**- 메소드 overloading**

**1개 클래스 같은 이름의 메소드 여러개 정의**

**매개변수 다르게 정의**

**리턴타입, modifier 제한없다**

**==> 매개변수 유형에 따라서 세분화**

**문자열--> 정수변경 기능 "동일"--> 10, 2, 8, 16**

**parseInt("10")**

**parseInt("10", 8)**

**- 생성자**

**객체 특성 -멤버변수 메소드**

**생성자--> 클래스 정의 --???-> 객체들 생성**

**1> 클래스이름 ( [매개변수] ){ new 객체생성시 실행문장}**

**2> 생성자 항상 사용자 정의할 필요는 없다**

**3> 기본 생성자 자동 정의**

**클래스이름 ( ){ }**

**4> 사용자 생성자 정의하면 기본 생성자 자동 삭제**

**5>생성자 overloading**

**6>**

**this.멤버변수명 --> 멤버변수와 매개변수(혹 지역변수) 이름이 같은 경우 구분**

**7> (생성자 오버로딩시) 다른 생성자 호출**

**this([변수값 전달])**

**private < (default)=no modifier < protected < public**

**- 상속**

**1개 클래스 {....}**

**이미 정의 클래스 "모든" 내용 필요 – IS A**

**class A{ i ma() {....} }**

**class B extends A{ i ma(){ 재정의 } }**

**-private 멤버 제외**

**- super.i / i**

**- 메소드 overriding**

**1> 상위 상속 메소드 하위클래스 그대로 사용**

**2> 하위클래스 재정의 수정 사용 가능**

**3> modifier 같거나 더 넓어야 한다.**

**4> 리턴타입 같다**

**5> 매개변수 같다**

**-다형성**

**메소드 overloading / 메소드 overriding**

**- 상위 생성자 자동 호출**

**class A 자동 extends Object{**

**--A(){super();} 자동정의 기본 생성자**

**A(int i){ i = 100;}**

**}**

**class B extends A{**

**B(){ super(100); }**

**}**

**B b1 = new B();**

**===> Object + A + B**

**- final / static / abstract**

**super. 상위변수명**

**super.상위메소드**

**super()**

**super(매개변수값 전달)**

**- 객체의 형변환**

**1> 상속관계**

**2> 자동형변환**

**상위클래스명 변수1 = 하위객체()**

**3> 변수1.멤버변수명===> 상위클래스**

**변수1.메소드명()===> 상위클래스**

**변수1.overriding메소드명()===> 하위클래스**

**4> 명시적형변환**

**상위클래스명 변수1 = 하위객체()**

**하위클래스명 변수2 = (하위클래스명)변수1**

**변수2.멤버변수명===> 하위클래스**

**변수2.메소드명()===> 하위클래스**

**5> 자동형변환?**

**"객체타입 통일"**

**1> 배열**

**2> 메소드 매개변수 다양한**

**3> 멤버변수 타입**

**static – 1 개 공유**

**클래스명. static변수**

**class A{**

**int i = 10;**

**static int j = 20;**

**}**

**A a1 = new A(); a1.i 객체참조변수명.non-static변수**

**A a2= new A(); a2.i**

**A.i===>오류**

**A.j ===> ok (a1.j a2.j 가능)**

**final – 상속 메소드 오버라이딩금지**

**객체생성, 호출**

**abstract – 객체생성 금지**

**상속 메소드 오버라이딩 의무화**

**final abstract cl;ass ---> error**

**- 인터페이스**

**교재 344p**

**1> 자바 클래스 단일상속만 가능**

**0> 다중상속 가능**

**1> public abstract 자동 선언 메소드만 가진다.**

**2> 인터페이스 타입 객체 생성 불가능하다.**

**3> 생성자 없다**

**4> public final static 자동 선언 변수만 가진다.**

|  |  |
| --- | --- |
| **interface 학생{**  **int 학번**  **[public abstract] void 공부하다();**  **void 점심먹다();**  **}** | **interface 교수{**  **가르치다();**  **점심먹다();**  **}** |
| **class 조교 implements 학생, 교수{**  **공부하다(){}**  **가르치다(){}**  **점심먹다(){}==> 상속 "모든" 메소드 오버라이딩 의무화**  **+멤버변수 메소드 생성자 추가 정의**  **}** | |
| **조교 s = new 조교();**  **s.공부하다();**  **s.가르치다();**  **s.점심먹다();** | **학생 s = new 학생();-->x**  **학생 s = new 조교();**  **--> 자동형변환**  **s.변수명-> 학생**  **학생.변수명-> 학생**  **s.overriding메소드()->조교**  **s.공부하다();->o**  **s.가르치다();->x**  **s.점심먹다();->o**  **교수 s = new 조교();**  **--> 자동형변환**  **s.공부하다();->x**  **s.가르치다();->o**  **s.점심먹다();->o** |

