1.什么是集合？ / Qu’est-ce qu’une collection ?

集合是对象的容器，用来存放和管理一组元素。

Une collection est un conteneur d’objets, utilisé pour stocker et gérer un ensemble d’éléments.

2.集合有什么用途？ / À quoi servent les collections ?

集合可以用来存储、查找、修改和遍历数据，让我们一次性处理一组对象。

Les collections servent à stocker, rechercher, modifier et parcourir des données, ce qui permet de manipuler un ensemble d’objets en même temps.

3.Pharo 标准库提供了哪些集合类型？ / Quels types de collections la bibliothèque standard de Pharo propose-t-elle ?

Array / 数组

固定大小，适合已知长度的数据。

Tableau de taille fixe, adapté aux données de longueur connue.

OrderedCollection / 有序集合

大小可变，可以随时添加和删除。

Collection dynamique, on peut ajouter ou supprimer des éléments facilement.

Set / 集合

无序且不允许重复元素。

Ensemble non ordonné, sans doublons.

Dictionary / 字典

以键值对的形式存储数据。

Stockage sous forme de paires clé-valeur.

4.如何迭代集合？ / Comment itérer sur une collection ?

在 Pharo 中，可以通过发送不同的消息来迭代集合：

do: 遍历所有元素

collect: 映射生成新集合

select: 过滤满足条件的元素

inject:into: 累积计算

En Pharo, on parcourt une collection en lui envoyant différents messages :

do: pour parcourir chaque élément

collect: pour transformer et créer une nouvelle collection

select: pour filtrer selon une condition

inject:into: pour accumuler un résultat

#(1 2 3 4 5) select: [ :each | each even ].

"Résultat : #(2 4)"

5.这些集合之间有什么区别？ / Quelles sont les différences entre ces collections ?

Array / 数组

固定大小，初始化时长度就确定。

Tableau de taille fixe, la taille est définie à l’initialisation.

OrderedCollection / 有序集合

大小可变，可以随时添加和删除元素。

Collection dynamique, on peut ajouter ou supprimer des éléments librement.

Set / 集合

无序，不允许重复元素。

Ensemble non ordonné, sans doublons.

Dictionary / 字典

按照键值对存储和访问元素。

Stocke et accède aux éléments sous forme de paires clé-valeur.

6.我是如何找到这些信息的？ / Comment ai-je trouvé ces informations ?

我观看了老师提供的教材视频和 PPT；

在 Playground 里自己写了一些小例子进行测试；

又在网上查阅了相关的资料，做了补充理解。

en regardant les vidéos et les PPT fournis par le professeur ;

en testant moi-même de petits exemples dans le Playground ;

en cherchant des ressources supplémentaires sur Internet pour compléter ma compréhension.

"Array"

a := #(1 2 3 4).

Transcript show: (a at: 2). "→ 2"

"OrderedCollection"

oc := OrderedCollection new.

oc add: 10; add: 20.

Transcript show: oc size. "→ 2"

"Set"

s := Set new.

s add: 1; add: 1; add: 2.

Transcript show: s size. "→ 2"

"Dictionary"

d := Dictionary new.

d at: 'name' put: 'Pharo'.

Transcript show: (d at: 'name'). "→ Pharo"

"迭代 "

#(1 2 3 4 5) do: [ :each | Transcript show: each printString; cr ].

#(1 2 3 4 5) select: [ :each | each even ]. "→ #(2 4)"

#(1 2 3) collect: [ :each | each \* 2 ]. "→ #(2 4 6)"

#(1 2 3 4) inject: 0 into: [ :sum :each | sum + each ]. "→ 10"

==2==

1.如何在 Pharo 中写条件语句？ / Comment écrire des conditionnels en Pharo ?

Pharo 使用消息而不是关键字来写条件语句。例如：

En Pharo, on utilise l’envoi de messages plutôt que des mots-clés pour écrire des conditionnels.

x > 5

    ifTrue: [ Transcript show: '1'; cr ]

    ifFalse: [ Transcript show: '2'; cr ].

2.Pharo 的条件语句和其他语言有什么不同？ / Quelle est la différence avec d’autres langages ?

