



Cégep de Saint-Hyacinthe
Département d'informatique

Programmation orientée objet

420-2DP-HY

(3-3-3)

Initiation à la documentation automatique

(Version 1.2)

3 heures

Préparé par

Martin Lalancette

Comprendre les éléments suivants:

- Commentaires XML
- Génération automatique via DefaultDocumentation
- Documentation sous Markdown

Table des matières

| | |
|--|----|
| Introduction..... | 3 |
| Préparation de base | 3 |
| Pourquoi documenter? | 3 |
| Documentation avec Markdown et YAML | 4 |
| Qu'est-ce que le Markdown? | 4 |
| Éditer le fichier README.md du référentiel (Racine)..... | 7 |
| Option #1: En ligne | 7 |
| Option #2: Dans la solution Visual Studio | 7 |
| Documentation automatique..... | 9 |
| Améliorer les commentaires XML | 9 |
| Génération de documentation automatique | 10 |
| Activer la génération du XML | 11 |
| Installation de l'extension DefaultDocumentation | 12 |
| Configuration de cette extension..... | 12 |
| Créer un wiki à partir du code - FACULTATIF | 12 |
| Bibliographie..... | 14 |

Introduction

Cette séquence a pour but de vous initier aux notions de base nécessaires à l'introduction de la programmation. Nous commencerons par énoncer les éléments théoriques appuyés d'exemples simples et faciles à reproduire. Afin d'axer l'attention sur la compréhension de ces notions, il y aura des exercices à faire tout au long de cette séquence. **La séquence portera sur la génération de la documentation automatique.**

Préparation de base

Pour bien suivre les instructions qui vont être mentionnées tout au long des séquences d'apprentissage, une préparation de base s'impose. Il est important de créer un répertoire de travail (sur votre C: ou clé USB). Voici une suggestion d'arborescence:



Exercice 1. : S'assurer d'avoir créé l'arborescence ici haut mentionnée sur votre P ou votre clé USB.

Pourquoi documenter?

La documentation d'un produit (application, bibliothèque, site web, ...) est nécessaire afin d'informer nos utilisateurs:

- **Applications:**
 - Comment utiliser l'application, comment il fonctionne. Souvent associé à un manuel d'instructions.
 - Les paramètres d'exécution (ligne de commandes).
 - La nomenclature des fichiers en entrée et en sortie, s'il y a lieu.
 - Sur quel(s) systèmes d'exploitation peut-il s'exécuter.
 - Sur quel gestionnaire de bases de données doit-il se connecter avec la définition des tables et autres paramètres, s'il y a lieu.

- **Bibliothèques:**

- Comment déclarer les objets, leurs limitations.
- Indiquer quels membres peuvent être utilisés, comment les appeler.
- Les erreurs pouvant survenir (via les exceptions par exemple).
- Les hiérarchies et les interdépendances entre les objets

Documentation avec Markdown et YAML

Qu'est-ce que le Markdown?



Le Markdown (Wikipedia, 2021) est un langage de balisage léger qui a été créé par John Gruber avec Aaron Swart en 2004. Son but: Offrir une syntaxe facile à lire et à écrire servant à produire de la documentation. À partir de cette syntaxe, nous pouvons produire des documents en format HTML, PDF et bien d'autres. Cette notation est utilisée pour créer de la documentation dans DevOps (wiki) de votre projet. Depuis, plusieurs variantes sont apparues:

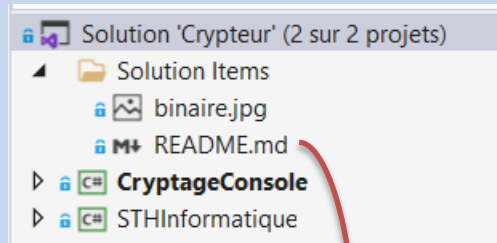
- [Daring FireBall](#) (site original)
- [MultiMarkdown](#)
- [GitHub Flavored Markdown Spec](#)
- [CommonMark](#) → Tentative de standardisation
- [Markdown Guide](#)
- [DevOps Wiki \(Markdown\)](#)

Extensions de fichiers: **.md** ou **.markdown**. Il peut y en avoir d'autres.

