## Week 1

1.1 + 1.2

Class名字和文件名一样

每行后面;

public static void main(String[] args){ 一个class里先call main function

Welcome welcome = new Welcome(); 创建object

welcome.function();

}

Compile: javac Welcome.java

Run: java Welcome

1.3

类/object的属性：类里的instance variables（就是类里，方法外写的变量）

字符串双引号，char单引号

\n

Primitive types / class types class type大写，String，char，Integer，int

（可查primitive type都有哪些）

Type inference - var： var x = 100;

算数同C：15/8 = 1

Switch

Break, continue

// 注释

\*python不支持++

python，haskell True

C，java true

Array: 固定长度

int[] myArray = new int[6];

myArray.length 无括号

for (int n : anArray)

{

sum += n;

}

Arraylist:

import java.util.ArrayList;

ArrayList<String> a = new ArrayList<>();

.add, .get .size .isEmpty .remove （元素或者下标应该都行）

\*对于int这种primitive type，有对应的Integer object，java会自动转换

Hashmap： key，value

HashMap<String,Integer> map = new HashMap<>();

map.put("One",1);

int n = map.get("One");

public static void main(String[] args)

static declares the method does not need to be called for an object.

就是说没有object也可以叫这个function

## Week 2

Array初始值为0/ 0.0/ null

Array没有class

int[] array = {1,2,3,4,5}; or int[] array = new int[]{1,2,3,4,5};

int[][] twoD = new int[3][7]; \*可以有N dimension

twoD.length

twoD[row].length

for (int[] row : twoD)

int[] oned1 = new int[20];

int[][] twod = new int[10][];

twod[0] = oned1;

int[] n = twod[1];

2.3

Class and abstraction (separate the essential from the full detail)

class SimpleBook{

private String title;

private String author;

public SimpleBook(String title, String author) { constructor

this.title = title;

this.author = author;

}

public String toString()

{ return title + “by” + author; }

}

SimpleBook aBook = new SimpleBook(“A Title”, “An Author”);

UML - The Unified Modeling Language

## Week 3

Design class的一些注意事项：

Class名大写， 名词单数

Private instance variable

Public方法尽量少，能private就private

getXxxx 方法

return new ArrayList<String>(emailAddresses); 能改的东西包装一下，传回一个新list，防止别人改动class里private的东西

Class 也是 User Defined Type

Reference

null, 也是未初始化变量的初始值

Java Call by value, 不是call by reference

Return by value

Object lifetime 和variable lifetime不一样

Design a program

Constructor 和 class同名

no return type declaration.

## Week 4

UML图上的钻石符号：aggregation association （空心钻 / 实心钻 区别）

private ArrayList<LineItem> lineItems;

public Order() {

lineItems = new ArrayList<LineItem>();

}

或许，应该在input output的类里create object, 而不是在中间类create

Constant values: public static final int ADD\_CUSTOMER = 1;

• static denotes a class variable

• final denotes the variable cannot be changed by assignment

null can be compared using == and !=

Optional 类

Exception:

assert

throw new EmptyStackException();

try, catch, throw, throws, finally

继承：

extends

Super

## Week 5

A subclass inherits from a superclass.

The subclass gains all the properties of the superclass, can specialise it and can add more features.

一个父类可以有多个子类，但一个子类只能有一个父类

如果父类里的variable是private，那么子类无法获取

可以改成protected，这样子类可以获取

super()，等于call父类的constructor

且必须为constructor里的第一句

选择继承，super就一定要写，不然compiler自己补上的，格式不对就报错了

父类可以有多个constructor，super传的参数是哪个，就用对应的

父类可以把一些方法abstract，也可以写一些默认通用的方法

public abstract void draw(Graphics g);

父类里可以什么都不用定义，但保证每一个子类都有这样一个方法

不过有abstract方法的类没法创造实例object，且类也应该定义为abstract

public abstract class Shape （无实例object）

实例类继承abstract class之后，重写所有的abstract方法

An overriding method must be declared in a subclass, have the same name, parameters and return type.