在 Java 或 Python 中，条件语句通常用关键字 if。

En Java ou Python, on utilise le mot-clé if pour exprimer une condition.

Pharo 则把条件判断看作对象的行为，通过 消息传递 来实现。

En Pharo, les conditions sont traitées comme des comportements d’objets et exprimées via l’envoi de messages.

3.这种写法的优缺点是什么？ / Quels sont les avantages et inconvénients de cette approche ?

优点 / Avantages :

保持“一切皆对象”，语法统一，没有特殊关键字。

更符合面向对象编程的思想，条件判断就是对象的行为。

条件语句可以作为对象传递，更灵活。

Avantages :

Tout est objet, la syntaxe reste uniforme, sans mot-clé spécial.

Plus cohérent avec l’esprit de la programmation orientée objet.

Les conditionnels peuvent être passés comme objets, ce qui donne plus de flexibilité.

缺点 / Inconvénients :

对习惯了传统 if 语法的人来说不直观。

初学者需要一些时间适应消息风格。

代码嵌套多时可读性下降。

Inconvénients :

Moins intuitif pour ceux qui sont habitués à la syntaxe classique if.

Les débutants doivent s’habituer au style basé sur l’envoi de messages.

Quand il y a beaucoup d’imbrications, la lisibilité du code peut diminuer.

4.我是如何找到这些信息的？ / Comment ai-je trouvé ces informations ?

看了老师给的教材视频和 PPT，对 Pharo 的条件语句有了总体认识。

J’ai regardé les vidéos et PPT fournis par l’enseignant pour une vue d’ensemble.

在 Help Browser 与 Pharo by Example 里查语法说明，并在 Playground 里自己跑小例子验证。

J’ai consulté le Help Browser et Pharo by Example, puis j’ai vérifié dans le Playground avec de petits exemples.

又在网上搜索社区讨论（Stack Overflow、mailing list 片段），对 ifTrue:ifFalse:、ifNil: 等常见用法做了对比。

J’ai complété par des recherches en ligne (discussions de la communauté) pour comparer ifTrue:ifFalse: et ifNil:.

5.我尝试过的代码（条件语句） / Exemples que j’ai testés (conditionnels)

"1) 基本分支 / Branche simple"

n := 7.

n > 5

    ifTrue: [ Transcript show: 'Bigger'; cr ]

    ifFalse: [ Transcript show: 'Smaller'; cr ].

"2) 只在真时执行 / Exécuter seulement si vrai"

(-3 > 0) ifTrue: [ Transcript show: 'Positive'; cr ].

"3) 作为表达式返回值 / Utiliser le conditionnel comme expression"

label := (n even

            ifTrue: [ 'even' ]

            ifFalse: [ 'odd' ]).

Transcript show: label; cr.

"4) 多层判断 / Imbrication"

m := 0.

m > 0

    ifTrue: [ Transcript show: 'Positive'; cr ]

    ifFalse: [

        m = 0

            ifTrue: [ Transcript show: 'Zero'; cr ]

            ifFalse: [ Transcript show: 'Negative'; cr ] ].

==3==

1.如何在 Pharo 中写一个带类和方法的小程序？

Comment écrire un petit programme avec classes et méthodes en Pharo ?

在 Pharo 中，类是对象，定义时需要指定实例变量。

En Pharo, une classe est aussi un objet ; on la définit avec des variables d’instance. Les méthodes définissent les comportements de cette classe.

一个简单的 Counter 类 / Exemple : une classe Counter simple

Object subclass: #Counter

    instanceVariableNames: 'count'

    classVariableNames: ''

    package: 'MyPackage'.

"初始化方法"

Counter >> initialize

    count := 0.

"增加计数"

Counter >> increment

    count := count + 1.

"获取当前值"

Counter >> value

    ^ count.

"重置计数"

Counter >> reset

    count := 0.

测试 / Test :

c := Counter new.

c increment.

c increment.

Transcript show: c value printString; cr. "→ 2"

c reset.

