

# **SOLID, TIENS-TOI BIEN !**

**11 patrons comportementaux en BD**



*De petites histoires  
pour apprendre et retenir  
les design patterns !*

**LAURIER Vincent**

**Illustré par ETIENNE Olivier**

# Solid, tiens-toi bien !

11 patrons comportementaux en BD

LAURIER Vincent et ETIENNE Olivier

Ce livre est en vente à <http://leanpub.com/solid-tiens-toi-bien>

Version publiée le 2021-01-09



Ce livre est publié par [Leanpub](#). Leanpub permet aux auteurs et aux éditeurs de bénéficier du Lean Publishing. [Lean Publishing](#) consiste à publier à l'aide d'outils très simples de nombreuses itérations d'un livre électronique en cours de rédaction, d'obtenir des retours et commentaires des lecteurs afin d'améliorer le livre.

© 2014 - 2020 Laurier Vincent, Etienne Olivier

# Tweet ce livre !

S'il vous plaît aidez LAURIER Vincent et ETIENNE Olivier en parlant de ce livre sur [Twitter](#) !

Le tweet suggéré pour ce livre est :

Je viens d'acheter une bande dessinée sur les design patterns. Ça m'a l'air original, j'ai hâte d'en savoir plus !

Le hashtag suggéré pour ce livre est [#BehaveSolid](#).

Découvrez ce que les gens disent à propos du livre en cliquant sur ce lien pour rechercher ce hashtag sur Twitter :

[#BehaveSolid](#)

# Table des matières

<b>Préface</b> . . . . .	<b>1</b>
Le mot de l’auteur . . . . .	1
Le mot de l’illustrateur . . . . .	2
<b>À propos du livre</b> . . . . .	<b>3</b>
Le public visé . . . . .	3
L’indépendance des chapitres . . . . .	3
La partie pour “enfants” . . . . .	3
Les exemples d’implémentation . . . . .	3
Les diagrammes UML . . . . .	4
Les réponses pour développeurs . . . . .	4
<b>À propos de l’auteur et de l’illustrateur</b> . . . . .	<b>5</b>
<b>Remerciements</b> . . . . .	<b>6</b>
<b>La méthode modèle</b> . . . . .	<b>1</b>
<i>“Gepetto fait le spectacle”</i> . . . . .	1
L’analogie avec la BD . . . . .	8
Exemple d’implémentation . . . . .	9
Modélisation UML . . . . .	11
Réponses pour développeurs . . . . .	12
L’inversion de contrôle . . . . .	13
<b>L’itérateur</b> . . . . .	<b>15</b>
<i>“Jour d’inventaire”</i> . . . . .	15
L’analogie avec la BD . . . . .	15
Exemple d’implémentation . . . . .	15
Modélisation UML . . . . .	15
Réponses pour développeurs . . . . .	15
Les itérateurs de la SPL . . . . .	15
<b>La commande</b> . . . . .	<b>16</b>
<i>“Sous la porte de la loge”</i> . . . . .	16
L’analogie avec la BD . . . . .	16



## TABLE DES MATIÈRES

Exemple d'implémentation . . . . .	16
Modélisation UML . . . . .	16
Réponses pour développeurs . . . . .	16
La macrocommande . . . . .	16
<b>L'interpréteur . . . . .</b>	<b>17</b>
<i>"Le langage des signes de Gepetto"</i> . . . . .	17
L'analogie avec la BD . . . . .	17
Exemple d'implémentation . . . . .	17
Modélisation UML . . . . .	17
Réponses pour développeurs . . . . .	17
BNF et EBNF . . . . .	17
<b>La stratégie . . . . .</b>	<b>18</b>
<i>"Peu importe l'accessoire !"</i> . . . . .	18
L'analogie avec la BD . . . . .	18
Exemple d'implémentation . . . . .	18
Modélisation UML . . . . .	18
Réponses pour développeurs . . . . .	18
Null Object . . . . .	18
<b>Le visiteur . . . . .</b>	<b>19</b>
<i>"La visite du parc animalier"</i> . . . . .	19
L'idée générale . . . . .	19
Exemple d'implémentation . . . . .	19
Modélisation UML . . . . .	19
Réponses pour développeurs . . . . .	19
Principe ouvert/fermé . . . . .	19
<b>La chaîne de responsabilités . . . . .</b>	<b>20</b>
<i>"A qui le tour ?"</i> . . . . .	20
L'analogie avec la BD . . . . .	20
Exemple d'implémentation . . . . .	20
Modélisation UML . . . . .	20
Réponses pour développeurs . . . . .	20
<b>L'observateur . . . . .</b>	<b>21</b>
<i>"Torp se la joue un peu..."</i> . . . . .	21
L'analogie avec la BD . . . . .	21
Exemple d'implémentation . . . . .	21
Modélisation UML . . . . .	21
Réponses pour développeurs . . . . .	21
Principe d'inversion des dépendances . . . . .	21