**class A{}**

**class B{}**

**class C extends A{}**

**interface D { m1() ; } 추가. B, C 상속하고싶다**

**class B implments D{ m1() {} }**

**class C extends A implements D{ m1() {} }**

**abstract class F{**

**abstract void ma();**

**void mb(){ }**

**}**

**-" 선택과 집중 "**

**-static메소드 / default메소드**

**10장 예외처리**

**자바 프로그램 작성 – 컴파일 – 실행**

**오동작 발생 = 오류 발생한다**

**컴파일오류- 자바 syntax 오류**

**오류발생표시부분 찾아서 소스 수정**

**(계속 발생)**

**int i = 3.14;**

**int i = (int) 3.14; double d=3.14;**

**실행오류- 실행 조건 발생 오동작**

**1> 자바소스 외적 – 자바소ㅡ스 처리x**

**전원off, 메모리부족, jvm오류**

**2>exception = 예외 = 자바소스 처리o**

**public static vid main(String [] args)**

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

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

**- 자바 객체지향언어**

**예외 발생 상황별 – 클래스로 정의**

**xxxException class들**

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

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

**args[0] ==> '0'--> ArithmeticException 발생**

**-int i = Integer.parseInt(args[0]) "-10"**

**int ar = new int[10];**

**ar[i] = i; => ArrayIndexOutOfBoundsException**

**class A{}**

**class B extends A{}**

**class C extends A{}**

**A a1 = new B();**

**C b1 = (C)a1;---> ClassCastException**

**String s = null;**

**s.toUpperCase()==>NullPointerException**

**IOException**

**Socket/....**

**args[0] ==> '0'--> ArithmeticException 발생**

**try{**

**1;**

**} catch(AException e){**

**3;**

**} catch(BException e){**

**4;**

**}finally{**

**2; => 반드시 수행**

**( 반복문 break;, 메소드 return;**

**}**

**5;**

**1 번문장 예외x – 1 – 2 - 5**

**1 번문장 AException예외o – 3-2-5**

**1 번문장 BException예외o – 4-2-5**

**1 번문장 CException예외o=예외 catch 못할 때 – 2**

**try-catch-finally- 예외 발생 문장을 포함한 메소드 내부 처리**

**throws- 예외 발생 문장을 포함한 메소드를 호출하는 다른 메소드 내부 처리**

**void ma() throws ArithmeticException { }**

**void ma(){ }**

**===> throws RuntimeException종류 생략 가능**

**class xxxxException extends Exception{}**

**===>컴파일러 체크한다. 예외처리 필수**

**checked exception**

**class xxxxException extends Runtimexception{}**

**===>실행 워낙 자주 발생, 컴파일러 체크하지 않는다.**

**예외처리하지 않아도 된다.**

**unchecked exception**

**class RuntimeException extends Exception{}**

**try-catch-finally- 예외처리**

**throws-예외 다른 메소드 처리 위임**

**throw – 예외 ( 인위적 강제적 의도적 ) 발생**

**100 / 0**

**Class.forName("A")**

**-교재**

**11장 기본 api 클래스**

**api 클래스 기능**

**Math.random()**

**Integer.parseInt()**

**+ 20**

**자바교안\_기본클래스. pdf**

**자바 라이브러리**

**패키지 >클래스 = 멤버변수+메소드(1개기능)**

**jdk 8 - 170여개 패키지**

**기본 자바 프로그램 자주 사용 기본 타입 클래스**

**- java.lang 패키지 자동 import**

**- Object**

**1> 자바 클래스들의 최상위 클래스**

**2> 클래스 선언부 extends Object 자동 정의**

**3> import 안해도 된다**

**4> 메소드들은 자바 모든 클래스 상속 자동 포함**

**5> toString()**

**6> equals()**

**================================**

**- String**

**- Wrapper**

**- Math, Integer, Calendar**

**-------------------------------------------**

**월- 12장 멀티스레드**

**15장 컬렉션 프레임워크**

**화, 수 – git 특강**

**목 13장 제네릭**

**금 14장 람다 - 중첩클래스와 중첩인터페이스**

**월 18장 io과 네트워크**

**화**