Transcript show: c value printString; cr. "→ 0"

2.我写了什么程序？遇到什么问题？

/ Quel programme ai-je écrit ? Quels problèmes ai-je rencontrés ?

我写了一个简单的 Counter 类，包含 increment、reset 和 value 方法。

J’ai écrit une petite classe Counter avec les méthodes increment, reset et value.

遇到的问题是：一开始忘了在 initialize 里给 count 赋初始值，导致 nil 出现错误。后来我在 initialize 中写了 count := 0. 解决了问题。

Le problème rencontré : j’avais oublié d’initialiser la variable count dans initialize, ce qui provoquait une erreur avec nil. J’ai corrigé en ajoutant count := 0. dans initialize.

3.我是如何找到这些信息的？

/ Comment ai-je trouvé ces informations ?

我参考了老师提供的 PPT 示例；

在 Playground 尝试创建自己的类并运行；

查阅了 Pharo by Example 和网上的一些教程。

Je me suis basé sur les exemples du PPT du professeur ;

J’ai testé dans le Playground en créant ma propre classe ;

J’ai consulté Pharo by Example et quelques tutoriels en ligne.

Ma ligne:

<https://github.com/jingyihuang123/c3p_my_count/tree/main/src>

==4==

1.Pharo 方法通常有哪些编码规则？ / Quelles sont les règles de style courantes en Pharo ?

方法应当 简短、可读，最好只做一件事。

方法名一般以小写字母开头，采用驼峰式，例如 incrementCounter。

不要写太长的方法链或过深嵌套，保持清晰。

Les méthodes doivent être courtes et lisibles, idéalement ne faire qu’une seule chose.

Les noms de méthodes commencent par une minuscule et utilisent le camelCase, par exemple incrementCounter.

Éviter les chaînes trop longues de messages et les imbrications complexes.

2.有哪些工具可以帮助发现违反规则的情况？

/ Existe-t-il des outils pour détecter les violations de style ?

Pharo 提供了 Critic Browser，可以自动检查常见的代码风格问题。

Pharo offre le Critic Browser, qui peut analyser automatiquement le code et signaler les violations de style.

3.展示违规代码示例

符合风格的代码：

Counter >> increment

    count := count + 1.

 不符合风格的代码（方法太长 + 名字不规范）：

Counter >> IncrementCounterAndPrintItNow

    count := count + 1.

    Transcript show: 'The value is now: ', count printString; cr.

    "方法名太长，不符合规范，le nom est trop long "

==5==

1.你能学到级联（cascades）和代码块闭包（block closures）吗？你是怎么学习的？

Peux-tu apprendre les cascades et les block closures ? Comment t’y es-tu pris ?

级联 / Cascades

在 Pharo 中，级联用分号 ;，可以对同一个对象连续发送多个消息，避免重复写对象名。

En Pharo, une cascade s’écrit avec ;, ce qui permet d’envoyer plusieurs messages de suite au même objet sans le répéter.

Transcript

    show: 'Hello';

    cr;

    show: 'World';

    cr.

2.代码块闭包 / Block closures

闭包用方括号 [] 表示，可以像函数一样传递。

Un block closure est écrit entre crochets [], il peut être passé comme une fonction.

square := [ :x | x \* x ].

Transcript show: (square value: 5) printString. "→ 25"

3.我是如何学习这些内容的？ / Comment ai-je appris ces notions ?

我主要通过：

阅读老师 PPT；

在 Playground 尝试自己写级联和闭包；

查阅 Pharo by Example 和网上的资料。

J’ai appris en :

lisant les slides du professeur ;

testant moi-même les cascades et block closures dans le Playground ;

consultant Pharo by Example et des ressources en ligne.

4.我问过老师的问题 / Question posée au professeur

我在练习过程中遇到一个问题，就去问了老师：

当我在 Pharo 里用 Ctrl+S 时，代码能正常保存，但用 Ctrl+D 时会出现错误。我不理解原因。

Pendant mes exercices, j’ai posé la question suivante au professeur :

Quand j’utilise Ctrl+S dans Pharo, le code se sauvegarde normalement, mais avec Ctrl+D une erreur apparaît. Je ne comprends pas pourquoi.