## TABLE DES MATIÈRES

<b>Le médiateur</b> . . . . .	<b>22</b>
<i>“Gepetto contrôle l’ouverture du parc”</i> . . . . .	22
L’analogie avec la BD . . . . .	22
Exemple d’implémentation . . . . .	22
Modélisation UML . . . . .	22
Réponses pour développeurs . . . . .	22
<b>L’état</b> . . . . .	<b>23</b>
<i>“La piste dans tous ses états”</i> . . . . .	23
L’analogie avec la BD . . . . .	23
Exemple d’implémentation . . . . .	23
Modélisation UML . . . . .	23
Réponses pour développeurs . . . . .	23
<b>Memento</b> . . . . .	<b>24</b>
<i>“Shoot photo pour Solid”</i> . . . . .	24
L’analogie avec la BD . . . . .	24
Exemple d’implémentation . . . . .	24
Modélisation UML . . . . .	24
Réponses pour développeurs . . . . .	24
<b>Annexes</b> . . . . .	<b>25</b>
Le principe d’Hollywood . . . . .	25
Itérateur pour filter sur les récompenses . . . . .	25
Exemple d’implémentation de l’expression Enchaîner . . . . .	25
<b>Bibliographie</b> . . . . .	<b>26</b>

# Préface

## Le mot de l'auteur

*“Solid, tiens-toi bien !”, c’est à mes yeux bien plus qu’un livre !*

Ce fût d’abord un concept, puis un projet, et enfin une histoire.

## Un concept

2014 : l’année où je me décide à poser sur le papier un ensemble d’idées, de projets entrepreneuriaux divers. Ils reviennent. Je les évoque fréquemment, mais jamais je ne passe réellement à l’action ! Alors, je liste des critères, une dizaine probablement, sur une feuille de papier égarée depuis. Chaque projet passera au crible de cet examen, et seul “Solid” y résistera !

Je ne saurais en retrouver l’intégralité, mais je me souviens des prérequis les plus importants pour moi à l’époque : pas d’investissement financier, un projet dématérialisé, fait en collaboration, et qui exploite au mieux les compétences de chacun. C’est décidé, j’embarque Olivier mon meilleur ami dans l’aventure ! Il exploitera sa passion pour le dessin ; j’utiliserai mes compétences en programmation. Le projet avait aussi su faire voler en éclats d’autres prérequis que j’avais posés :

- originalité : une bande dessinée pour enfants pour parler des patrons de conception, vraiment ?
- presque intemporel : “Design Patterns : Elements of Reusable Object-Oriented Software”, le livre du Gang Of Four date de 1994, et reste à ce jour une référence !

## Un projet

L’idée est bonne, le choix est fait. Du concept, on passe au projet.

Points réguliers, croquis, recherches graphiques, storyboards, premières implémentations de code, élaboration de la couverture, choix du titre... tout y passe, et l’on travaille sérieusement avec régularité. Pour ceux au passage qui se poseraient des questions concernant le titre : “Solid, tiens-toi bien !”, c’est pour faire écho au terme “comportementaux” du sous-titre.

Un an passe ! Mais le sort s’en mêle. De sérieux problèmes de santé m’empêchent de travailler, et Olivier, ayant aussi sa propre vie, m’annonce qu’il arrête le projet.

## Une histoire

260 pages déjà écrites, un livre illustré à 40% environ, “Solid, tiens-toi bien !” s’arrête là en mars 2016. Olivier et moi gardons un contact beaucoup plus épisodique. Le spectre d’un projet inachevé planera plusieurs années...

2019 : mes problèmes de santé étant derrière moi, j’ai repris un poste de développeur, et je parle parfois à mes collègues de ce projet, de l’idée, du concept. Les retours encourageants me laissent un goût amer sur la façon dont tout cela s’est arrêté. Je décide de m’octroyer du temps pour m’y remettre, même seul. J’apprendrai à dessiner !

Puis le coup du destin : alors que je réalise, que seul, je n’aboutirai pas, voici qu’Olivier est de nouveau disponible et motivé. On s’y remet avec un regard nouveau.

260 pages ? Bien trop long ! L’esprit de simplicité voulue était perdu. Une deuxième mouture doit voir le jour, et l’on s’y attèle sans la moindre difficulté. Nos mécanismes de travail sont là, comme si l’on reprenait le travail d’hier ! Magique !

5 octobre 2020, alors que j’écris ces lignes, nous prévoyons de sortir le livre pour la fin novembre. Ce sera aussi bien un aboutissement, que le démarrage d’une aventure !

## Le mot de l’illustrateur

J’ai toujours été passionné par la programmation, mais jusqu’alors mon approche était procédurale. Ce qui m’importait avant tout, c’était la réalisation de tâches concrètes.

À mesure que j’illustrais et relisais l’ouvrage, je comprenais peu à peu les concepts de base de l’orienté-objet, l’utilité des interfaces et des classes abstraites. En ce sens, le livre m’a fait évoluer en tant que programmeur.

Grâce à la souplesse de notre collaboration en équipe, chaque chapitre m’aura permis sur le plan artistique, d’explorer continuellement des techniques et styles graphiques variés.

En définitive, j’ai vraiment aimé travailler sur ce livre. Il m’aura permis de publier ma première bande dessinée, ce qui pour moi, avait toujours été un rêve !

# À propos du livre

## Le public visé

J'aimerais croire qu'il sera suffisamment large. C'est un livre qui s'adresse normalement aux développeurs, mais l'effort de vulgarisation a été poussé jusqu'au défi ! L'idée qu'un enfant en bas âge puisse appréhender les bandes dessinées, et en comprendre un jour toute la teneur, me fait sourire...