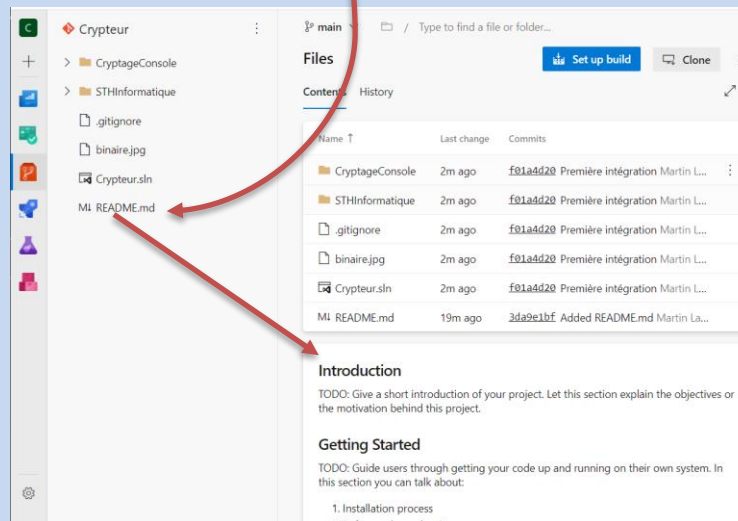
Exercice 2. : Dans DevOps:

- Créer une organisation appelée **2DPPrénomNomDocs**,
- Ajouter un **projet** nommé **Crypteur**,
- Initialiser un référentiel GIT avec README.md.
- Cloner le référentiel dans un dossier de travail,
- Extraire la solution jointe avec cet énoncé dans le dossier
- Faire les deux commits de base avec les fichiers à utiliser uniquement

- Ajouter les deux éléments existants à la solution comme suit:



- Ouvrir votre projet dans Azure DevOps également:



Un résumé de la [syntaxe](#) de base:

| Élément | Exemple code (Markdown) | Résultat |
|---|--|--|
| Entêtes (comme H1 à H6) | <pre># Niveau 1 ou ===== ## Niveau 2 ou ----- ### Niveau 3 #### Niveau 4 ... Placer une espace après le dernier #.</pre> | Niveau #1 Niveau #2 Niveau #3 Niveau #4 Niveau #5 |
| Paragraphe et sauts de ligne (2 ou plusieurs espaces) Comme <p> 2 espaces → | Indiquer quels membres peuvent être utilisés, comment les appeler. Indiquer quels membres peuvent être utilisés, comment les appeler. | Indiquer quels membres peuvent être utilisés, comment les appeler. Indiquer quels membres peuvent être utilisés, comment les appeler. |

