

4000번의 도전
드디어 찾아낸 반응 단백질



2012 인텔 국제과학경진대회 고든무어상 수상
(Intel ISEF 2012, Gordon E. Moore Award)



AN ANTONIO SPURS
처음에는 막막한 느낌이지만 시간이 지나면 감이 오거든요.



처음 배울때 감이 안나는게 뭐가 있을까요?



프로그래밍은 천재들만 할 수 있는것이 아니에요. 의지력이 필요할 뿐입니다.



Java 프로그래밍 수업내용 정리 및 실습 과제

책을 보거나 인터넷을 찾아보면서 배우고 거기에 저만의 방식을 추가했어요.



JACK ANDRAKA
IS CREATING A REVOLUTION



저는 여러분에게 컴퓨터를 통해 어떻게
세상을 바꿀 수 있는진 얘기하고 싶습니다

그러나 사실, 구글이나 위키피디아 등을 이용해
알고 싶은 모든 것을 찾을 수 있고

GOOGLE, WIKIPEDIA
AND STACKOVERFLOW

2020.04.23.목

B반 송명훈



15세 소년의 유일한 발명 도구는 “더 좋은 진단 키트를 만들고 싶었다.”
인터넷(internet)
- 잭 안드라카

목 차

1. 과제 리뷰 및 오답 정리
2. 수업 내용 정리
3. 실습화면 캡처
4. 연습문제 13-1
5. 연습문제 13-2
6. 연습문제 13-3
7. 문자열 조작] 연습문제 1-1
8. 문자열 조작] 연습문제 1-2
9. 문자열 조작] 연습문제 1-3



1. 과제 리뷰 및 오답 정리

1. 과제 리뷰 및 오답 정리

Assat class 수정

```
3 public abstract class Asset {
4     private String name;
5     private int price;
6
7     public Asset(String name, int Price) {
8         this.name = name;
9         this.price = price;
10    }
11
12    public String getName() {
13        return name;
14    }
15
16    public int getPrice() {
17        return price;
18    }
19    public abstract void printScript();
20 }
```

오타 수정

```
3 public abstract class Asset {
4     private String name;
5     private int price;
6
7     public Asset(String name, int price) {
8         this.name = name;
9         this.price = price;
10    }
11
12    public String getName() { return name; }
13
14    public int getPrice() { return price; }
15
16    public void setName(String name) { this.name = name; }
17
18    public void setPrice() { this.price = price; }
19
20    public void setPrice(int price) { this.price = price; }
21
22    public abstract void printScript();
23 }
```

getter와 setter는 선언해두는게 정석!

setter 추가!!



2. 수업 내용 정리

2. 수업 내용 정리 [Polymorphism]

다형성 (Polymorphism) : 하나의 객체가 여러 개의 자료형 타입을 가질 수 있는 것 [\[1\]](#)

1. 다형성이란 하나의 메서드나 클래스가 있을 때 이것들이 다양한 방법으로 동작하는 것을 의미. [\[2\]](#)
 - 키보드의 키를 예로 들면, 키보드의 키를 사용하는 방법은 '누른다'이다. [\[2\]](#)
 - 하지만 똑같은 동작 방법의 키라고 하더라도 ESC는 취소를 ENTER는 실행의 목적을 가지고 있다. [\[2\]](#)
 - 결과적으로 다형성이란, 동일한 조작방법으로 동작시키지만 동작방법은 다른 것을 의미한다. [\[2\]](#)
2. 오버로딩은 가장 이해하기 쉬운 다형성의 예.
 - 같은 이름이지만 서로 다른 동작 방법을 가지고 있기 때문에 오버로딩은 다형성의 한 예라고 할 수 있다. [\[2\]](#)
3. 인터페이스는 메서드만을 정의하고 인터페이스를 데이터 타입으로 하는 인스턴스는 마치 메서드만을 가지고 있는 것처럼 동작. [\[2\]](#)
 - 이것은 인터페이스의 매우 중요한 특징 중의 하나 [\[2\]](#)
 - 인스턴스의 데이터 타입을 인터페이스로 한다는 것은 인스턴스를 외부에서 제어할 수 있는 조작 장치를 인스턴스 멤버로 제한한다는 의미. [\[2\]](#)
4. 자바에서는 다형성을 위해 부모 클래스 타입의 참조 변수로 자식 클래스 타입의 인스턴스를 참조할 수 있도록 한다.
 - 이때 참조 변수가 사용할 수 있는 멤버의 개수가 실제 인스턴스의 멤버 개수보다 같거나 적어야 참조가능. [\[3\]](#)