Mais ne vous y trompez pas, c'est bien à des informaticiens qu'il s'adresse ! Il suppose que vous connaissiez les concepts de la programmation objet, car ceux-ci ne sont pas expliqués, mais juste utilisés dans un contexte simple. Les programmeurs PHP seront nécessairement avantagés, mais les autres ne devraient pas être perdus sur l'essentiel.

Pour les lecteurs plus avancés, j'espère qu'il pourra servir d'appui pédagogique. Et enfin à tous, de support mnémotechnique !

## L'indépendance des chapitres

*"Solid, tiens-toi bien !"*, c'est avant tout onze chapitres indépendants. Chacun d'eux traite d'un unique patron de conception. Il est fait référence dans de rares cas à une autre partie du livre, mais globalement, vous êtes libre sur l'ordre de lecture.

## La partie pour "enfants"

Chaque chapitre commence par la bande dessinée, suivie de questions/réponses. Ces trois premières sections sont accessibles à un jeune public. J'invite cependant le développeur à lire même la dernière d'entre elles : les réponses enfantines. Elles sont courtes, et attirent l'attention sur des éléments clés de l'histoire.

## Les exemples d'implémentation

J'ai fait le choix de ne pas suivre les standards. La raison est simple : j'ai voulu optimiser la facilité de compréhension. Aussi, vous rencontrerez des noms de méthodes, propriétés, ou classes, écrites en français dès que possible ; l'idée étant d'éviter une gymnastique intellectuelle inutile de traduction, et de faire rapidement les correspondances avec l'histoire. Une notation en *"snake\_case"* plutôt qu'en *"CamelCase"* a aussi été choisie pour une facilité accrue de lecture. Enfin, j'ai évité d'alourdir la syntaxe du code par un typage strict systématique. Cela permettra à ceux qui ne sont pas familiers au PHP d'aller plus rapidement à l'essentiel. En résumé, la compréhension a été préférée au respect des standards.

## Les diagrammes UML

Conséquence de la vulgarisation expliquée plus haut, les diagrammes UML du GoF ont été “adaptés” au cas par cas. Tantôt ils reprennent les termes en anglais génériques, car suffisamment transparents et compréhensibles, tantôt j’y introduis des expressions françaises pour faciliter les correspondances. Uniformiser aurait été idéal, mais je pense avoir conservé ce qui marchait le mieux. Les diagrammes UML ont été placés à la suite directe de l’implémentation. Ils offrent à ce stade une vue globale résumant l’essentiel. Des éléments numérotés vous faciliteront la lecture, quand bien même les notations UML vous sont étrangères.

## Les réponses pour développeurs

Cette session du chapitre répond aux questions du début, mais en utilisant les termes techniques préférentiellement. Elle sert à asseoir la bonne compréhension du patron. Parfois, elle ouvre sur plus de détails d’implémentation, parfois sur une réflexion théorique. J’ai finalement voulu la conserver relativement courte, le principe du livre étant de comprendre rapidement l’essence de chaque design pattern.

# À propos de l’auteur et de l’illustrateur

Développeur PHP depuis 2009, **LAURIER Vincent** travaille principalement avec le framework Symfony. Ancien professeur de mathématiques/physique, il a conservé son goût pour la pédagogie, et la transmission du savoir. Son premier livre “Solid, tiens-toi bien !” illustre à merveille son effort pour la vulgarisation des connaissances. La conception, l’élégance et la qualité du code sont des problématiques qui lui tiennent à cœur.

**ETIENNE Olivier** est un autodidacte curieux et passionné. Entre autre la programmation et la musique, c’est le dessin qui occupe une place prépondérante dans sa vie depuis son enfance. Seul, il a fait de Blender, Inkscape et Gimp ses armes open source principales. Aujourd’hui, il met son talent d’illustrateur au service d’un projet pédagogique, à la croisée des domaines qu’il affectionne.

# Remerciements

## De l'auteur

*À Olivier, pour son amitié, sa positivité, son travail.*

*À ma femme, pour son soutien et sa patience. C'est grâce à elle que tout a commencé.*