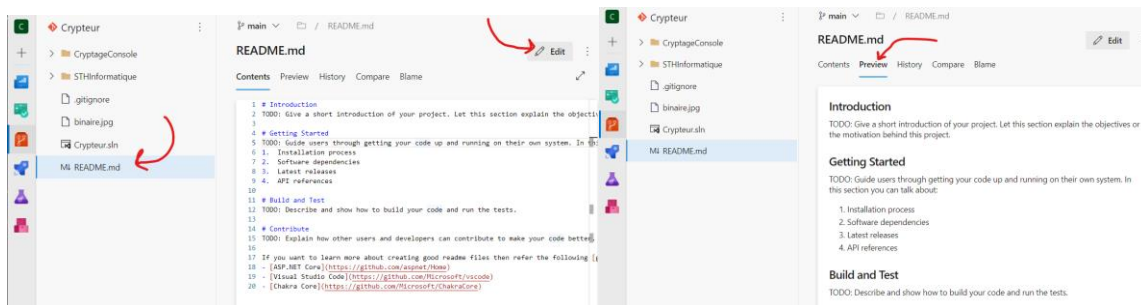
| Gras (2 X * ou _) | **Documentation** __Documentation__ | Documentation | | | | | | | | | | | | | | | | | | | | | |
|--|---|--|-----|-------|----------|--|--|--------|----------|-----|---------|--------|-----|--|--------|-----|-------|--------|----------|-----|---------|--------|-----|
| Italique (1 X * ou _) | <i>*Documentation*</i> _Documentation_ | <i>Documentation</i> | | | | | | | | | | | | | | | | | | | | | |
| Gras et Italique (3 X * ou _) | ***Documentation*** __Documentation__ | <i>Documentation</i> | | | | | | | | | | | | | | | | | | | | | |
| Liste ordonnée | <ol style="list-style-type: none"> 1. Élément1 2. Élément2 3. Élément3 4. Élément4 <ol style="list-style-type: none"> 1. Élément1 2. Élément2 <ol style="list-style-type: none"> 1. Sous-élément1 2. Sous-élément2 3. Élément3 4. Élément4 | <ol style="list-style-type: none"> 1. Élément1 2. Élément2 3. Élément3 4. Élément4 <ol style="list-style-type: none"> 1. Élément1 2. Élément2 <ol style="list-style-type: none"> 1. Sous-élément1 2. Sous-élément2 3. Élément3 4. Élément4 | | | | | | | | | | | | | | | | | | | | | |
| Liste à puces (-, * ou +) | <ul style="list-style-type: none"> - Élément1 - Élément2 | <ul style="list-style-type: none"> • Élément1 • Élément2 | | | | | | | | | | | | | | | | | | | | | |
| Liens (links) | [Cegep de Saint-Hyacinthe] (http://www.cegepsth.qc.ca/) | Cegep de Saint-Hyacinthe  | | | | | | | | | | | | | | | | | | | | | |
| Images | | | | | | | | | | | | | | | | | | | | | | | |
| Citations | <p>> Ceci est une citation.</p> <p>></p> <p>>> Elle contient un 2e niveau de citation.</p> | <p>Ceci est une citation.</p> <p>Elle contient un 2e niveau de citation.</p> | | | | | | | | | | | | | | | | | | | | | |
| Ligne horizontale (3 fois ou plus) | <p>***</p> <p>---</p> <p>_____</p> | <hr/> | | | | | | | | | | | | | | | | | | | | | |
| Tableau | <table> <tr> <td> Prénom </td> <td>Nom </td> <td>Poste </td> </tr> <tr> <td> -- -- -- </td> <td></td> <td></td> </tr> <tr> <td> Martin </td> <td>Tremblay </td> <td>450 </td> </tr> <tr> <td> Richard </td> <td>Durand </td> <td>321 </td> </tr> </table> <p>Pour aligner le contenu:</p> <p> -- = à gauche</p> <p> :--: = centrer</p> <p> --: = à droite</p> | Prénom | Nom | Poste | -- -- -- | | | Martin | Tremblay | 450 | Richard | Durand | 321 | <table> <tr> <th>Prénom</th> <th>Nom</th> <th>Poste</th> </tr> <tr> <td>Martin</td> <td>Tremblay</td> <td>450</td> </tr> <tr> <td>Richard</td> <td>Durand</td> <td>321</td> </tr> </table> | Prénom | Nom | Poste | Martin | Tremblay | 450 | Richard | Durand | 321 |
| Prénom | Nom | Poste | | | | | | | | | | | | | | | | | | | | | |
| -- -- -- | | | | | | | | | | | | | | | | | | | | | | | |
| Martin | Tremblay | 450 | | | | | | | | | | | | | | | | | | | | | |
| Richard | Durand | 321 | | | | | | | | | | | | | | | | | | | | | |
| Prénom | Nom | Poste | | | | | | | | | | | | | | | | | | | | | |
| Martin | Tremblay | 450 | | | | | | | | | | | | | | | | | | | | | |
| Richard | Durand | 321 | | | | | | | | | | | | | | | | | | | | | |

| | | |
|-------------------------|--|--|
| Checklist | - [] ou [x] (pour coché) | <input type="checkbox"/> Consultation en entreprise <input type="checkbox"/> Récupération de données cryptées <input type="checkbox"/> Sécurisation des données <input type="checkbox"/> Hébergement |
| Code | <code>`int byAge = 10` `Console.WriteLine(byAge.ToString());`</code> | <pre>int byAge = 10 Console.WriteLine(byAge.ToString());</pre> |
| Table de contenu | [[_TOC_]] | <div>Contents</div> <ul style="list-style-type: none"> • STH Informatique - Crypteur inc. (fictive) <ul style="list-style-type: none"> • Compagnie spécialisée dans la protection des données. • Histoire • Services offerts • Adresse • Contacts |