[1] <https://wikidocs.net/269>, 05-7 다형성

[2] <https://opentutorials.org/course/1223/6127>

[3] http://tcpschool.com/java/java_polymorphism_concept

2. 수업 내용 정리 [Polymorphism]

5. 어떤 것을 대충 통 치는 것

- Ex] 핸들은 있고, 오른 페달이 액셀, 왼쪽이 브레이크, 세부적인 부분 부분은 다르지만, 어쨌든 대충 보면 어떤 차도 자동차이다.
- 애매한 선언 = new 상세 정의

6. 인스턴스를 애매하게 통치기

- 상속에 의한 is - a 관계 성립 시, 인스턴스를 부모 클래스 타입의 변수에 대입 가능
- 부모 클래스 타입 변수에 대입하는 것으로, 통 칠 수 있음

예제

```
class Parent { ... }  
class Child extends Parent { ... }  
...  
Parent pa = new Parent(); // 허용  
Child ch = new Child(); // 허용  
Parent pc = new Child(); // 허용  
Child cp = new Parent(); // 오류 발생.
```

7. 참조 변수가 사용할 수 있는 멤버의 개수가 실제 인스턴스의 멤버 개수와 같기 때문에, 특정 타입의 참조 변수로는 당연히 같은 타입의 인스턴스를 참조 가능 [3]
8. 참조 변수가 사용할 수 있는 멤버의 개수가 실제 인스턴스의 멤버 개수보다 적으므로 부모 클래스 타입의 참조 변수로도 자식 클래스 타입의 인스턴스를 참조 가능 [3]
9. But, 참조 변수가 사용할 수 있는 멤버의 개수가 실제 인스턴스의 멤버 개수보다 많은, 반대의 경우인 자식 클래스 타입의 참조 변수로는 부모 클래스 타입의 인스턴스를 참조 불가 [3]
10. 클래스는 상속을 통해 확장은 가능하나 축소는 불가 [3]
 - 자식 클래스에서 사용할 수 있는 멤버의 개수가 언제나 부모 클래스와 같거나 많음

[3] http://tcpschool.com/java/java_polymorphism_concept

2. 수업 내용 정리 [Polymorphism]

- ✓ 상자의 타입 과 내용의 타입의 역할
 - 어떤 멤버를 이용할 수 있는가는 상자의 타입이 결정한다.
 - 멤버가 어떻게 움직이는지는 내용의 타입이 결정한다.
- ✓ 취급 변경
 - 캐스트 연산자를 이용하면, 강제 대입이 가능
 - 부정확한 대입 발생 시, ClassCastException 발생
- ✓ 다형성
 - 다른 인스턴스를 동일시하여, 부모 클래스 타입의 배열에 담을 수 있다.
 - 마찬가지로, 부모 클래스 타입의 인수나 리턴 값을 이용하여, 다른 클래스를 모아서 처리 가능
 - 동일시 취급해도, 각각의 인스턴스는 각 클래스의 정의를 따르고 다른 동작을 한다.

