

Documentation de l'architecture et du code

Table des matières

I. Documentation auto-générée	3
II. Exercice : Quiz	5
III. Documentation Technique de l'Architecture - DAT	6
IV. Exercice : Quiz	9
V. Essentiel	11
VI. Auto-évaluation	11
A. Exercice	11
B. Test	11
Solutions des exercices	12

I. Documentation auto-générée

Contexte

Ce cours est destiné à la documentation du code. Nous verrons comment mettre en place les différentes méthodes de documentation auto-générée, la mise en place d'une documentation technique, puis nous parcourrons les différentes méthodes à mettre en place pour pallier les incidents pouvant intervenir durant le projet. Nous allons apprendre comment intervenir rapidement et efficacement.

Introduction

Lorsque vous développez, que ce soit sur iOS, Java, HTML ou quel que soit votre langage de programmation, il est possible de mettre en place des commentaires. Ces commentaires servent principalement à faire comprendre le principe d'une ligne de code pour que quelqu'un d'autre puisse la reprendre ou alors simplement assimiler le processus de développement.

Ces commentaires sont exprimés sous cette forme :

```
1  //
2  //  ContentView.swift
3  //  DocumentationAutoGeneree
4  //
5  //  Created by Ruben Figlarz on
   //    26/04/2021.
6  //
7
8  import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         Text("Hello, world!")
13             .padding()
14     }
15 }
16
17 struct ContentView_Previews:
   PreviewProvider {
18     static var previews: some View {
19         ContentView()
20     }
21 }
22
```

Les commentaires sont les lignes grises commençant par « // »

Pourquoi parlons-nous de documentation auto-générée si le développeur intègre ces commentaires manuellement ?

Compilation des commentaires

Lorsque l'on parle de documentation générée, cela ne veut pas dire qu'une documentation se matérialise naturellement. Elle est seulement le résultat d'une compilation des commentaires et d'un assemblage dans un document Word, PDF, XML ou HTML.

Cette compilation se produit grâce à un **générateur de documentation**.

Qu'est-ce qu'un générateur de documentation ?

Le générateur de documentation est un outil de programmation, utilisant tous les commentaires de codes pour générer automatiquement une documentation à usage technique ou directement pour l'utilisateur final.

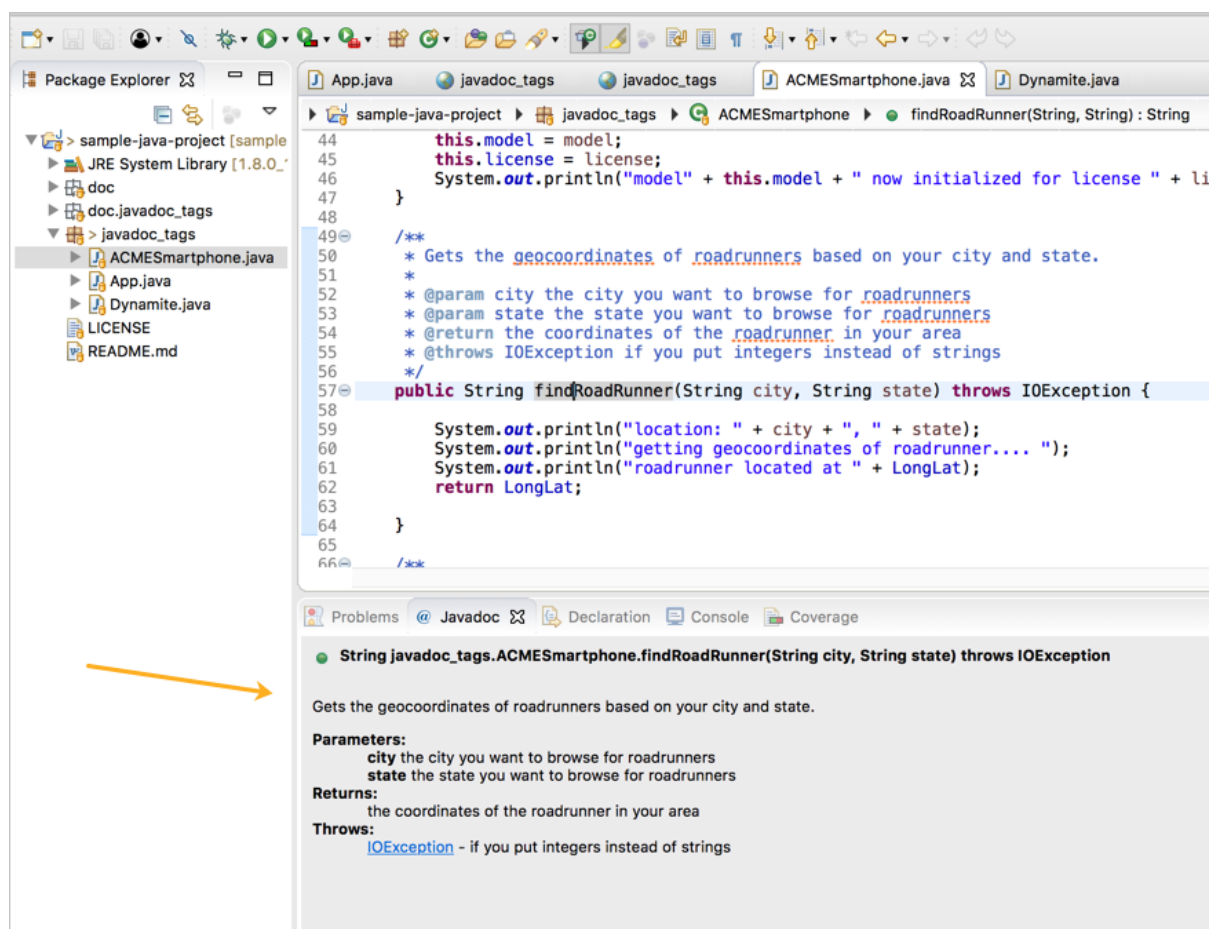
Comment fonctionne un générateur de documentation ?

