AP CS (Java)

Env

Oracle JDK, IntelliJ IDEA

Book

AP Computer Science A

Core Java

Comments

```
/*
*/

//
```

Input

Scanner (read int, read string, read a line)

Output

System.out.println()

format print:

```
double d=123.456;
System.out.printf("%.2f\n",d);
System.out.printf("%10.2f",d);
//
int i=123;
System.out.printf("%10d",i);
System.out.printf("%010d",i);
```

If

```
if (condition) {
} else if (condition) {
} else {
}
```

Loop

for loop
for each loop
while loop
do while loop
break, continue
Iterator loop: Remove by iterator to avoid ConCurrentModificationException

Debug:

set breakpoint,
start debugger
stepover
stepinto, stepout

primitive:

byte, short, char, **int**, long, float, **double**, **boolean**

Wrapper:

int vs Integer (boxing, unboxing)

String: (immutable)

String s="abc";
== vs equals

Special chars: \n, \b, \t

"hello \n world"

Escape chars:

"hello \\ world"

ArrayList:

```
// create
ArrayList<Integer> l=new ArrayList();

List<Integer> a=List.of(1,2,3);
ArrayList<Integer> l=new ArrayList(a);
// read
l.get(0)
// update
l.add(100)
l.set(1,200)
// delete
l.remove(1)
l.remove(Integer.valueOf(1))
// query
l.indexOf(100)
l.contains(100)
l.size()
// sort
Collections.sort(l)
// reverse
Collections.reverse(l)
```

Convert type:

char <-> int

```
(int)'a'
(char)97
```

char <-> string

```
""+'A'
char c=s.charAt(index)
```

int <-> string

```
""+100
or
String.valueOf(…)

Integer.parseInt(s)
```

Integer <-> string

```
String d="100"
Integer
i=Integer.valueOf(d)
Integer.valueOf("f",16)==15

d=i.toString()
```

String apis:

```
"abc".charAt(1)
"abcdefg".substring(2,4)
"abcdefg".substring(2)
int length()
s.contains("a")
s.indexOf("a")
s.replace("a","A")
s.replace("a","")
```

Class

full class name =
package name + class name

Compile and run:

```
javac -d . TestClassName.java
java c3/TestClassName
```

package:

1 Create a package, and a class 'A' in it
2 in different package, a class 'B' use 'A'

Javadoc

Write Javadoc: @author, @param, @return, @throws
Generate htmls from Javadoc

Storage number

Decimal, Binary, Hex, Octal table

Convert:

int -> binary:
Integer.toBinaryString(i)

binary -> int
Integer.parseInt(binary, 2)

binary literal: 0b0111

int -> binary (manually)
binary -> int (manually)

int -> hex
Integer.toHexString(i)
hex str -> int
Integer.parseInt(hexStr, 16)

int -> binary -> hex (manually)
hex -> int (manually)

Overflow:
byte b=127;
b++;
System.out.println(b)
// why -128?

Two's Complement
a negative binary -> negative int
a negative int -> negative binary

Float number accuracy
double a=0.1
double b=0.2
double c=0.3
a+b==c ?

How to compare 2 floating number:
(Math.abs(d1 - d2) < 1e-8)

println(Math.sqrt(-1)); // NaN
println(0.0/0.0); // NaN
println(100/0.0); // Infinity
println(-100/0.0); //-Infinity
println(100/0); // exception

Float number accuracy
Math.round(d*100.0)/100.0;
System.out.format("%.2f\n",d);

Operators

```
2/4 -> 0
2.0/4 -> 0.5
remainder: % (not modulus)
Logical: !,&&,||
increment/decrement:++,—
```

Float number accuracy
*, /, % (at same level)
&& > ||

Final variable

```
Final instance vals
Final class
```

Static

```
static property
static methods
Static method vs instance
method
```

Exception

hierarchy:

```
    Throwable
      /    \
Error    Exception
            /
       RuntimeException
```

Example:

```
if invalid parameter, throw
IllegalArgumentException

if not handle, what happen?
(Optional) how to handle
exception?
```

AP only test:

```
ArithmeticException
NullPointerException
ArrayIndexOutOfBoundsException
IndexOutOfBoundsException
IllegalArgumentException
```

Object

```
What is object?
State (instance variable)
Behavior (methods)

constructor (default, non-
default)
```

Encapsulation = information hiding +
getter/setter
accessor/mutator

Init
auto init instance vars
not auto init local vars

Access control
public, private

Java.lang.Object
Boolean equals(Object obj)
String toString()
Override equals, and toString

Methods

Overloading:
Method signature: name, a list of parameter types

Overriding

“This” keyword

```
‘this’ in constructor
‘this’ in methods
static methods have no ‘this’
```

pass by value

Java always pass by value (no matter the val type is primitive or reference)

Reference value

Primitive var vs Reference var

Reference type includes:
reference value
object value

Inheritance

Superclass, Subclass

IS-A, Has-A relation (Dog has a Tail, Dog is a animal)

Super keyword
super.member
super(...)