*À ma mère, pour sa relecture attentive.*

*Aux amis, collègues et famille, pour leurs encouragements.*

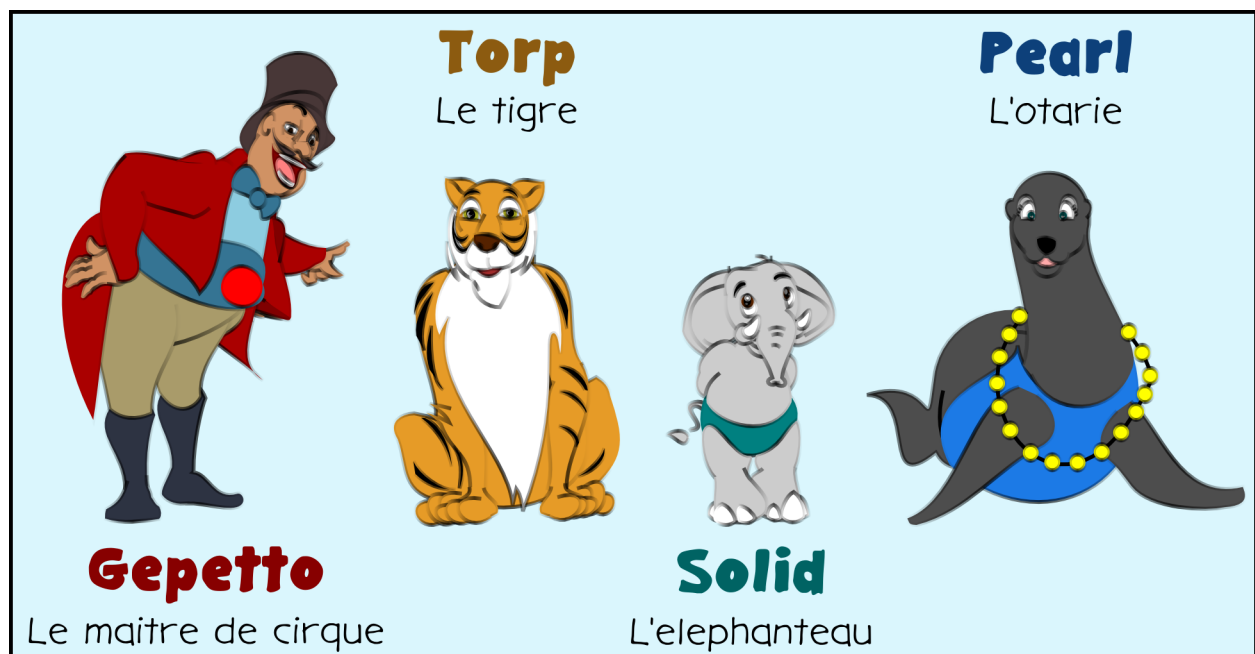
## De l'illustrateur

*À mes fils, Nathan et Jérôme, qui auront su faire évoluer leur professeur de dessin.*

*À Vincent, pour ce que m'apporte au quotidien notre amitié.*

*À ma famille, pour leur support inébranlable.*

**Nous vous laissons maintenant en compagnie des vedettes du livre...**





# La méthode modèle

## Le but du patron selon le GOF

Définir le squelette d'un algorithme au sein d'une opération, en déléguant des étapes aux sous-classes. La méthode modèle laisse les sous-classes redéfinir certaines étapes d'un algorithme sans changer la structure de celui-ci.<sup>a</sup>

a. Traduction personnelle de la définition anglaise originale : Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure.

Expliquons cette définition à l'aide d'une petite histoire illustrée :

## ***"Gepetto fait le spectacle"***



Ce jour-là au cirque, c'était répétition générale ! Le spectacle du samedi approchait. Gepetto devait s'assurer que tout le monde était prêt.



*Solid se précipita pour déposer son tabouret...*



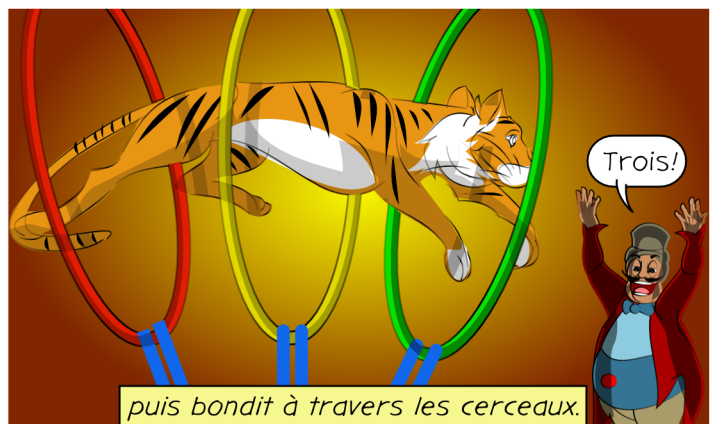
*...si bien qu'il oublia de soigner son entrée en scène.*