Le générateur de documentation va extraire, à partir d'un code, tous les commentaires placés dans le code pour les intégrer dans un fichier reprenant tous les éléments de description de code.

Quels sont les différents générateurs de documentation ?

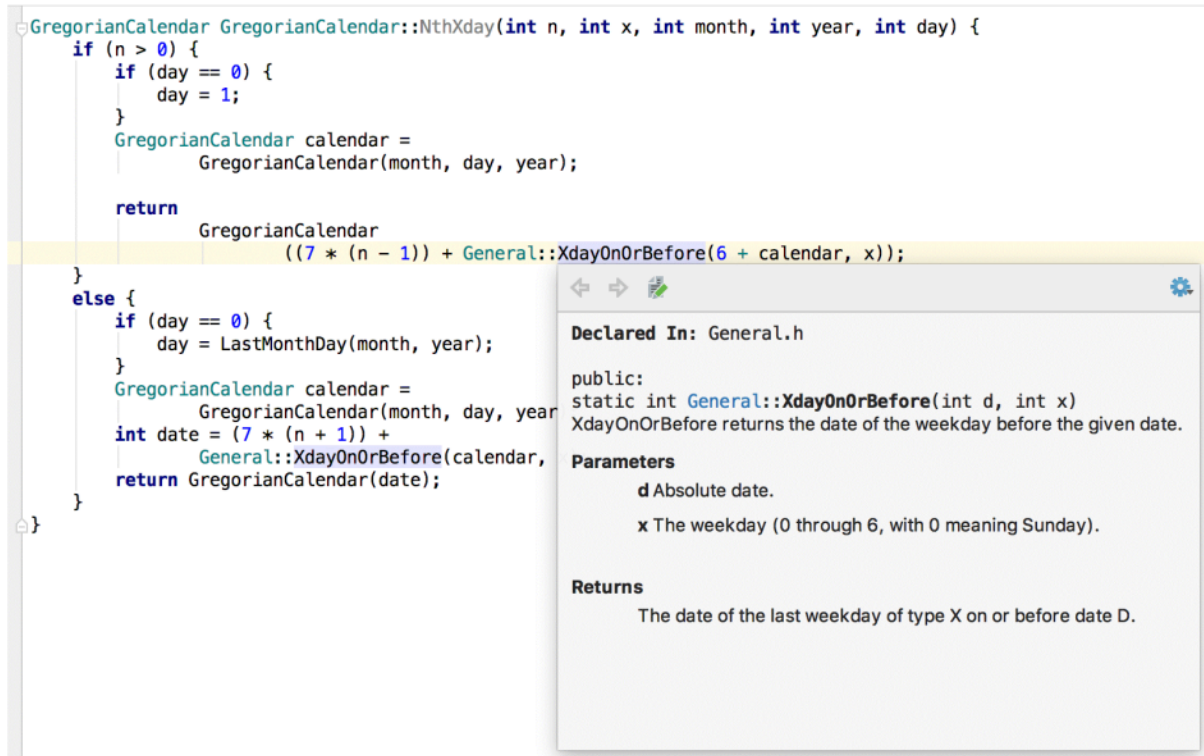
Il existe de nombreux générateurs de documentation ; ils sont généralement développés pour un langage ou une catégorie de langages.