Polymorphism

What?
Why?

Latebinding

Casting:

- downcasting
- upcasting
- ClassCastingException

Interface

Only learn ‘Comparable’ interface, how to implement compareTo

Recursive

Form recursive pattern
Recursive case
Base case

Algorithms

insertion sort
selection sort
merge sort
(optional) quick sort

binary search

Math

Math.abs, Math.pow, Math.sqrt, Math.max, Math.min

2 ways:
Math.random()*(high-low)+low
Random class

Array

declare:
int[] oneD; int[][] twoD;

construct:
oneD=new int[3];
twoD=new int[3][]

init:
oneD[0]=100
twoD[0]=new int[4]

Literal:
int[][] i={{1,2},{3,4,5}}

Array assignment, such as char[] to int[], Dog[] to Animal[]

How arrays stored? Jagged array

Loop array (for, for each)

Print array: Arrays.toString(arr)

sort:
Arrays.sort(arr [,comparator])
sort by natural order
sort by comparable

program design and analysis

Assert:
assert condition:message

enable/disable assert: -ea, -da, -ea:p, -da:p

Software lifecycle:
Agile manifesto

UML
different diagrams
a tool: umlet

3 type of errors:

```
1 compile error
2 runtime error
3 logic error
```

CCC Java

Debug

1.sysout
2.breakpoint
3.try/catch
4.assert

Data type

Numeric literal with underscore:
long l=1234_5678_9012_3456L

flow control

switch
Ternary: (a>b)? a:b;

method

varargs

String

Query:
str.startsWith("a")
str.endsWith("c")

Count:
"abab".chars().filter(ch -> ch == 'a').count();

Is String a digit?

```
try {
    Double.parseDouble("+100.08");
} catch (NumberFormatException e) {
    b=false
}
b=true
```

Is Char a digit?
Character.isDigit('1')

Convert case:
"ABC".toLowerCase()
"abc".toUpperCase()

Join:
String.join(",", cities);

StringBuilder

Collections

implements Comparator
Search:
Collections.sort(list, comparator)
Collections.binarySearch(list, obj, comparator)

Override hashcode, equals

set:
Hashset:
add, contains, containsAll, remove, isEmpty, clear

addAll (union)
retainAll (intersection)
removeAll (difference)

map:
HashMap:
put, get

TreeMap:
SortedMap m=tm.headMap("key")
m.size(), m.lastKey()
m=tm.tailMap("e")
m.firstKey()

tm.lowerKey("e")
tm.floorKey("e")
tm.higherKey("e")
tm.ceilingKey("e")

tm.pollFirstEntry()
tm.pollLstEntry()

NavigableMap
nm=tm.descendingMap()

Operators

bitwise: &, |, ^, ~, <<, >>
>>> (unsigned right shift)
instanceof

Exception

```
method throws exception
void method() throws Exception {
throw new Exception("msg")
}

handle exception
try {
…
} catch (Exception ex) {
…
} finally {
…
}

"checked exception" vs "unchecked (runtime) exception"

Exception Propagation (see stack trace)
```

Collection

Init a set:
Set<Integer> s = Set.of(1,2,3);
HashSet<Integer> ss=new HashSet<>(s);

Init a map:
Map<Integer,String> m = Map.of(1, "A", 2, "B", 3, "C");
HashMap<Integer, String> mm=new HashMap<>(m);

Sort:
Collections.sort(names, (a, b) -> b.compareTo(a));

Created by TeensProgramming.com

Other

Memory

make suggestions to GC
islands of isolation

System.gc()

protected void finalize() throws Throwable

Java memory model

File IO

f=new File("file_name")
f.exists()
aBoolean=f.createNewFile()
f.isFile()
f.getAbsolutePath()
f.renameTo(new File("new.txt"))

d=new File("dir name")
d.mkdir()
fInDir=new File(f, "a.txt")
fInDir.createNewFile()
d.list()
fInDir.delete()
d.delete()

f=new File("file name")
fw=new FileWriter(f)
fw.write("hello\n");
fw.flush()
fw.close()

fr=new FileReader(f)
num=fr.read(charArr)
integer=fr.read()

High level api:
bw=new BufferedWriter(fw)
bw.write("a")
bw.newLine()
bw.flush()
bw.close()

br=new BufferedReader(fr)
br.readLine()
br.read()
br.close()

pw=new PrintWriter(bw)
pw.println(…)