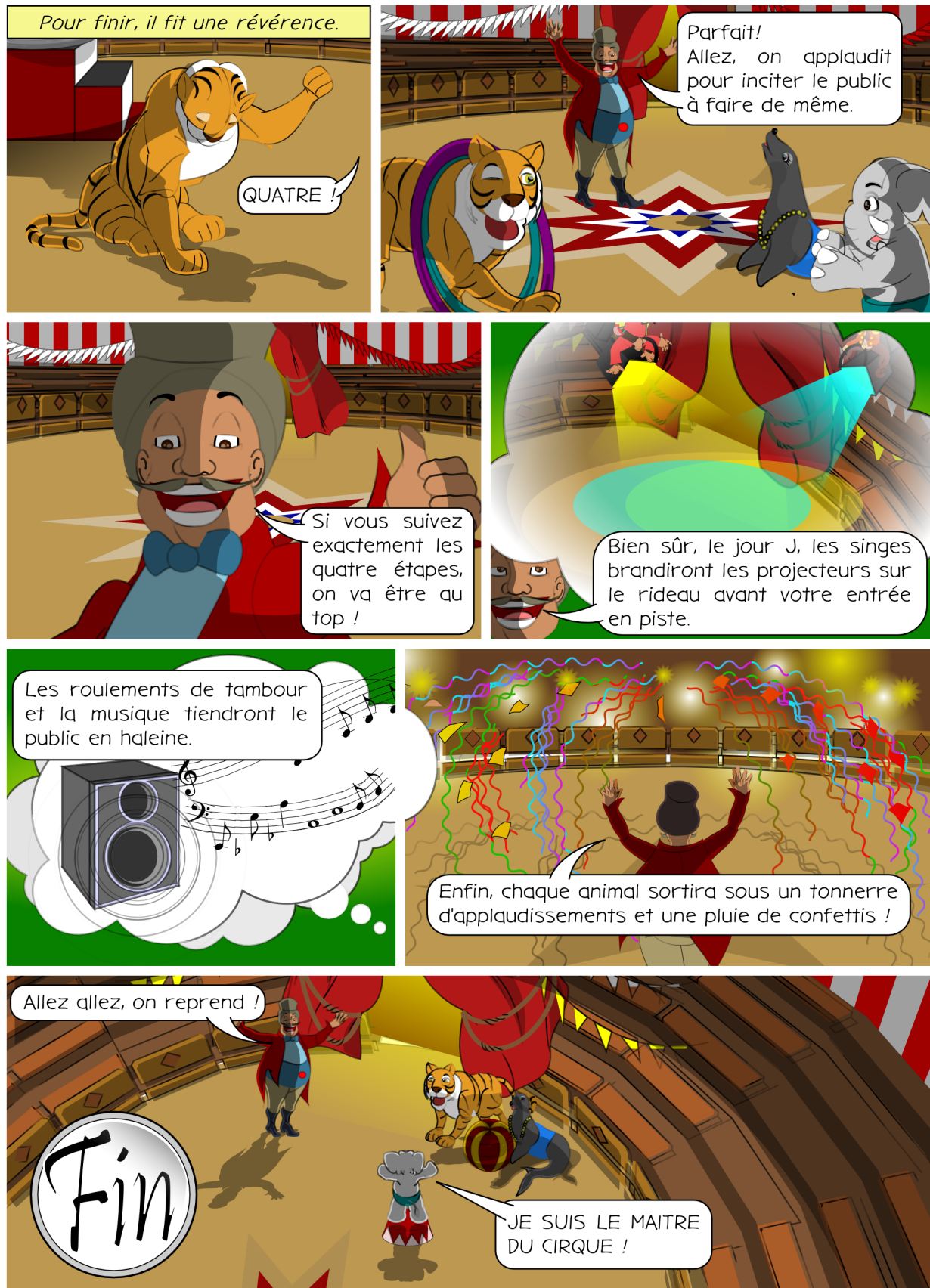
*Solid enchaina ses acrobaties et termina par une révérence.*

















 Questions Globales


- ① Quel est le but de la séance d'aujourd'hui au cirque ?
- ② Pourquoi Gepetto est-il obligé de montrer l'exemple ?


 Questions spécifiques

- ③ Quelles étapes doivent suivre les animaux pour faire leur spectacle ?
- ④ Gepetto a-t-il du matériel pour réaliser un numéro ?
- ⑤ Quel animal réalise une prestation parfaite ?
- ⑥ Le jour J, il y aura des petits plus par rapport aux répétitions. De quoi s'agit-il ?

 À ton avis

- ⑦ Regarde la dernière vignette. Selon toi, pourquoi Solid crie-t-il : « Je suis le maître du cirque ! » ?





## Questions Globales

- ① Aujourd'hui, le but est de faire une séance de répétition pour les animaux.
- ② Gepetto doit montrer l'exemple car Solid et Pearl ont oublié une étape dans leur spectacle.

## Questions spécifiques

- ③ Pour faire leur spectacle, les animaux doivent suivre 4 étapes : déposer leur matériel, saluer le public, faire leur acrobatie, puis faire une révérence.
- ④ Non, Gepetto ne possède pas de matériel pour faire une acrobatie. Il montre donc simplement l'endroit au sol où les animaux devront déposer le leur.
- ⑤ C'est Torp qui réalise une prestation parfaite, car il suit l'ensemble des étapes.
- ⑥ Lors du vrai spectacle, la lumière et la musique accompagneront l'entrée de chaque animal. Les confettis et les applaudissements suivront la sortie de chacun.

## À ton avis

- ⑦ On peut penser que Solid crie « Je suis le maître du cirque ! » pour taquiner Gepetto. Il est un peu espiègle, et aime croire qu'il est aux commandes, même si ce n'est pas tout à fait vrai...

## L'analogie avec la BD

### L'intérêt



Au début de l'histoire, Solid et Pearl se produisent librement sur scène. Malheureusement, l'un oublie de saluer le public en entrant, et l'autre de faire une révérence à la fin. Gepetto sent donc le besoin d'imposer un cadre aux opérations. C'est pourquoi il va mettre en place un patron de conception : la **méthode modèle** (habituellement traduit par "patron de méthode").