Prenons l'exemple de **javadoc**, un générateur de documentation spécialement développé pour les programmes en Java.



Nous pouvons apercevoir les commentaires de code commençant par « /* » puis terminant par « */ ». Ces commentaires ont été générés par javadoc sur la partie inférieure de la fenêtre représentée par une flèche.

Ce dernier ne prend pas en compte les documentations d'un code développé en C. C'est pour cette raison que le générateur de documentation **doxygen** prend le relais sur ce langage et sur tous les langages de la famille du C (C++, C#, Objective-C).



Remarque Quelle est l'importance de la documentation générée ?

La compilation de documentation générée est extrêmement importante mais reflète également votre discipline à commenter votre code. Il n'existe pas de documentation générée sans commentaires. Et, malheureusement, beaucoup de développeurs ne prennent pas le temps de commenter correctement le code, ce qui produit, le moment venu, une documentation incomplète pouvant poser des problèmes aux utilisateurs ou aux développeurs utilisant la solution.

Lorsqu'on parle de documentation de code, il est impératif de bien penser à comment commenter convenablement son code. C'est vraiment la première étape de la gestion d'un projet de création d'application.

Exercice : Quiz

[solution n°1 p.13]

Question 1

Sur quoi se base la documentation auto-générée ?

- ☐ Les commentaires du code
- ☐ Un assemblage des différentes variables du code pour en créer une documentation

Question 2

Les commentaires ne sont pas possibles sur tous les langages de programmation.

- ☐ Vrai
- ☐ Faux

Question 3

Doxygen est un générateur de documentation principalement orienté sur le langage :

- ☐ Java
- ☐ C
- ☐ Pascal

Question 4

Quel est l'enjeu principal quand nous parlons de documentation auto-générée ?

- ☐ Il faut un code long
- ☐ Il faut des commentaires de code très bien expliqués
- ☐ Il faut que les différentes variables soient bien décrites

Question 5

Trouvez l'intrus.

- ☐ .doc
- ☐ .html
- ☐ .pdf
- ☐ .xml
- ☐ .php

III. Documentation Technique de l'Architecture - DAT

Définition	Qu'est-ce que la DAT ?
-------------------	-------------------------------

La Documentation Technique d'une Architecture est un document reprenant toutes les notions techniques de l'architecture d'un logiciel, d'une application ou autre.

Ce document reprendra également différents points, tels que les investissements financiers nécessaires au développement de telle ou telle fonctionnalité, les calendriers de production avec un diagramme de Gantt et une notion sur la provision pour risque, ainsi que les éléments contractuels entre le prestataire et un client.

Quel est l'objectif d'une DAT ?

Lorsque nous mettons en place une Documentation technique de l'Architecture, il est nécessaire de garder à l'esprit 5 points essentiels pour assurer une bonne documentation :