Éditer le fichier README.md du référentiel (Racine)

Option #1: En ligne

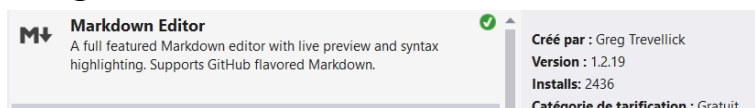
Dans Azure DevOps, sélectionner le fichier README.md et cliquer sur le bouton **Edit**:



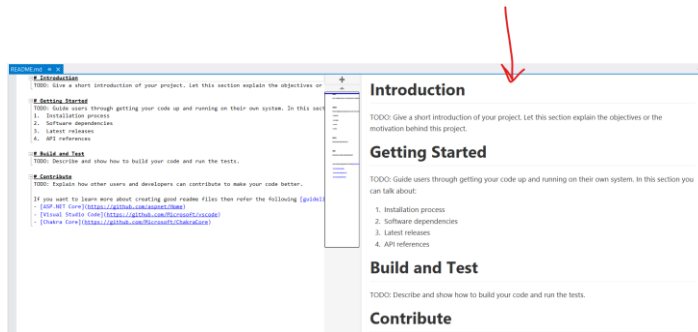
Option #2: Dans la solution Visual Studio

Afin de pouvoir éditer du code Markdown et prévisualiser dans Visual Studio, je vous suggère d'installer l'extension **Markdown Editor**. Voici la procédure:

1. Dans **Visual Studio**, aller dans le menu **Extensions** et choisir **Gérer les extensions**,
2. **En ligne**, chercher **Markdown** et choisir:



3. Une fois installée, lorsque vous ouvrirez un fichier .MD, vous verrez une prévisualisation (pas parfaite cependant). Exemple:



Exercice 3. : Modifier le fichier README.md pour produire ceci:

Qu'est-ce que STHInformatique - Crypteur

Unfollow 1 Edit

Martin Lalancette 2m ago

STH Informatique - Crypteur inc. (fictive)

Compagnie spécialisée dans la protection des données.

Contents

- STH Informatique - Crypteur inc. (fictive)
 - Compagnie spécialisée dans la protection des données.
 - Histoire
 - Services offerts
 - Adresse
 - Contacts

Histoire

Cette compagnie fut créée en 1998 par **M. Tremblay** alors qu'une demande croissante pour l'encryption et la déryption des données était forte.

Elle fut sollicitée par la compagnie IBM pour augmenter la protection de ses données.

Services offerts

- ☐ Consultation en entreprise
- ☐ Récupération de données cryptées
- ☐ Sécurisation des données
- ☐ Hébergement

Adresse

3000 ave. Boullé
Saint-Hyacinthe, QC
Canada
Cegep de Saint-Hyacinthe

Contacts

| Prénom | Nom | Poste |
|---------|----------|-------|
| Martin | Tremblay | 450 |
| Richard | Durand | 321 |

Suivre les instructions de l'enseignant.

Documentation automatique

Améliorer les commentaires XML

En C#, vous avez la possibilité d'alimenter les générateurs de documents afin de fournir le plus de détails possible via les commentaires XML que vous ajoutez dans votre code. Voici un lien à consulter:

[Documenter votre code C# avec des commentaires XML | Microsoft Docs](#)

Voici des [balises](#) de commentaires que nous avons déjà vues et de nouvelles à voir:

| Balise | Description |
|---|--|
| <summary> </summary> | <p>Cette balise doit être utilisée pour décrire un type ou un membre de type. Exemple:</p> <pre> /// <summary> /// Auteur: Martin Lalancette /// Description: Permet d'encrypter et décrypter du texte. /// Date: 2021-03-08 /// </summary> public class STHCryptage... public class STHCryptage Membre de ExercicesLib Résumé : Auteur: Martin Lalancette Description: Permet d'encrypter et décrypter du texte. Date: 2021-03-08 </pre>  |
| <param> </param> | <p>Cette balise doit être utilisée dans le commentaire d'une déclaration de méthode pour décrire l'un des paramètres de la méthode. Exemple:</p> <pre> /// <param name="sPhrase">Phrase à décrypter</param>... public string Décrypter(string sPhrase) Membre de ExercicesLib.STHCryptage Résumé : Auteur: Martin Lalancette Description: Décrypter une phrase en décalant de -1. Date: 2021-03-08 Paramètres : sPhrase: Phrase à décrypter </pre>  |
| <returns> </returns> | <p>Cette balise doit être utilisée dans le commentaire pour une déclaration de méthode afin de décrire la valeur de retour. Exemple:</p> <pre> /// <returns>Phrase normale</returns>... Retourne : Phrase normale </pre>  |

