

- 객체의 형변환

기본형 / 참조형 변수 형변환

(데이터타입)

(int)

(A)

(String)

1> 상속관계

2> 자동형변환

상위클래스명 변수명=new 하위생성자()

3> 변수명. 상위클래스멤버변수나메소드접근

단, 오버라이딩메소드는 하위클래스포함

4> 명시적형변환

상위클래스명 변수명=new 하위생성자()

하위클래스명 변수명2 = (하위클래스명)변수명

5> 변수명2. 상하위클래스멤버변수나메소드접근

```
class A{
```

```
class B extends A{
```

```
B b1 = new B();//형변환필요x
```

```
A a1 = new B();//자동형변환(서로 다른 타입 객체 통일)
```

```
==>a1.(B클래스포함)변수사용불가능
```

```
A a2 = b1;//자동형변환
```

```
B b2 = (B)a2;명시적형변환
```

```
((B)a1).(B클래스포함)변수사용가능
```

```
test(A a){....}
```

```
test(new B())
```

```
test(new A())
```

```
test(A 하위객체)
```

1> 여러 객체 저장 배열

2> 매개변수에 전달 객체 유형 다양

3> 멤버변수 대입 객체 유형 다양

- 인터페이스 – 추상 메소드만 정의/상수/공유 다중상속 가능 참조형변수 일종
객체 생성 불가능/생성자 없다(기본생성자 없다)

<pre>interface I1{ /*public final static*/ int x = 10; /*abstract public */void m1(); }</pre>	<pre>class A implements I1, I2{ int x = 30; int y = I1.x; public void m1(){ s.o.p(x);->30 s.o.p(I1.x);->10 s.o.p(I2.x);->20 } public void m2(){...} }</pre>
<pre>interface I2{ /*public final static*/ int x = 20; /*abstract public */void m2(); }</pre>	

자바 상속 – 단일상속 / 다중상속 (오버라이딩, 형변환.....)

9장 중첩 클래스와 중첩 인터페이스

중첩 = 내부 클래스==>변수처럼 사용(멤버변수(클래스의 모든 메소드) / 지역변수(메소드만))

1> non-static 멤버변수처럼 사용 중첩 클래스(인스턴스=객체)

2> static 멤버변수처럼 사용 중첩 클래스(정적)

3> 지역변수처럼 사용 중첩 클래스

<pre>class A{ B ab = new B(); class B{ 생성자 멤버변수 메소드 정의} static class C { 생성자 멤버변수 메소드 정의 } }</pre>
<pre>A.B ab = new A().new B(); A.C ac = new A.C();</pre>
<pre>class C{ 상속 오버라이딩 interface I {}</pre>

}
용도 - 다른 클래스에서는 해당 중첩 클래스나 중첩 인터페이스 사용 금지

10장 예외처리

오류 -> 프로그램 작성 - 컴파일 - 실행 도중 발생 오동작

컴파일오류 - 자바 문법=구문=syntax 어긋난 경우

==> 자바소스 원천적 수정하지 않으면 계속 발생

int i = 3.14;

1> double i = 3.14; 2> int i = (int)3.14;

실행오류

실행 조건에 따라 발생 사건- 발생 / 발생x

int i = Integer.parseInt(args[0]);

int j = 100 / i;

명령행매개변수 입력x -> 프로그래머 해결 가능 오류

명령행매개변수 입력값 "자바"

명령행매개변수 입력값 "0"

명령행매개변수 입력값 "100"

전원 off-> 실행 오류(프로그램 해결 x) --> 전원 공급

메모리 부족 -> 자바 메모리 정리 작업 개입x

컴파일	실행오류
문법 어긋난 경우	시스템오류 프로그램수정오류==>예외=exception

예외개념 프로그램 실행 도중 발생 오류로 프로그래머 제어 오류

예외상황별 자바 객체 취급 = ◦예외클래스

args[0] --> java.lang.ArrayIndexOutOfBoundsException 발생

"자바"-->정수x --> java.lang.NumberFormatException

100 / 0 -->java.lang.ArithmeticException

xxxxxxException

=====

예외처리

try – catch – finally- throws – throw

try{

예외발생가능성문장;

예외o- 중단하고 catch 이동

예외x-try 정상 수행

}

catch(예외타입클래스명1 변수명){

예외 발생되지 않도록 처리 문장

}

catch(예외타입클래스명2 변수명){

예외 발생되지 않도록 처리 문장

}

classa Exception { 변수 메소드 }

class ArithmeticException extends Exception{ + }

class ArrayIndexOutOfBoundsException extends Exception{ + }

class NumberFormatException extends Exception{ + }

Exception e = new xxxxxxException();

catch(Exception e)