- Le premier objectif se concentre sur **la vision de votre architecture**. Vous devrez impérativement indiquer quelles sont les parties prenantes sur les différentes couches de votre architecture.
 - Il existe 4 couches d'architecture :
 - La première couche est fonctionnelle ; elle reprend les éléments fonctionnels de votre projet comme les workflows d'application, les chemins de chaque fonction à une autre.
 - La deuxième couche est applicative ; elle base les éléments protocolaires comme l'utilisation du protocole *http* sur tel ou tel composant, etc.
 - La troisième couche se concentre sur l'infrastructure de votre solution ; nous déterminons qui va donc gérer les serveurs physiques, quel type de serveurs physiques nous allons exploiter, etc.
 - La quatrième et dernière couche se concentre sur l'opérationnel, avec les différentes liaisons créées entre les fonctionnalités de l'application finale.
- Le deuxième objectif se concentre sur **les engagements de service**. Lorsque vous mettez en place un système, il est important de déterminer quel service va effectuer quelle tâche et à quelle mesure.
- Le troisième objectif se concentre sur **les réponses de l'architecture aux différents stimuli** de disponibilité, de performance, de résilience, etc.
- Le quatrième objectif et l'un des plus importants : **la gestion du travail à faire**.
 - Chaque travail doit être décrit de façon macro dans un premier temps, afin de pouvoir apporter une visibilité sur la charge de travail à allouer pour ce projet.
 - De plus, la gestion du travail à réaliser permet de donner des métriques au chef de projet opérationnel.
- Le cinquième et dernier point se concentre sur **la communication autour de l'architecture**. Les profils ayant la capacité de lire et de comprendre l'architecture d'un système, d'une application ou d'un logiciel sont peu nombreux ; il est donc fondamental de mettre en place des éléments permettant de présenter proprement le résultat du travail aux différentes équipes prenantes du projet complet.

Conseil

Quelles sont les questions à se poser lors de la création d'une documentation d'architecture technique ?

Lorsque nous mettons en place une DAT, il est nécessaire de se poser plusieurs questions. Nous allons essayer de les voir chacune ensemble et de déterminer les différentes réponses pouvant survenir.

Question 1 : À qui le document est-il destiné ?

Réponse 1 : Aux chefs de projet

Si vous devez créer une documentation destinée aux chefs de projet, il va de soi que vous devrez cadrer votre documentation sur des termes compréhensibles par les chefs de projet. Ces chefs de projet n'ont pas la compétence d'architecture. Il est donc impératif qu'ils puissent avoir les tenants et aboutissants nécessaires à la bonne gestion du projet.

Réponse 2 : Aux administrateurs systèmes et réseaux

Les administrateurs ont tendance à comprendre les documents d'architecture, cependant, il est impératif qu'ils puissent avoir des éléments factuels leur permettant d'agir rapidement en cas d'incident ou en cas de maintenance. Leur communiquer la totalité de l'architecture serait plus inutile qu'autre chose.

Question 2 : Quel est le message à porter ?

Réponse : Le message à porter est forcément en lien avec la destination du document. Vous n'allez pas faire passer le même message à un architecte qu'à un chef de projet. Il est donc impératif de cibler à qui votre document est destiné, et le message à véhiculer sera tout aussi simple.

Remarque

À ce jour, les documentations techniques d'architecture ne sont plus ce qu'elles étaient il y a une dizaine d'années. Depuis l'ère du Wiki et de la documentation en ligne sur des plateformes, il est beaucoup plus simple d'accéder à l'information. Il est beaucoup plus judicieux de découper cette documentation en différentes parties reprenant des éléments par morceau de code, afin que les experts réseaux puissent se concentrer sur cette partie de la documentation pendant que les experts en développement se concentrent sur l'architecture de l'application à utiliser.

Les éléments fondamentaux à une bonne contextualisation

Pour bien contextualiser une documentation technique, il est impératif de répondre aux questions fonctionnelles suivantes :

- Qui sont les acteurs du projet ?
- Par quel pôle le projet est-il suivi ?
- Quelles sont les contraintes métier majeures (pouvant porter atteinte à l'intégrité de l'architecture) ?
- Quel est le parcours du projet ?
- Quelles sont les attentes de ce projet ?

Puis, la question la plus importante :

- Quel est le rôle final de l'application / logiciel / système une fois fini et livré ?

Nous allons donc répondre à ces différentes questions.

Quels sont les acteurs du projet ?

Vous n'êtes pas sans savoir que, dans un projet, il peut y avoir de nombreux partis :

- Des développeurs :
 - Front-End,
 - Back-End,
 - Full Stack,
 - Lead Dev.
- Des chefs de projet :
 - Fonctionnels,
 - Opérationnels.
- Des graphistes :
 - Designers (UI),
 - Ergonomes (UX).
- Des Data Scientists et plein d'autres profils.

Dans la contextualisation, il est impératif de nommer les principaux acteurs du projet et leur périmètre d'action sur ce projet.