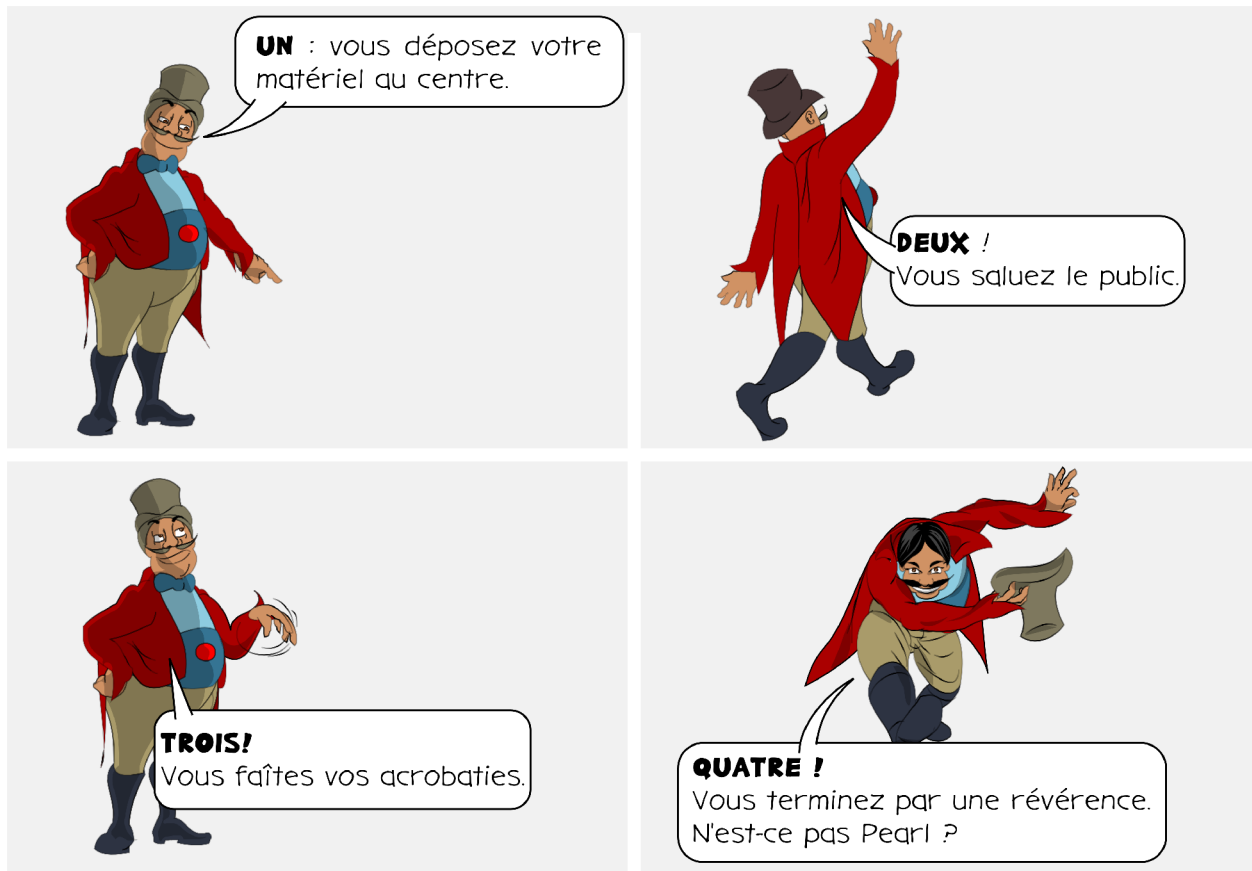
Si l'on rapproche la BD au monde du développement, c'est comme si vous aviez laissé le soin à deux développeurs (Solid et Pearl) d'implémenter une fonctionnalité. Ils n'auraient codé que trois étapes, alors que quatre étaient attendues dans un ordre précis. Vous allez donc devoir jouer le rôle de Gepetto : voyons comment !

### L'idée centrale

Le maître de cirque indique la marche à suivre dans une "**méthode modèle**" que l'on pourra nommer : "***se produire sur scène***". Elle sera le point d'entrée de notre classe.

Observez les quatre étapes ci-dessous. Trouvez en deux qui soient concrètes, c'est-à-dire où Gepetto montre (implémente) exactement ce qu'il faut faire. Essayez de comprendre pourquoi les deux autres pourraient être considérées comme abstraites.





Vous avez trouvé ?

Gepetto n'a ni matériel à déposer, ni acrobatie à réaliser ! Ce seront les deux étapes abstraites à définir par les animaux.

## Exemple d'implémentation

```
abstract class Spectacle_abstrait
{
    // Gepetto veut que les animaux suivent strictement sa façon de faire.
    // En déclarant "final" la méthode, on interdit toute surcharge.
    // Étant publique, c'est néanmoins un point d'entrée de la classe.
    final public function se_produire_sur_scène()
    {
        // Voici le squelette de l'algorithme: 4 étapes dans la BD.
        $this->déposer_son_matériel();
        $this->saluer_le_public();
        $this->faire_son_acrobatie();
        $this->faire_une_révérence();
    }
}
```