2. 수업 내용 정리 [문자열 조작]

1. 문자열 처리 (결합)

❖ "Hello" + " Java" ⇨ "Hello Java"

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "Hello";  
        String s2 = " Java";  
        System.out.println(s1 + s2);  
    }  
}
```

GameMain ×
"C:\Program Files\Java\jdk-13.0.2\bin\java.
Hello Java
Process finished with exit code 0

2. 문자열 처리 (일부 떼어내기)

❖ "Hello" ⇨ "He"

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "HELLO";  
        System.out.println(s1.substring(0,2));  
    }  
}
```

GameMain ×
"C:\Program Files\Java\jdk-13.0.2\bin\java.ex
HE
Process finished with exit code 0

2. 수업 내용 정리 [문자열 조작]

3. 문자열 처리 (일부 치환)

❖ "HELLO" ⇨ "HEXXO"

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "HELLO";  
        System.out.println(s1.replace( target: "LL", replacement: "XX"));  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program
HEXXO

Process finished with exit code 0

4. 문자열 처리 (분리)

❖ "1,2,3" ⇨ "1", "2", "3"

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "1,2,3";  
        String[] splited = s1.split( regex: ",");  
        for (String s : splited) {  
            System.out.println(s);  
        }  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"

1
2
3

Process finished with exit code 0

2. 수업 내용 정리 [문자열 조작]

5. 문자열 처리 (대소문자 변경)

❖ "HELLO" ⇨ "hello"

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "HELLO";  
        System.out.println(s1.toLowerCase());  
    }  
}
```

GameMain ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"  
hello
```

Process finished with exit code 0

6. 문자열 처리 (검색)

❖ "HELLO" ⇨ E는 2번째 글자

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "HELLO";  
        System.out.println(s1.indexOf('E'));  
    }  
}
```

GameMain ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"  
1
```

Process finished with exit code 0

2. 수업 내용 정리 [문자열 조작]

7. 문자열 처리 (내용 비교)

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "Java";  
        String s2 = "java";  
        if (s1.equals(s2)) {  
            System.out.println("s1과 s2는 같다");  
        }  
        if (s1.equalsIgnoreCase(s2)) {  
            System.out.println("s1과 s2는 대소문자 무시하면 같다");  
        }  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Pr
s1과 s2는 대소문자 무시하면 같다

Process finished with exit code 0

8. 문자열 처리 (길이)

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "Java";  
        System.out.println(s1.length()); // 4  
        System.out.println(s1.isEmpty()); // false  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-

4

false

length() : 길이

isEmpty : 길이가 0인지

Process finished with exit code 0

2. 수업 내용 정리 [문자열 조작]

9. 문자열 처리 (검색)

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "Java and JavaScript";  
        System.out.println(s1.contains("Java"));  
        System.out.println(s1.endsWith("Java"));  
        System.out.println(s1.indexOf("Java"));  
        System.out.println(s1.lastIndexOf("Java"));  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaa

true	contains(): 포함 관계
false	endsWith() : 끝나는 단어가 맞는지
0	indexOf() : 단어가 몇 번째에 있는지
9	lastIndexOf() : 뒤에서 몇 번째에 단어가 있는지

Process finished with exit code 0

10. 문자열 처리 (변환)

```
public class GameMain {  
    public static void main(String[] args) {  
        String s1 = "Java and JavaScript";  
        System.out.println(s1.toLowerCase()); // 소문자로  
        System.out.println(s1.toUpperCase()); // 대문자로  
        System.out.println(s1.trim()); // 좌우 공백 제거  
        System.out.println(s1.replace("and", "or")); // 교체  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetB

java and javascript

JAVA AND JAVASCRIPT

Java and JavaScript

Java or JavaScript

Process finished with exit code 0

2. 수업 내용 정리 [문자열 조작]

문자열 결합 방법

- 1) + 연산
- 2) StringBuilder
- 3) StringBuffer

StringBuilder

- append() 메서드로 결합한 결과를 내부 메모리 (버퍼)에 담아 두고 toString()으로 결과를 얻음
- String을 + 연산자로 더하는 것은 매우 비효율적임
 - 자바 뿐만 아니라 다른 언어에서도 마찬가지임
 - "Hello" 라는 String이 저장되는 메모리 할당
 - "Alice"라는 String이 저장되는 메모리 할당
 - +연산시 "Hello Alice" 라는 String이 저장되는 메모리 또 할당
 - Hello 메모리와 Alice 메모리는 활용 없이 참조 끊고 GC 대상됨
 - 연산 숫자가 증가할수록 메모리 증가 속도가 메모리 정리속도보다 빨라짐. 메모리 부족으로 프로그램 다운 가능

[illegible]

2. 수업 내용 정리 [문자열 조작]

- 이러한 +연산을 사용한 문자열 조작 문제를 해결하기 위한 클래스가 StringBuilder와 StringBuffer 클래스.
- StringBuilder는 새로운 메모리를 계속 할당하는 방식이 아니라 기존에 존재하는 메모리에 String을 계속 추가하는 방식으로 작동하므로, + 연산보다 훨씬 효율적임
- StringBuffer는 Thread를 배워야 이해할 수 있음. 나중에 알려주신다하심.