Quels sont les pôles suivant le projet ?

Dans une société, il est fréquent d'avoir différents pôles. En fonction de la volumétrie des entreprises, il n'est pas anodin que chaque pôle bénéficie de son propre budget, de sa propre direction et de ses propres effectifs. Bien que chaque pôle possède une certaine indépendance, ils poursuivent le même objectif : servir le siège.

Dans des projets à grande envergure, il est important de rappeler les différents pôles suivant le projet.

Contraintes métier

Certains métiers travaillant sur l'architecture peuvent rencontrer des contraintes, qu'elles soient temporelles, spatiales, informatiques, opérationnelles. Ces contraintes peuvent créer un risque pour l'architecture en cours de création. Il est donc important de mentionner toutes les contraintes fonctionnelles.

Quel est le parcours du projet ?

À l'instant où vous créez votre documentation technique, il n'est pas impossible que le projet ait subi différentes modifications au cours de son développement. Ces événements, aussi infimes soient-ils, doivent être mentionnés afin que chacun puisse avoir une visibilité globale sur la santé actuelle du projet et, surtout, savoir quels ont été les facteurs amenant ces métriques.

Que propose le rendu final ?

Nous sommes davantage dans un contexte de bon sens. Mais il est bien évidemment impératif d'expliquer quel est l'objectif du rendu final du projet en fonction de comment ce dernier se présente : une application, un système, un logiciel, une Infra, etc.

Contextualisation technique

Si nous jetons un œil plus avisé sur ce que nous avons vu ci-dessus, les éléments contextuels que nous avons mis en place sont strictement fonctionnels. Nous parlons des profils travaillant sur le projet, les pôles finançant le projet, les rendus, etc. Mais à aucun moment nous n'avons évoqué la moindre notion technique.

C'est pour la simple et bonne raison que nous mettons toute la contextualisation technique à la suite de la contextualisation fonctionnelle.

Dans la contextualisation technique de votre documentation technique d'architecture, il est important d'ajouter différentes propositions matérielles.

Ces propositions ne seront en rien définitives et seront soumises à un comité avec les différentes parties prenantes, pour juger de la crédibilité et de la pertinence de la configuration proposée.

Cette configuration se crée et gagne en crédibilité, proportionnellement à l'exhaustivité que vous souhaitez intégrer.

Une configuration simple avec un OS, un espace disque, un peu de RAM ne sera absolument pas aussi exploitable qu'une configuration avec ces informations, mais également avec les informations sur le SGBD + sa version, les environnements additionnels installés, les commentaires sur les choix proposés.

Exercice : Quiz

[solution n°2 p.14]

Question 1

Combien existe-t-il de couches dans une architecture ?

- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 7

Question 2

La crédibilité de la contextualisation technique est proportionnelle à l'exhaustivité de celle-ci.

- ☐ Vrai
- ☐ Faux

Question 3

Quelle est la première couche d'une architecture ?

- ☐ Fonctionnelle
- ☐ Applicative
- ☐ Opérationnelle

Question 4

Lorsqu'on parle d'une architecture, nous parlons :