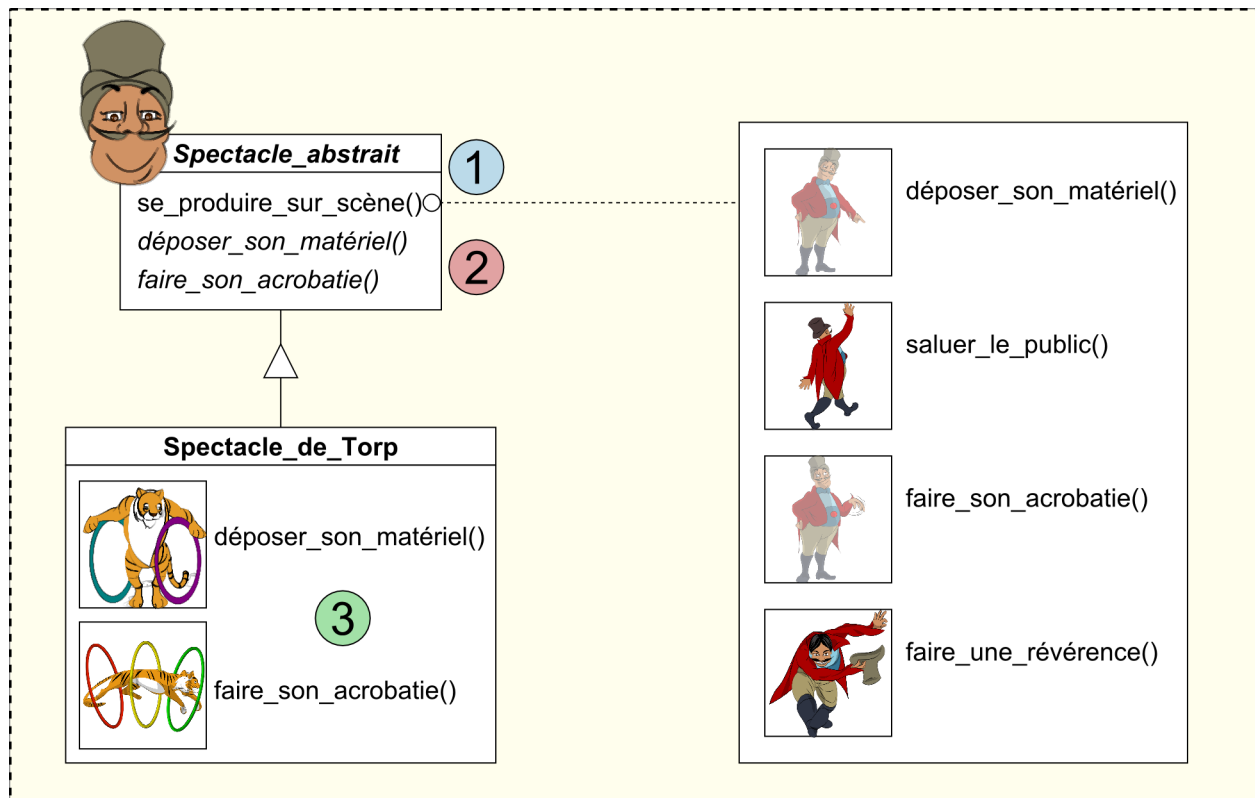
```
}
// Gepetto n'a pas de matériel, donc ne proposera pas d'implémentation.
// Le mot clé "abstract" forcera par contre les classes filles à en proposer une.
// Et nous choisissons une visibilité protégée afin qu'elles puissent le faire.
abstract protected function déposer_son_materiel();

private function saluer_le_public()
{
    echo "Un tour de piste est nécessaire pour saluer le public.";
}
// Gepetto n'a pas d'acrobatie.
abstract protected function faire_son_acrobatie();

private function faire_une_révèrence()
{
    echo 'Une révérence conclut la prestation.';
}
}
// Torp suit l'exemple donné par Gepetto, en étendant la classe abstraite.
// Il ne personnalise que les étapes nécessaires.
class Spectacle_de_Torp extends Spectacle_abstrait
{
    protected function déposer_son_materiel()
    {
        echo 'Torp dépose ses cerceaux.';
    }

    protected function faire_son_acrobatie()
    {
        echo 'Il saute ensuite à travers les cerceaux. Spendide !';
    }
}
// Sa prestation suit ainsi l'enchaînement défini.
(new Spectacle_de_Torp())->se_produire_sur_scène();
```

## Modélisation UML



1. Gepetto propose une méthode modèle : `se_produire_sur_scène()` dont une pseudo implémentation est visible sur la droite.
2. Certaines étapes sont abstraites dans la classe mère.
3. Les classes filles, par exemple le spectacle de Torp, doivent donc en proposer une implémentation concrète.

# Réponses pour développeurs

## Questions globales

1) *Quel est le but de la séance d'aujourd'hui au cirque ?*

Le but, c'est de s'assurer que les animaux soient prêts pour le spectacle, autrement dit, qu'ils vont suivre un cheminement parfaitement déterminé. Rappelez-vous donc de la chose suivante : il est intéressant de **mettre en place le patron "méthode modèle" pour imposer un cadre au déroulement d'une opération.**

2) *Pourquoi Gepetto est-il obligé de montrer l'exemple ?*

Pour éviter que les animaux ne se trompent ! Par le biais de l'héritage, on crée instantanément de la rigidité.

## Questions spécifiques

3) *Quelles étapes doivent suivre les animaux pour faire leur spectacle ?*

Elles sont au nombre de quatre : déposer son matériel, saluer, faire son numéro, puis sa révérence. Elles ne font pas partie de l'API publique de la classe abstraite, mais décrivent le fonctionnement interne du spectacle. Vous noterez donc leurs visibilité protégées ou privées dans l'exemple de code.