```
public class GameMain {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder();  
        for (int i = 0; i < 10000 ; i++) {  
            sb.append("Java"); //결합  
        }  
        String s = sb.toString(); // 완성된 결과  
        System.out.println(s);  
    }  
}
```

GameMain ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "  
JavaJavaJavaJavaJavaJavaJavaJavaJavaJavaJavaJavaJavaJavaJava
```

Process finished with exit code 0

2. 수업 내용 정리 [Regular Expression]

정규 표현식 (Regular Expression, regexp)

- 자바 뿐만 아니라 거의 모든 언어에서 사용되며 특정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는 형식 언어
- 일반적인 조건에 따른 처리

```
boolean isValidPlayerName(String name) {  
    if (name.length != 8) {  
        return false;  
    }  
    char first = name.charAt(0);  
    if (!(first >= 'A' && first <= 'z')) {  
        return false;  
    }  
    for (int i = 1; i < 8; i++) {  
        char c = name.charAt(i);  
        if (!((c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9'))) {  
            return false;  
        }  
    }  
}
```

정규표현식 활용 예

```
boolean isValidPlayerName(String name) {  
    return name.matches("[A-Z][A-Z0-9]{7}");  
}
```

2. 수업 내용 정리 [Regular Expression]

정규 표현 기본 문법

1. 문자: 그 문자가 아니면 false

```
String s = "Java";
s.matches( regex: "Java"); // true
s.matches( regex: "JavaJava"); // false
s.matches( regex: "java"); // false
```

2. 점(.) : 임의의 한 문자

```
"Java".matches( regex: "J.va"); // true
```

3. * : 앞 문자 0회 이상 반복

```
"Jaaaaava".matches( regex: "Ja*va"); // true
"abcd한글011".matches( regex: ".*"); // true
"Java".matches( regex: "Ja.*"); // Ja로 시작하는 문자열
"Java".matches( regex: ".*va"); // va로 끝나는 문자열
```

4. {} : 지정 횟수만큼 반복

패턴	의미
{n}	앞 문자가 n번 반복
{n,}	앞 문자가 n번 이상 반복
{n,m}	앞 문자 n번 이상 m번 이하 반복
?	앞 문자 0번 또는 1번
+	앞 문자 1번 이상 반복

5. [] : 어느 한 문자

Ex] UR[LIN] :
첫번째 문자는 U, 두번째 문자는 R,
세 번째 문자는 'L, I, N' 중 하나

2. 수업 내용 정리 [Regular Expression]

6. [-] : 지정 범위 중 한 문자

```
"url".matches( regex: "[a-z]{3}"); // true
```

정규표현을 활용한 예

split() 메서드 : 문자열 분기

```
public class GameMain {  
    public static void main(String[] args) {  
        String s = "abc,def:ghi";  
        String[] words = s.split( regex: "[,:]" );  
        for (String word : words) {  
            System.out.println(word + "->");  
        }  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"

abc->

def->

ghi->

Process finished with exit code 0

7. ^ : 맨 앞, \$: 맨 뒤

Ex] "^j.*p\$"

⇒ 맨 앞 글자가 j, 맨 뒤 문자가 p인 문자열

replaceAll() 메서드 : 문자열 교체