- ☐ D'une application
- ☐ D'un système
- ☐ D'un logiciel
- ☐ De l'un des trois

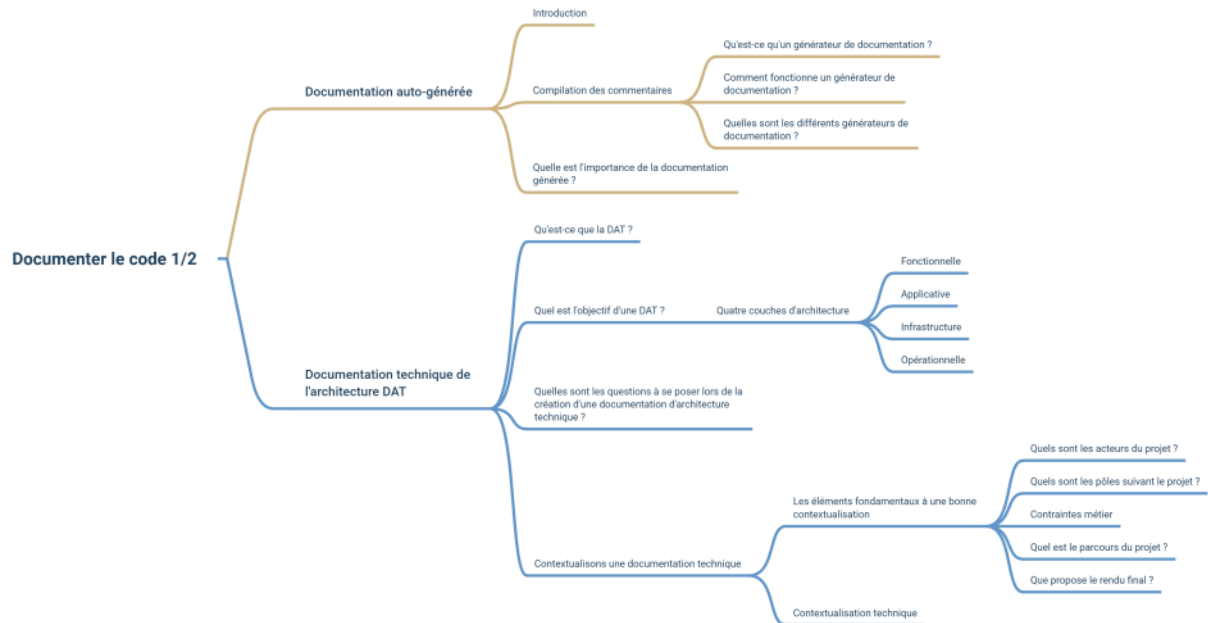
Question 5

Entre technique et fonctionnel, quel est l'élément contextuel à bien différencier ?

- ☐ Les contraintes
- ☐ L'historique
- ☐ Les acteurs

V. Essentiel

Dans ce cours, nous avons vu les éléments fondamentaux à la documentation du code et à la documentation technique. Nous avons évoqué les fondements de la documentation auto-générée ainsi que la documentation d'une architecture technique avec les différents enjeux contextuels techniques et fonctionnels.



VI. Auto-évaluation

A. Exercice

Vous entrez dans une agence de développement informatique. Vous êtes responsable de toute l'ingénierie documentaire de chaque projet. Vous êtes donc responsable de mettre en place les documentations techniques nécessaires au bon déroulement d'un projet. Votre arrivée a provoqué beaucoup de questionnements chez vos collaborateurs. Une session de questions-réponses a donc été organisée afin que vous puissiez expliquer plus en détail quel est votre métier.

Question 1

[solution n°3 p.15]

Pourriez-vous expliquer rapidement ce qu'est la documentation auto-générée ?

Question 2

[solution n°4 p.15]

Pourriez-vous nous expliquer comment se matérialise une architecture ?

B. Test

Exercice 1 : Quiz

[solution n°5 p.15]

Question 1

Javadoc est :

- ☐ Une méthode de documentation technique
- ☐ Un générateur de documentation

Question 2

Il est conseillé de mettre en place une documentation technique :

- ☐ Unique pour tout le monde
- ☐ Divisée pour chaque pôle d'expertise

Question 3

Quel est le prérequis pour la documentation générée ?

- ☐ Des commentaires
- ☐ Du code

Question 4

La documentation technique d'architecture est facultative.

- ☐ Vrai
- ☐ Faux

Question 5


Trouvez l'intrus lorsqu'on parle d'une architecture.

- ☐ Simplicité
- ☐ Performance
- ☐ Intégrité

Solutions des exercices


Exercice p. 5 Solution n°1**Question 1**

Sur quoi se base la documentation auto-générée ?

- ☒ Les commentaires du code
- ☐ Un assemblage des différentes variables du code pour en créer une documentation
-  La documentation auto-générée se base sur les commentaires du code.


Question 2

Les commentaires ne sont pas possibles sur tous les langages de programmation.

- ☐ Vrai
- ☒ Faux
-  Il est possible dans tous langages de programmation de mettre en place des commentaires.


Question 3

Doxygen est un générateur de documentation principalement orienté sur le langage :

- ☐ Java
- ☒ C
- ☐ Pascal
-  Le générateur de documentation de Doxygen est principalement orienté sur le langage C.

Question 4

Quel est l'enjeu principal quand nous parlons de documentation auto-générée ?