有改写，优先call子类改写的版本，没有改写则跑父类的版本

Shape sh = new Square(10,10,40); 可以这么写

不过如果有的function是Square特有的，不能用

其他情况下，一切都按正常square object走

Dynamic binding

method body is determined at runtime by looking at the class of the object the method is called for at runtime

Static binding

method body to be executed is always uniquely determined

(The same can technically be done for instance methods if no overriding has taken place.)

继承的优点：

降重，其他code可以在不知道具体shape的情况下用shape的方法等

类型为Shape的arraylist可以放所有子类object

Static variables are class variables

每个类就一份，不像instance variable，每个object有自己的一份

## Week 6

All classes either directly or indirectly inherit from class Object.

第一个ppt里有Object类已经定义的一些methods

Java：strong type checking.

Object[] elements = new Object[n];

Array elements can reference any kind of object.

super.g() 叫父类的方法

Template method：

父类：

public void doSomething()

{

doThis();

... // Some interesting code

doThat();

}

A subclass might override the doThis and doThat methods but not doSomething

partly specialised by a subclass

Final class：没有子类

Final method:无法被override

Interface:（可inherit）

defining a type separately from classes

a collection of method signatures (return type, name, parameters)

Allow different objects to be used with the same code

separation of types from classes

public interface ShapeIF

{

void draw(Graphics g);

void move(int x, int y);

}

class can implement an interface （可implement several interfaces）

class MyShape implements ShapeIF

{

// Class must override draw and move methods

// or be abstract.

}

UML: Dashed line with open triangle denotes implements.

使用：

public void drawPicture(ArrayList<ShapeIF> shapes, Graphics g){

for (ShapeIF shape : shapes){

shape.draw(g);

}

}

Polymorphism： 一个方法，不同type可用

An Object has state. The values of its instance variables.

Type Conformance

public void sort(Comparable[] a) comparable是一个interface， 在第三个ppt

{

// Sorting algorithm

if (a[i].compareTo(a[i+1]) < 0) { ... }

// ...

}

sort any array of objects that conform to Comparable

上层client class不需要知道具体的implementation class，通过interface和factory即可

下层code改了或者加了，上层不用动

Generic Interface

• public interface List<E> // A generic interface{

boolean add(E obj);

E get(int index);

boolean isEmpty();

// etc...

}

E is a type variable, instantiated during type checking

由输入的parameter决定

generic class 见ppt

Generic Methods：

parametric polymorphism， use/return values of different types.

An alternative to overloading.

A way of avoiding duplication of code

public <T extends Comparable> T max(T t1, T t2) {

if (t1.compareTo(t2) > 0)

{ return t1; }

else

{ return t2; }

}

## Week 7

A module is a collection of packages and a module descriptor.

A package is a collection of classes, interfaces, enums, records that provides a named scope.

high-level structure，manage complexity in large programs，Encapsulation/information hiding，enable useful implementation techniques

package mypackage; declaration， 放在程序文件第一行

一个class只能在一个package里

Package name应该都是小写

A package contains classes/interfaces, but not packages – only the names are hierarchical.

.class files must be stored in a directory structure matching the package hierarchy

two parallel sets of directories：

* src to hold the source files
* build to hold the .class files

Import: package declaration之后，class declaration 之前

import datastructures.matrix.normal.\* \*用来import package里的所有class

你要import的东西需要存在 (at runtime)

Classes from java.lang are imported automatically. 其他都需要写import

如果不想import，可以把class的全名写出来（也可避免两个class重名）

CLASSPATH – a list of directories where .class files in their packages are stored

If classes from multiple packages are used then pathnames for all must be supplied

工具：Maven

jar = Java Archive （类似zip）

The .class files for package(s), library or program can be collected together into a single .jar file

内部directory structure is preserved.

可以是executable jar file

A package defines a scope

Public class可以从package外获取，otherwise，non-public class

Classes cannot be declared private or protected.

Methods and variables declared protected can be accessed by from any subclass AND any class in the same package.

This allows a class to declare a protected interface to other classes in the same package.

UML Package Diagram