```
public class GameMain {  
    public static void main(String[] args) {  
        String s = "abc,def:ghi";  
        String words = s.replaceAll( regex: "[beh]", replacement: "X" );  
        System.out.println(words); // aXc,dXf:gXi  
    }  
}
```

// static void printStatus(Hero hero) {

// System.out.println(hero.getName() + "의 상태");

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program

aXc,dXf:gXi

Process finished with exit code 0

2. 수업 내용 정리 [문자열 포맷]

String.format("%s 입문 %d일 코스 %s", "Java", 14, "후반전");

⇒ Java 입문 14일 코스 후반전

```
public static String format(String format,  
                             Object... args)
```

Returns a formatted string using the specified format string and arguments.

The locale always used is the one returned by `Locale.getDefault()`.

% [기호] [갯수] [형태]

, : 세 자리마다 콤마를 삽입
0 : 남은 영역을 0으로 채움
- : 왼쪽 정렬 (숫자)
+ : 기호 강제 표시

표시할 문자 갯수
n.m 형태로 작성한 경우는
전체 n개, 소숫점 m 자리까지

d : 정수
s : 문자열
f : 소수
b : 불리언

```
public static void main(String[] args) {  
    System.out.printf("제품번호%s-%02d", "ABC", 3); //제품번호ABC-03  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program F

제품번호ABC-03

Process finished with exit code 0



3. 실습화면 캡처

3. 실습 화면 캡처

13-1. 인터페이스를 변수의 타입으로 사용하기

```
3 public interface Life {  
4 }  
  
public class GameMain {  
    public static void main(String[] args) {  
        Life life = new Wizard(); // 살아있는 것으로 통  
    }  
}
```

13-2. attack과 fireball을 사용하는 마법사

```
3 public abstract class Character {  
4     String name;  
5     int hp;  
6  
7     public abstract void attack(Kinoko kinoko);  
8     public void run() { System.out.println(name + "은 도망쳤다"); }  
11 }  
  
3 public class Wizard extends Character {  
4     int mp;  
5  
6     @Override  
7     public void attack(Kinoko kinoko) {  
8         System.out.println(this.name + "의 공격!");  
9         System.out.println("적에게 3포인트의 데미지");  
10        kinoko.setHp(kinoko.getHp() - 3);  
11    }  
12  
13    @Override  
14    public void fireball(Kinoko kinoko) {  
15        System.out.println(this.name + "는 불의 구슬을 맞았다!");  
16        System.out.println("적에게 20포인트의 데미지");  
17        kinoko.setHp(kinoko.getHp() - 20);  
18        this.mp -= 5;  
19    }  
20 }
```

3. 실습 화면 캡처

13-3. attack, fireball을 사용하는 예

```
public class GameMain {  
    public static void main(String[] args) {  
        Wizard w = new Wizard();  
        Kinoko k = new Kinoko();  
        w.name = "레이나";  
        w.attack(k);  
        w.fireball(k);  
    }  
}
```

GameMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.
레이나의 공격!
적에게 3포인트의 데미지
레이나는 불의 구슬을 맞았다!
적에게 20포인트의 데미지

Process finished with exit code 0

13-4. 잘못 된 예

```
public class GameMain {  
    public static void main(String[] args) {  
        Wizard w = new Wizard();  
        Character c = w;  
        Kinoko k = new Kinoko();  
        c.name = "스랄";  
        c.attack(k);  
        c.fireball(k);  
    }  
}
```

```
3 public abstract class Monster {  
4     public void run() {  
5         System.out.println("몬스터는 도망쳤다");  
6     }  
7 }  
3 public class Slime extends Monster {  
4     @Override  
5     public void run() {  
6         System.out.println("슬라임은 슬금슬금 도망쳤다");  
7     }  
8 }
```

```
3 public class GameMain {  
4     public static void main(String[] args) {  
5         Slime slime = new Slime();  
6         Monster monster = new Slime();  
7         slime.run();  
8         monster.run();  
9     }  
10 }
```