- ☐ Il faut un code long
- ☒ Il faut des commentaires de code très bien expliqués
- ☐ Il faut que les différentes variables soient bien décrites
-  Lorsque l'on parle de documentation auto-générée, il est impératif de mettre des commentaires de code très précis.

Question 5

Trouvez l'intrus.

- ☐ .doc
- ☐ .html
- ☐ .pdf
- ☐ .xml

☒ .php

 Le .php est l'intrus, il n'est en effet pas un format d'export d'une documentation générée.

Exercice p. 9 Solution n°2

Question 1


Combien existe-t-il de couches dans une architecture ?

☐ 3

☒ 4

☐ 5

☐ 7


 Une architecture possède 4 couches.

Question 2

La crédibilité de la contextualisation technique est proportionnelle à l'exhaustivité de celle-ci.

☒ Vrai

☐ Faux

 Plus l'exhaustivité de la documentation technique est avancée, plus la crédibilité de celle-ci est élevée.

Question 3

Quelle est la première couche d'une architecture ?

☒ Fonctionnelle

☐ Applicative

☐ Opérationnelle

 La première couche d'une architecture est la couche fonctionnelle.

Question 4


Lorsqu'on parle d'une architecture, nous parlons :

☐ D'une application

☐ D'un système


☐ D'un logiciel

☒ De l'un des trois

 Nous parlons de l'un des trois ; une architecture est nécessaire pour toutes les typologies de développement.

Question 5

Entre technique et fonctionnel, quel est l'élément contextuel à bien différencier ?

- ☒ Les contraintes
- ☐ L'historique
- ☐ Les acteurs
-  Les contraintes sont bien à séparer entre les contraintes métier qui sont fonctionnelles et les contraintes techniques.

p. 11 Solution n°3


En tant que développeurs, nos clients vous demandent de mettre en place une application, un logiciel ou autre. Votre rôle sera donc de développer les fonctionnalités pour qu'elles puissent être utilisables. Vous créez donc des lignes de code. En tant que bons développeurs, il est normal que vous mettiez en place des commentaires sur vos codes. Ces commentaires vont alors donner des informations sur les lignes que vous avez composées. Ces lignes seront réinterprétées par un générateur de documentation pour rendre automatiquement un document reprenant tous les commentaires de votre code pour vous le mettre en page.

p. 11 Solution n°4

Dans un premier temps, il est important d'expliquer qu'une architecture peut être mise en place pour un système, un logiciel, une application, etc. Cette architecture se matérialise en 4 couches. La première couche est la couche fonctionnelle, déterminant les fonctionnalités souhaitées dans le rendu final. La deuxième couche est la couche applicative ; cette couche va mettre en exergue les différentes utilisations des différents protocoles utilisés par le projet. La troisième couche est la couche d'infrastructure qui permet de déterminer les différentes configurations matérielles et logicielles avec des informations déterminantes sur l'utilisation des ressources. Pour finir, la quatrième et dernière couche de l'architecture est la couche opérationnelle, permettant d'indiquer comment la solution finale va utiliser tous ses organes.


Exercice p. 11 Solution n°5**Question 1**

Javadoc est :

- ☐ Une méthode de documentation technique
- ☒ Un générateur de documentation
-  Javadoc est un générateur de documentation.


Question 2

Il est conseillé de mettre en place une documentation technique :

- ☐ Unique pour tout le monde
- ☒ Divisée pour chaque pôle d'expertise
-  Il est conseillé de mettre en place une documentation technique divisée pour chaque pôle d'expertise.


Question 3

Quel est le prérequis pour la documentation générée ?

- ☒ Des commentaires
- ☐ Du code
-  Pour bénéficier d'une documentation générée, il est nécessaire d'avoir des commentaires.


Question 4

La documentation technique d'architecture est facultative.

- ☐ Vrai
- ☒ Faux
-  La documentation technique n'est absolument pas facultative et se présente même comme un fondamental d'un projet.

Question 5

Trouvez l'intrus lorsqu'on parle d'une architecture.

- ☒ Simplicité
- ☐ Performance
- ☐ Intégrité
-  Lorsqu'on parle d'une architecture, nous attendons de celle-ci qu'elle soit performante et intégrale. La simplicité n'est pas un paramètre à prendre en compte.