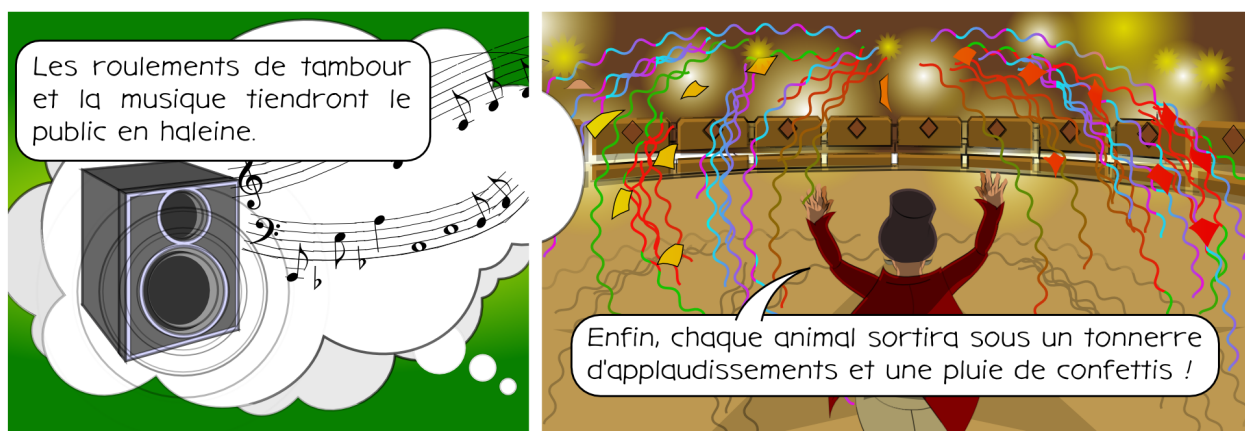
4) *Gepetto a-t-il du matériel pour réaliser un numéro ?*

Non ! Et c'est pourquoi "déposer son matériel" ou "faire son acrobatie" sont les méthodes abstraites. Le maître de cirque représente lui-même par conséquence une classe abstraite.

5) *Quel animal réalise une prestation parfaite ?*

Torp est celui qui va étendre la classe proposée par Gepetto. Le gain est alors immédiat ! Quand il réalise sa prestation, il est contraint de suivre les quatre étapes dans l'ordre défini.

6) *Le jour J, il y aura des petits plus par rapport aux répétitions. De quoi s'agit-il ?*



Le jour J il faudra agrémenter les spectacles

Il s'agit par exemple de la musique ou des confettis lors des entrées et sorties de piste. On soulève par ce biais une question sur le patron de conception “**méthode modèle**”. Comment fait-on si l'on souhaite plus tardivement ajouter de la logique dans le déroulement des opérations ? Pour que cela soit possible, il faut que la classe abstraite l'ait prévu. Observez l'exemple de code qui suit.

```
public function se_produire_sur_scène()
{
    // Pour anticiper des besoins ultérieurs de personnalisation,
    // l'on peut prévoir d'ajouter des "étapes vides"...
    $this->en_entrant_sur_piste();

    $this->déposer_son_materiel();
    $this->saluer_le_public();
    $this->faire_son_acrobatie();
    $this->faire_une_révèrence();

    // ...au début, comme à la fin !
    $this->en_sortant_de_piste();
}

// On appelle cela des "hooks". Ce sont des méthodes, qui par défaut ne font rien,
// mais qui sont surchargeables pour ajouter au besoin de la logique.
protected function en_entrant_sur_piste(){};
protected function en_sortant_de_piste(){};
```

À ton avis

7) Regarde la dernière vignette. Selon toi, pourquoi Solid crie-t-il : « Je suis le maître du cirque ! » ?

## L'inversion de contrôle



Solid aime bien sentir qu'il a du pouvoir, qu'il peut faire ce qu'il veut. Mais en réalité, Gepetto dicte comment les choses se déroulent.

Le patron "**méthode modèle**" suit le principe que l'on dit "**d'inversion de contrôle**". Pourquoi ? Imaginez que Solid représente une classe de votre application, et le maître de cirque, celle d'un framework utilisé. En étendant la classe du framework, on s'affranchit d'une certaine dose de travail, mais l'on perd la main sur au moins deux choses :

- le déroulement de l'algorithme, qui est alors dicté par le framework
- ce qu'il est possible de personnaliser, contrôlé aussi par le framework

Les plus curieux iront lire pourquoi "**l'inversion de contrôle**" est aussi connu sous le terme de **principe d'Hollywood**.

[Voir en annexe page 136.](#)

# L'itérateur

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***"Jour d'inventaire"***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **L'analogie avec la BD**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Exemple d'implémentation**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Modélisation UML**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Réponses pour développeurs**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Les itérateurs de la SPL**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# La commande

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“Sous la porte de la loge”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L’analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d’implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## La macrocommande

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.



# L'interpréteur

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“Le langage des signes de Gepetto”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L'analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d'implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## BNF et EBNF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# La stratégie

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***"Peu importe l'accessoire!"***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L'analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d'implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Null Object

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Le visiteur

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“La visite du parc animalier”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **L’idée générale**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Exemple d’implémentation**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Modélisation UML**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Réponses pour développeurs**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Principe ouvert/fermé**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# La chaîne de responsabilités

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“A qui le tour ?”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **L’analogie avec la BD**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Exemple d’implémentation**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Modélisation UML**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## **Réponses pour développeurs**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# L'observateur

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“Torp se la joue un peu...”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L'analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d'implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Principe d'inversion des dépendances

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Le médiateur

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“Gepetto contrôle l’ouverture du parc”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L’analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d’implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# L'état

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“La piste dans tous ses états”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# L'analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Exemple d'implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Mémento

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## ***“Shoot photo pour Solid”***

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## L’analogie avec la BD

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d’implémentation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Modélisation UML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Réponses pour développeurs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.



# Annexes

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Le principe d'Hollywood

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Itérateur pour filter sur les récompenses

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Exemple d'implémentation de l'expression Enchaîner

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

# Bibliographie

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.

## Utilisation de ressources externes

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/solid-tiens-toi-bien>.