Run: GameMain ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaa  
슬라임은 슬금슬금 도망쳤다  
슬라임은 슬금슬금 도망쳤다  
  
Process finished with exit code 0
```

13-5. 결과 예측 →

3. 실습 화면 캡처

타입 변경 방법

```
public class Main {  
    public static void main(String[] args) {  
        Character character = new Wizard();  
        Wizard wizard = (Wizard) character;  
    }  
}
```

캐스트 실패

```
public class GameMain {  
    public static void main(String[] args) {  
        Character character = new Wizard();  
        Hero hero = (Hero) character;  
    }  
}
```

캐스트 실패: 결과 콘솔 화면

```
Exception in thread "main" java.lang.ClassCastException: class  
Game.Wizard cannot be cast to class Game.Hero (Game.Wizard and  
Game.Hero are in unnamed module of loader 'app')  
    at Game.GameMain.main(GameMain.java:6)
```

Process finished with exit code 1

인스턴스의 타입 체크 instanceof

```
public class GameMain {  
    public static void main(String[] args) {  
        Character character = new Wizard();  
  
        if (character instanceof Hero) {  
            Hero hero = (Hero) character;  
        } else {  
            System.out.println("형변환 불가");  
        }  
    }  
}
```

GameMain ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"  
형변환 불가
```

Process finished with exit code 0

3. 실습 화면 캡처

13-6. 다형성의 메리트를 활용 못 한 코드

```
public class Main {  
    public static void main(String[] args) {  
        Hero h1 = new Hero();  
        Hero h2 = new Hero();  
        Thief t1 = new Thief();  
        Wizard w1 = new Wizard();  
        Wizard w2 = new Wizard();  
        // 모험개시!  
        // 우선 여관에 머물기  
        h1.setHp(h1.getHp() + 50);  
        h2.setHp(h2.getHp() + 50);  
        t1.setHp(t1.getHp() + 50);  
        w1.setHp(w1.getHp() + 50);  
        w2.setHp(w2.getHp() + 50);  
    }  
}
```

13-7. 코드의 중복 제거 (다형성의 메리트 : 동일한 타입으로 취급)

```
public class Main {  
    public static void main(String[] args) {  
        Character[] characters = new Character[5];  
        characters[0] = new Hero();  
        characters[1] = new Hero();  
        characters[2] = new Thief();  
        characters[3] = new Wizard();  
        characters[4] = new Wizard();  
        // 모험개시!  
        // 우선 여관에 머물기  
        for (Character character : characters) {  
            character.setHp(character.getHp() + 50);  
        }  
    }  
}
```

3. 실습 화면 캡처

13-8.

다형성을 활용 못 한 다양한 적을 공격하는 Hero

```
public class Hero extends Character {  
  
    @Override  
    public void attack(Kinoko kinoko) {  
        System.out.println(this.name + "의 공격!");  
        System.out.println("적에게 10포인트의 데미지를 주었다!");  
        kinoko.hp -= 10;  
    }  
  
    public void attack(Goblin goblin) {  
        System.out.println(this.name + "의 공격!");  
        System.out.println("적에게 10포인트의 데미지를 주었다!");  
        goblin.hp -= 10;  
    }  
  
    // 아래에 슬라임용 작성  
}
```

13-9. 다형성을 활용한 Hero

```
public class Hero extends Character {  
  
    @Override  
    public void attack(Monster monster) {  
        System.out.println(this.name + "의 공격!");  
        System.out.println("적에게 10포인트의 데미지를 주었다!");  
        monster.hp -= 10;  
    }  
  
}
```

