String chaine){

String hexa = chaine;

hexa += '\n';

String H1 = "";

String H2 = "";

**char** c;

**int** i = 0;

**int** etat = 1;

**boolean** erreur = **false**;

Format unFormat = **null**;

**while**(i<hexa.length() && erreur == **false**){

**int** TabOctetB10[];

c = hexa.charAt(i);

TEvent event = *carToEvent*(c);

**switch**(etat){

**case** 1:

**switch**(event){

**case** *Hexa*:

etat = 2;

H1 += c;

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 2:

**switch**(event){

**case** *Hexa*:

etat = 3;

H1 += c;

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 3:

**switch**(event){

**case** *Hexa*:

etat = 5;

H2 += c;

i++;

**break**;

**case** *Esp*:

etat = 6;

i++;

**break**;

**case** *RC*:

TabOctetB10 = **new** **int**[1];

TabOctetB10[0] = Integer.*parseInt*(H1,16);

unFormat = **new** Format(H1,1,TabOctetB10);

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 5:

**switch**(event){

**case** *Hexa*:

etat = 7;

H2 += c;

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 6:

**switch**(event){

**case** *Hexa*:

etat = 9;

H2 += c;

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 7:

**switch**(event){

**case** *RC*:

TabOctetB10 = **new** **int**[2];

TabOctetB10[0] = Integer.*parseInt*(H1,16);

TabOctetB10[1] = Integer.*parseInt*(H2,16);

unFormat = **new** Format(H1+H2,2,TabOctetB10);

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 9:

**switch**(event){

**case** *Hexa*:

etat = 10;

H2 += c;

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

**case** 10:

**switch**(event){

**case** *RC*:

TabOctetB10 = **new** **int**[2];

TabOctetB10[0] = Integer.*parseInt*(H1,16);

TabOctetB10[1] = Integer.*parseInt*(H2,16);

unFormat = **new** Format(H1+" "+H2,3,TabOctetB10);

i++;

**break**;

**default** :

erreur = **true**;

}

**break**;

}

}

**if**(erreur){

**return** **new** Format(chaine,0,**null**);

}

**else**

**return** unFormat;

}

**private** **static** TEvent carToEvent(**char** c){

**if**('A' <= c && c <= 'F' || '0' <= c && c <= '9')

**return** TEvent.*Hexa*;

**else**{

**if**(c == ' ')

**return** TEvent.*Esp*;

**else**{

**if**(c == '\n'){

**return** TEvent.*RC*;

}

**else**

**return** TEvent.*Autre*;

}

}

}

}

### Les fichier d’entrée et de sortie

[IN] - Data Tests\TraitFormat.jdt

1 A0

2 17 B4

3 42EF

4 C2G3

[OUT] - Data Tests\Oracle.run

1 A0

2 17 B4

3 42EF

4 C2G3