| | |
|--|--|
| <u><para></u> <u></para></u> | Cette balise est destinée à être utilisée à l'intérieur d'une balise, telle que <summary> , <remarks> ou <returns> , et vous permet d'ajouter une structure au texte. Permet de segmenter en paragraphe. |
| <u><code></u> <u>et</u> <u><c></u> | Utilisez <code> pour indiquer plusieurs lignes en tant que code. La <c> balise vous donne un moyen d'indiquer que le texte d'une description doit être marqué comme étant du code. |
| <u><exception></u> | La <exception> balise vous permet de spécifier les exceptions qui peuvent être levées. Cette balise peut être appliquée à des définitions de méthodes, de propriétés, d'événements et d'indexeurs. |
| <u><list></u> | Permet de définir une liste à afficher. |
| Etc. | |

Exercice 4. : Ajouter les commentaires manquants à la bibliothèque ainsi que la console.

Génération de documentation automatique

Il existe des compagnies qui offrent des extensions payantes facilitant la gestion de la documentation. Exemples:

- [GhostDoc](#)
- [Atomineer](#)
- [Doc-O-Matic](#)
- [VSdocman](#)
- Etc.

D'autres offrent la possibilité de les utiliser gratuitement:

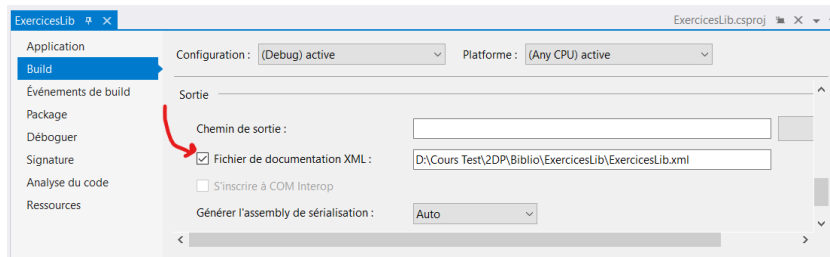
- [DocFx](#)
- [Doxygen](#)
- [Vsxml](#)
- [DefaultDocumentation](#)
- Etc.

Dans ce cours nous allons utiliser **DefaultDocumentation** qui va s'alimenter de fichiers XML générés à partir de nos commentaires.

Activer la génération du XML

Voici les étapes à suivre pour générer et régénérer le fichier XML à chaque compilation:

1. Cliquer sur le projet,
2. Dans le menu **Projet**, choisir **Propriétés de...**
3. Cliquer l'élément suivant:



4. Recompiler le projet et un fichier XML devrait apparaître. Le contenu devrait ressembler à ceci avec tous les commentaires:

```
<?xml version="1.0"?>
<doc>
  <assembly>
    <name>STHInformatique</name>
  </assembly>
  <members>
    <member name="T:STHInformatique.STHCryptage">
      <summary>
        Auteur: Martin Lalancette
        Description: Classe contenant les opérations d'encryption et de décryption.
        Date: 2021-04-25
      </summary>
    </member>
    <member name="M:STHInformatique.STHCryptage.Encrypter(System.String)">
      <summary>
        Auteur: Martin Lalancette
        Description: Permet d'encrypter une phrase
        Date: 2021-04-25
      </summary>
      <param name="sPhrase">Phrase à encrypter</param>
      <returns>Phrase cryptée</returns>
    </member>
    <member name="M:STHInformatique.STHCryptage.Décrypter(System.String)">
      <summary>
        Auteur: Martin Lalancette
        Description: Permet de décrypter une phrase.
        Date: 2021-04-25
      </summary>
      <param name="sPhrase">Phrase cryptée</param>
      <returns>Phrase décryptée</returns>
    </member>
  </members>
</doc>
```