13-10. 타입은 통 치고, 잘 동작하게 하기

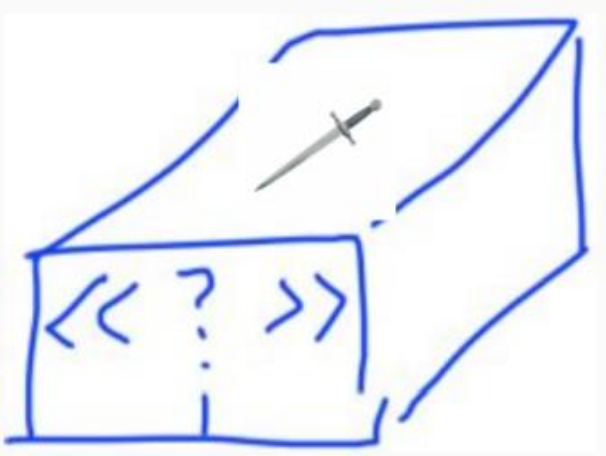
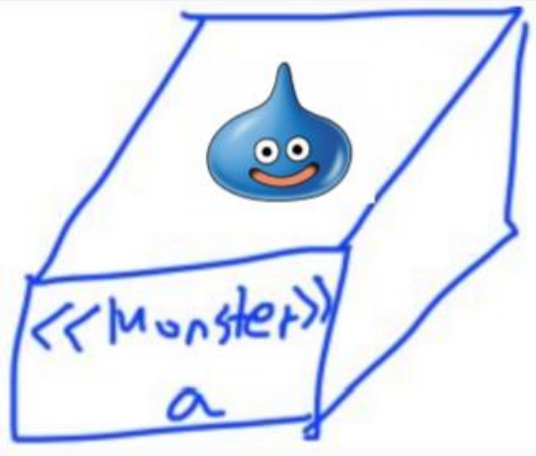
```
public class Main {  
    public static void main(String[] args) {  
        // 타입은 Monster 로 통 치기  
        Monster[] monsters = new Monster[3];  
        monsters[0] = new Slime();           // run 재정의  
        monsters[1] = new Goblin();          // run 재정의  
        monsters[2] = new DeathBat();        // run 재정의  
  
        // 동작은 안에 담긴 객체를 따름  
        for (Monster monster : monsters) {  
            monster.run();  
        }  
    }  
}
```




4. 연습문제 13-1

4. 연습문제 13-1

13-1. 문제 : 빈칸에 들어갈 적절한 클래스명을 정하시오

	(1)	(2)
코드	<code>item i = new Sword();</code>	<code>Monster a = new Slime();</code>
이미지	<div><p><u>item</u> 인스턴스</p></div>	<div><p><u>Monster</u> 인스턴스</p></div>
해설문	<p><u>Sword</u> 를 생성했지만 어쨌든 <u>Item</u> 로 보임</p>	<p><u>Slime</u> 을 생성했지만 어쨌든 <u>Monster</u> 로 보임</p>



5. 연습문제 13-2

5. 연습문제 13-2

13-2. 문제

이런 클래스가 선언되어 있다.

다음 물음에 답하시오

1. `X obj = new A();`로 인스턴스를 생성한 후, 변수 `obj`에서 호출할 수 있는 메서드를 `a()`, `b()`, `c()` 중에 골라보시오
2. `Y y1 = new A();`
`Y y2 = new B();`로 `A`와 `B`의 인스턴스를 생성한 후
`y1.a();`
`y2.a();`를 실행했을 때에 화면에 표시되는 내용을 말하시오

```
3 public class A extends Y {
4     @Override
5     public void a() { System.out.println("Aa"); }
8     @Override
9     public void b() { System.out.println("Ab"); }
12    public void c() { System.out.println("Ac"); }
15 }

3 public class B extends Y {
4     @Override
5     public void a() { System.out.println("Ba"); }
8     @Override
9     public void b() { System.out.println("Bb"); }
12    public void c() { System.out.println("Bc"); }
15 }

3 public abstract class Y implements X {
4     public abstract void a();
5     public abstract void b();
6 }

3 public interface X {
4     void a();
5 }
```

