package 实质就是文件夹

package my;

public class Hello ···

编译、执行 Hello、my.Hello 均不可

放在 my 文件中，返回上层，那么运行 my.Hello 可执行

不属于任何 package 的属于 default package

Java -cp ~/java my.Hello ⇐ 这样好，不改环境变量

Java accesss specifiers

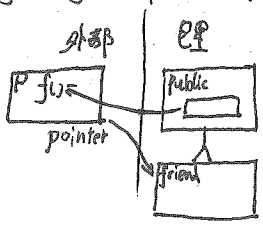
public "friend" (default package)

private

protected: "sort of friendly"

There can be only one public class per compilation unit

private/friendly 类所做的对象可被人所用



不能拿这块的名字在外用

Inheritance ~~syntax~~ syntax

class myClass extends {

Static binding: call the function as the code

Dynamic binding: call the function of the object

慎用 switch case

final: this cannot be changed 不能被 override

一个函数有 abstract 关键字那么不能有 body;

一使有一个 abstract 函数，那么它就是 abstart 类

一个 abstract 类不能拿来做对象

没有 abstract 函数的类也可是 abstract.

面向接口的编程模型

写代码先接口，再写类

内部类

编译时内部类，内部类一定会生成对应的 class 文件

Enums

enum Season {WINTER, SPRING, SUMMER, FALL}

WINTER 是 Season 类的对象，也是静态全局变量

JAVA 中的函数名可与一个变量名相同.

public static void main(String[] args)

C 中这个参数第0个是文件本身

Java 中第0个直接是手输入的参数

enum 对象，可直接拿名字输出字符串

toString() 会保证这个程序实现

{} 有可能指匿名的内部类.

static import

```
double r= Math.cos(Math.PI * theta);
import static java.lang.Math.PI;
import static java.lang.Math.*;
double r= cos(PI * theta);
```

C 现在可以进行动态数组.

实质是 malloc + free.

C++17 支持 for each 循环

Length ⇐ 成员变量

Java 的数组是对象 (必须被 new 出来，只能放在堆里) Java 的数组可以知道自己长度

Java 的数组变量是指针

new 出来必为0

```
String[] a= new String[20]; // 只是制造了20个null pointer.
for( int i: k) // i是一个独立变量，在循环地每一轮被k的每个元素赋值
{
}
for(int x= 0; x<k.length; x++){
    int i= k[x];
}
public class a
{
    int x;
}
```

deque / deck /

‖

dek

queue 发音自法语

k音 u音

Collection = container 的总和

1. Collection 所有类的对象都可以赋给一个 object 类的变量
   I. List ArrayList
   II. Set HashSet 此容器内放的都是不同的
2. Map HashMap 放两个 "dog"，后进去的会把替换前者

实现 Hashcode() 与 echo()

Java 旧版本容器是型不安全.

放入拿出都当成 object 类的对象.

Java 的容器放的是指针

Java vs. C++ What has been put into the collections

Thread

线程是 oop 的方式

线程将一个程序分开

thread 中 start() 函数即刻返回.

不会阻塞

Multi-Threads

It's not really simultaneously (for one core)

1. Create a class which implements interface

join() 向调度器发起信号

wait() 线程发起信号

stop()
suspend()
resume()
被弃用 线程可能会占用资源不释放，直接杀死会出问题

output 进数据

input 出数据

volatile 这变量会在线程之外被修改

Thread.yield(); 释放

如何不空等？不循环轮询

wait() 到时还给

对数据的访问是串行化的

All access to the data is synchronized

多个线程可共用同一个静态的代码块.

Private data. 公用数据不可能加锁

The key is in the data itself.

Nested synchronized is safe in Java

```
synchronized(a){
    synchronized(a){
        fu();
    }
}
```

Synchronized method

```
void f(){
    synchronized(){
    }
}
```

⇒ synchronized void f().

Unix 的读分阻塞与非阻塞.

类被装载，但其对象不见得会立刻初始化 (静止时候这个概念)

forname 会装载+初始化)