Installation de l'extension DefaultDocumentation

Pour convertir les informations contenues dans le XML, il faut installer dans le projet l'extension [DefaultDocumentation](#) via NuGet. Voici les étapes:

1. Cliquer-droit sur le nom du projet et choisir **Gérer les packages Nu-Get...**
2. Dans **Parcourir**, rechercher **DefaultDocumentation**. Exemple:



3. Prendre la dernière version et cliquer sur **Installer**.

Configuration de cette extension

Dans votre fichier .csproj, vous pouvez ajouter les options suivantes selon les besoins:

```
<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|AnyCPU'">
  <DocumentationFile>STHInformatique.xml</DocumentationFile>
  <DisableDefaultDocumentation>false</DisableDefaultDocumentation>
  <DefaultDocumentationFolder></DefaultDocumentationFolder>
  <DefaultDocumentationInvalidCharReplacement>.</DefaultDocumentationInvalidCharReplacement>
  <DefaultDocumentationFileNameMode>Name</DefaultDocumentationFileNameMode>
  <DefaultDocumentationGeneratedPages>Types, Members</DefaultDocumentationGeneratedPages>
  <DefaultDocumentationAssemblyPageName>readme</DefaultDocumentationAssemblyPageName>
</PropertyGroup>
```

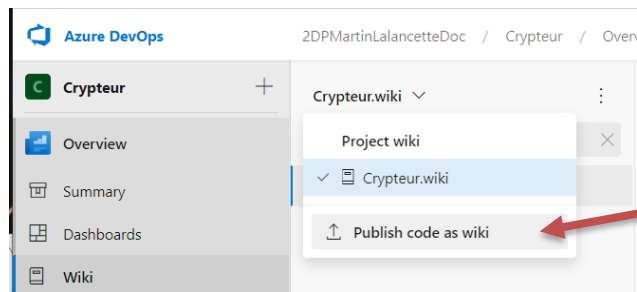
Exercice 5. : Ajouter les options à votre projet (.csproj) après explications de l'enseignant. Une fois terminée:

- Régénérer le projet avec les fichiers (.md)
- Valider et pousser tous les fichiers modifiés vers Azure

Créer un wiki à partir du code - FACULTATIF

Afin de rendre disponible la documentation disponible dans DevOPS (wiki – code), voici la [procédure](#) à suivre:

1. Accéder à votre **projet** dans DevOps,
2. Dans **Overview**, cliquer sur **Wiki**,
3. Cliquer sur la flèche à côté de Crypteur.wiki



4. Cliquer sur Publish code as wiki et remplir comme suit:

Publish code as wiki ✕

Markdown pages from the selected folder in the code repository will be published as Wiki.

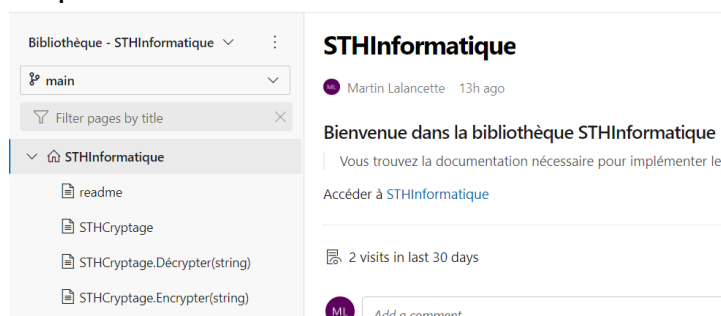
Repository

Branch

Folder

Wiki name

5. Cliquer sur le bouton **Publish**



Vérifier les informations de chaque page.

6. Ajouter un lien dans le wiki du projet pour donner l'accès à cette bibliothèque.

Exercice 6. : Exécuter cette procédure. Vérifier l'affichage. Faire des changements.

Bibliographie

Wikipedia. (2021, 04 22). *Markdown*. Récupéré sur Wikipédia - L'encyclopédie libre:
<https://fr.wikipedia.org/wiki/Markdown>