

Secret key : sk |  $\text{Sign}(\text{Message}, \text{sk}) = \text{Signature}$   
Public key : PK |  $\text{Verify}(\text{Message}, \text{Signature}, \text{PK}) = \text{T/F}$   
Ledger dollars : LD 账元  
比特币 : Currency = Transaction history

私钥加密 & 公钥验证  
唯一编号

Cryptographic hash functions 加密散列函数

↓  
 $\text{SHA256}(\text{"ABCD"}) = \text{固定长度的比特字符串}$

↑  
Message/file

↓  
«信息散列值» / «指纹»  
Hash / Digest

Block-reward  $\Rightarrow$  No sender/signature & Adds to total money supply

Une variable permet de stocker de l'information qui peut ensuite être manipulée par le programme.

- Exemple de modification : création, suppression, modification des données.

- Declaration : indique le nom et le type de la variable

- Initialisation : attribue une valeur à la variable

- Manipulation

- La déclaration et l'initialisation se font souvent ensemble

- Chaque variable a un scope.

- Il y a 8 types primitifs

- 5 pour les entiers : byte, short, int, long et char

- 2 pour les nombres à virgule flottante (réels) : float et double

- le type boolean pour les valeurs booléennes.

```
// On peut déclarer et initialiser une
// variable en une instruction
int unEntier = 42 ;
float pi = 3.14159f;
double e = 2.7183 ;
char unCaractere = 'a' ;
boolean valeurBooleenne = true ;
boolean autreValeurBooleenne = false ;
// On peut ensuite réassigner une valeur
// à une variable. Attention à respecter
// les types, unEntier doit rester un entier
unEntier = 0 ;
```

## switch

- Lorsque l'on connaît les valeurs possibles d'une variable, on peut utiliser la structure `switch`. Dans l'exemple, *langage* est un objet de type `String`.

```
1 switch(langage){
2   case "Chinese" :
3     return "  我爱熊猫! ";
4   case "Greek" :
5     return "μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην";
6   default :
7     return "Call me Ishmael. Some years ago";
8 }
```

14 / 34

## switch (cont'd)

- La structure `switch` ne peut s'utiliser qu'avec certains types de variables.
- Les types acceptés sont
  - `char`, `byte`, `short`, `int`
  - Leur version objet : `Character`, `Byte`, `Short`, `Integer` (on y reviendra)
  - Les chaînes de caractères `String`
  - Les énumérations, qui seront abordées plus tard
- Les mots clef `break` ou `return` sont **nécessaires**, il permet de sortir de la structure `switch`; autrement, on se retrouve dans le cas des `if` enchaînés.
- Le mot clef `default` permet de gérer le cas où la variable ne correspond à aucun des cas traités.

15 / 34

## Opérateur ternaire

- Quand on cherche simplement à parcourir les valeurs d'un tableau ou d'un objet, on utilise la boucle `for each`.
- Si l'on a besoin de l'indice d'un élément, par exemple pour modifier l'élément, on utilise la boucle `for`.

Il existe trois types de boucles itératives

- boucle `while`
- boucle `do`
- boucle `for` (cette dernière a deux versions)

**1. Quelles sont les différences entre les types primitifs short, int, et long ?**

short: entier signé de 16 bits;  
int: entier signé de 32 bits;  
long: entier signé de 64 bits.

**2. Quelle est la différence entre une boucle while et une boucle do while ?**

while: jugement avant la boucle, peut-être n'entrer pas dans la boucle.  
do while: la condition est jugée après l'entrée dans la boucle. La boucle fait au moins une fois.

**3. Qu'est-ce qu'une classe ? Qu'est-ce qu'un objet ?**

Classe: définir les caractéristiques et les comportements communs aux objets.  
Objet: avoir les valeurs de propriétés spécifiques et des méthodes exécutables.

**4. Quels sont les quatre types de visibilité ? Quelles sont les caractéristiques de chacun de ces types ?**

public: les membres publics sont accessibles.  
protected: accessible à lui-même, aux héritées, au même paquet.  
private: accessible uniquement par la class elle-même.  
default: accessible uniquement par le paquet lui-même.

**5. Que signifie le mot clé static lorsqu'il est mis devant un attribut ? Que signifie le mot clé final lorsqu'il est mis devant un attribut ? Que signifie la combinaison static final lorsqu'elle est mise devant un attribut ?**

static: la propriété est au niveau de la class.  
déclarer des propriétés statiques, accès direct par le nom de la classe  
final: la propriété ne peut pas être modifiée.  
static final: la propriété est. une constante au niveau de la classe qui ne peut pas être modifiée.

**6. Quels sont les types primitifs ?**

entiers : byte, short, int, long et char  
virgule flottante (réels) : float et double  
booléennes: boolean.

**7. Quelles sont les structures conditionnelles en Java ? 条件结构**

if; if-else; switch

**8. Quelles sont les structures itératives en Java ? 迭代结构**

while; do while; for

**9. Une fonction est-elle obligée d'afficher ou de retourner une valeur ? Peut-elle retourner plusieurs valeurs ?**

Non, peut n'y a aucune/ une seule/ plusieurs

**10. Quelle fonction permet d'instancier un objet ?**

new: `Personne personne = new Personne();`

**14.Qu'est-ce que le passage par valeur ? Quels sont les types concernés ? Quel est l'autre type de passage de parametre ?**

pass by value: int, float, boolean...

pass by reference: classe, tableau...(transmission d'une référence(adresse))

**16. Qu'est-ce qu'un constructeur ?**

initialiser des objets(new)

**17. Quest ce que l'heritage ?**

permettre à une classe d'hériter des propriétés et des méthodes d'une classe.

**19. Que signifie le mot clé super ?**

référencer et appeler des membres de la classe parente dans la sous class.(hériter)

**22. Qu est-ce que l'encapsulation ?封装**

encapsuler les données et les opérations dans une class, opérer via une interface publique.

**23.Qu'est-ce qu'une interface ? Pourquoi utiliser une interface plutôt qu'une classe ab-straite ?**

interface: définir la manière des communication entre les classes.

l'effet de l'héritage multiple, dissociant le code, partageant et réutilisant les méthodes.

多继承, 解耦合代码, 方法的共享和重用

**24. Qu'est-ce que le polymorphisme ?**

indique que l'objet peut prendre différentes formes et avoir des comportement différents.

**25. Qu'est-ce que la complexite algorithmique ? Donner des exemples de complexés classiques.**

le degré d'augmentation des ressources de calcul requises.(aspect et temporelle)

$O(1)$ ,  $O(n)$ ,  $O(n^2)$ ,  $O(\log n)$