

Créer un comportement (construire une fonction = un traitement)

4. Je reçois QUOI ? de quelle NATURE ? Bien examiner cette NATURE (c'est le produit de QUOI, ça viens d'OU) (ou TYPE si le langage est typé)

6

- 1. Je veux faire QUOI?
- 2. Une méthode pré-définie par le langage pour effectuer ce QUOI n'existe t-elle pas ? (dans ce cas je ne crée rien, j'utilise de l'existant)
- 3. Je donne quel nom EXPLICITE à ce traitement ?

Accessibilité Statisme Type Nom (Paramètres){Corps}

8. Le comportement que je veux effectuer est il un traitement propre à UN objet ou PLUSIEURS objets différents manipulés?

11. Ma méthode doit elle pouvoir

être appelée directement depuis l'extérieur, ou bien passer par un

contrôle, une protection préalable ?

Si UN SEUL objet -> c'est un comportement propre plus à la CLASSE qu'a des objets = faire du STATISME.

- 9. Ce comportement, est-il à répèter de manière régulière sur des objets différents?
- => Si réponse est NON => STATISME
- 10. Est-ce que je suis en train de vouloir en fait appliquer des traitements sur UN seul objet?
- => Si réponse est OUI => STATISME

- 5. Je retourne QUOI ? De quel NATURE ? (= j'obtiens un résultats via des traitements à travers les propriétés de mon objet ou Je modifie QUOI ? (= pas de retour mais une modification des attributs de l'objet recherché via des traitements)
- 6. Ce QUOI va retourner, va aller dans QUOI? ou modifier QUOI? Le contenu produit résultant de l'opération est-il compatible avec le conteneur dans lequel je veux le placer ?
- 7. Est-ce que je modifie uniquement l'objet que je manipule via un appel d'une méthode le composant ? Autrement dit, est-ce que je ne suis pas en train accidentellement de modifier d'autres objets à travers mes traitements ? (Effets de bord à éviter)

Algorithmique = écriture du BLOC d'instructions via des outils

OUTIL 1: STRUCTURES DE CONTROLE

Mon programme nécessite t'il des stockages de valeurs, des compteurs à l'intérieur du traitement sans réutilisation extérieure?

=> VARIABLES DE PORTEE LOCALE

Dois-je conditionner l'execution de certaines portions d'instruction selon la réalisation ou non d'une expression parenthésée (appelée condition)?

=> BRANCHEMENTS CONDITIONNELS

Dois-je forcer l'exécution perpetuelle d'une portion d'instructions tant que la valeur d'une condition ne correspond pas à ce que j'attends?

=> DO WHILE

Est-je besoin de faire des instructions un certain nombre de fois?

=> FOR, FOR(){FOR(){}}

Doit je passer un revue l'ensemble des éléments qui compose un objet, une liste(qui est un objet elle même)?

Que DOIT faire la fonction ? Je le fais COMMENT ?

Est-ce que j'ai besoin de variables à portée locale ? De quel TYPE?

Il y a t'il des modifications à apporter sur d'autres attributs de l'objet en dehors du RETOUR du traitement ? => ACCESSEURS ET MUTATEURS

Est-ce que je veux executer des instructions uniquement di la condition est remplie?

=> IF() CLASSIQUE

Est-ce que je veux executer un bloc d'instruction ATTACHE à une valeur qui doit être le résultat du contrôle de ma condition?

=>SWITCH CASE:;

Est-ce que je veux qu'un bloc par défaut soit exécuté si la condition n'est pas remplie, ou une autre condition? elseif() else()

=> FOREACH

Autres questionnements importants en orienté Objet

Accessibilité

Certaines propriétés de ma classe (attributs et fonctions) doivent elles être accédées directement depuis l'extérieur de ma classe à travers des manipulations ?

Certaines propriétés de ma classe doivent elles pouvoir être protégées de toute modification directe de l'extérieur 2

=> Cette protection se nomme ENCAPSULATION

Pour chaque attribut (variable) de mon objet, je veux les manipuler directement ou bien passer par des mécanismes de contrôle avant de les manipuler ?

=> ACCESSEURS ET MUTATEURS

(Manipuler = soit modifier la valeur de la variable via une affectation, soit récupérer la valeur de la variable)

This

Dans ma feuille de classe ou je construit mon objet, est ce que j'ai des traitements à effectuer qui portent directement sur des propriétés de ma classe dans laquelle je suis ? => précéder dans ce cas l'attribut ou la méthode de THIS

this-> ou this. se manipule directement dans une classe. this veut dire -> l'attribut, la méthode de cette classe ou je travailler, ou je suis.

Constructeur

Quand je crée un nouvel objet via une instanciation, est ce que je veux lui affecter des valeurs par défaut à ses attributs au moment de la création pour gagner du temps?

Les différents objet que je crée doivent il initialiser leurs attributs en nombre et en valeur de la même façon à leur création ?

Le constructeur est une fonction qui porte le même nom que sa classe, et n'a pas de RETOUR. En effet, la création d'un objet se fait au travers d'un new Classe();

=> Ce classe(), est en fait une méthode, l'instanciation passe par une fonction, il peut y avoir plusieurs constructeurs, avec ou sans paramètres (on peut écrire plusieurs constructeurs dans une classe).

Quand je crée un nouvel objet via une instanciation, est ce que je veux lui affecter des valeurs par défaut à ses attributs au moment de la création pour gagner du temps ?

Les différents objet que je crée doivent il initialiser leurs attributs en nombre et en valeur de la même façon à leur création ?

Le constructeur est une fonction qui porte le même nom que sa classe, et n'a pas de RETOUR. En effet, la création d'un objet se fait au travers d'un new Classe();

=> Ce classe(), est en fait une méthode, l'instanciation passe par une fonction, il peut y avoir plusieurs constructeurs, avec ou sans paramètres (on peut écrire plusieurs constructeurs dans une classe).