1>직접처리 – 예외발생 메소드 내부 처리 문장 포함

```
try{
1; 2;
}
catch(AException e){ 3; }
catch(BException e){ 4; }
5;
```

-1번문장 예외발생x : 1 – 2 - 5

-1번문장 AException 예외발생o : 3 - 5

-1번문장 BException 예외발생o : 4 - 5

-1번문장 CException 예외발생o(=예외처리블록없는경우) : X

```
try{1; }
catch(AException e){ 3; }
catch(BException e){ 4; }
finally{ 2; } --> 예외발생유무, 예외타입, 처리유무 무관하게 항상 실행 블록
5;
```

-1번문장 예외발생x : 1 – 2 - 5

-1번문장 AException 예외발생o : 3 – 2- 5

-1번문장 BException 예외발생o : 4 – 2- 5

-1번문장 CException 예외발생o(=예외처리블록없는경우) : 2

- 반드시 항상 수행 경우

<p>- 자바 a.txt 파일 입력 a.txt 파일 연다(os 승인) a.txt 파일 처음-끝 읽어온다 a.txt 파일 닫는다 (os 사용파일끝 알려줌- 반드시 수행 닫지 않으면 os 대기중- a.txt 파일 사용 대기 다 른 프로그램)</p>	<pre>try{ FileReader fr = new FileReader("a.txt"); while(true){ if(fr.read() == -1) {break;} } fr.close(); }catch(IOException e){.....} finally{ fr.close(); }</pre>
<p>- db 이용 프로그램 작성 – jdbc</p>	<p>파일이나, db – 다른 프로그램 같이 사용 공유자 원들</p>

try-catch-finally

직접처리 – 예외 발생 메소드 내부에서 처리

2> 간접처리- 예외 발생 메소드를 호출하는 다른 메소드 내부에서 처리

throws

class Exception extends Object{	
class RuntimeException extends Exception { class FileNotFoundException extends Exception}' ==> 자바 throws, try-catch 반드시 처리 블록 정의 의무 예외 종류들-컴파일러 체크	class ArithmeticException extends RuntimeException{ ==> 자바 throws, try-catch 처리 블록 정의 선택 예외 종류들

try-catch-finally- 구현부

throws – 선언부 throws 예외클래스명

throw – 구현부. 조건문. 예외 의도적 발생 throw new xxxException()

100 / 0; --> ArithmeticException 발생 상황 자바 내부 정의

100 / -1 --> 의도적 ArithmeticException 발생

```
int i = -1;
```

```
if(i == -1){
```

```
    throw new ArithmeticException();
```

```
}
```

```
else{
```

```
    s.o.p(100 / i);
```

9, 10장--> 문법 10장

11장 – jdk 라이브러리

파일 입력 라이브러리

db 연결 라이브러리

jdk 패키지갯수 180여개

11장 java.lang/ java.util 기본 클래스들

- java.lang.Object 클래스

1> 다른 클래스 자동 상속

2> extends Object 자동 선언

3> Object 클래스의 모든 메소드들은 다른 클래스에 자동 포함-10

4> 상속 그대로 사용

5> 오버라이딩 사용

6> equals() / toString()

7> jdk 클래스들은 패키지화

import 패키지명.클래스명;

java.lang 패키지의 자바 언어 기본 빈번 사용 클래스 모음-"자동" import

java.lang.Object

```
import java.lang.Object;
```

```
class A extends Object
```

- toString

1> 객체참조변수 출력하면 자동 toString 메소드 호출하여 결과 출력

2> [패키지명.클래스명](#)@16진수객체주소 - Object 정의 그대로

3> 모든 클래스 객체 내용 출력 변경 -

- 기본형변수 값 동등성 비교

1>== 연산자 이용

- 객체 동등성 비교

1> == 연산자 이용-주소값 비교=false

2> equals 메소드 이용(상속 그대로= 타입 동일하고 주소 동일=false)

(String 클래스 equals 메소드는 타입 동일하고 문자열 내용 동일

Object 하위클래스 상속 오버라이딩)

12장 java.lang 멀티스레드구현 클래스들

15장 컬렉션 프레임워크 클래스 = java.util 유사 배열 클래스들

13, 14장 – (jdk 1.5, 1.8 추가 문법) – 제네릭 , 람다식

16,17, 19장 – 제외

18장 java.io. 파일입력클래스들

java.net 네트워크클래스들(기본)