5. 연습문제 13-2 : #1. 풀이

13-2. 문제

이런 클래스가 선언되어 있다. 다음 물음에 답하시오

1. `X obj = new A();`로 인스턴스를 생성한 후, 변수 `obj`에서 호출할 수 있는 메서드를 `a()`, `b()`, `c()` 중에 골라보시오. ⇨답: `a()`
 - 풀이:
 - `X` 자료형의 `obj`를 `A` 클래스의 인스턴스로 생성하더라도 `A` 클래스의 범위가 `X` 를 포함해 `X`보다 더 넓다.
 - 범위가 넓은 데이터를 좁은 자료형에 대입했으므로 `obj`에 저장되는 데이터는 `X` 자료형 범위로 한정된다.
 - 인터페이스 `X`에 존재하는 메서드는 `a()` 뿐이므로, 변수 `obj`가 호출할 수 있는 메서드는 `a()` 하나 뿐이다.

```
3 ▶ public class TestMain {  
4 ▶   public static void main(String[] args) {  
5       X obj = new A();  
6       obj.a();  
7       obj.b();  
8       obj.c();  
9   }  
10 }
```

컴파일 에러 발생

5. 연습문제 13-2 : #2. 풀이

2. Y y1 = new A();

Y y2 = new B();로 A와 B의 인스턴스를 생성한 후 y1.a();, y2.a();를 실행했을 때에 화면에 표시되는 내용을 말하시오

- 답: Aa

Ba

- 풀이:

- ✓ Y 클래스는 A, B의 Super Class

- Y 자료형에 A, B의 데이터를 대입하면

- Y와 교집합 부분만큼만 남아 형 축소가 발생

- ✓ y1.a() 메서드 == A의 a() 메서드

- System.out.println("Aa"); 실행

- ✓ y2.a() 메서드 == B의 a() 메서드

- System.out.println("Ba"); 실행

```
public class TestMain {  
    public static void main(String[] args) {  
        Y y1 = new A();  
        Y y2 = new B();  
  
        y1.a();  
        y2.a();  
    }  
}
```

TestMain ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"

Aa

Ba

Process finished with exit code 0



6. 연습문제 13-3

6. 연습문제 13-3

13-3. 문제

- 문제 13-2에서 이용한 A클래스나 B클래스의 인스턴스를 각각 1개씩 생성하여, 요소의 수가 2개인 1차원 배열에 차례로 담는다
- 그 후에 배열의 요소를 루프로 차례대로 꺼내 각각의 인스턴스의 b() 메서드를 호출 하여야 한다. 이상을 전제로 다음 물음에 답하시오
 1. 배열변수의 타입으로 무엇을 사용하여야 하는가?
 2. 위에서 설명하고 있는 프로그램을 작성하시오

6. 연습문제 13-3 : #1. 풀이

13-3. 문제

- 문제 13-2에서 이용한 A클래스나 B클래스의 인스턴스를 각각 1개씩 생성하여, 요소의 수가 2개인 1차원 배열에 차례로 담는다
- 그 후에 배열의 요소를 루프로 차례대로 꺼내 각각의 인스턴스의 b() 메서드를 호출 하여야 한다. 이상을 전제로 다음 물음에 답하시오
 1. 배열변수의 타입으로 무엇을 사용하여야 하는가? 답: Y
이유 :
 1. A와 B클래스의 인스턴스를 담아야 하므로, 배열 변수 Data type은 A와 B의 super 클래스 Y 이거나 Y의 implements interface인 X 이다.
 2. 각 인스턴스의 b() 메서드를 호출해야 하므로, b() 메서드가 없는 interface X는 제외된다.
 3. ∴ 배열 변수의 Data type은 abstract class인 Y 이어야 한다.

6. 연습문제 13-3 : #2. 풀이

2. 위에서 설명하고 있는 프로그램을 작성하시오.

```
public class TestMain {  
    public static void main(String[] args) {  
        A a = new A();  
        B b = new B();  
        Y[] ys = {a, b};  
  
        for(Y y : ys) {  
            y.b();  
        }  
    }  
}  
  
TestMain ×  
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"  
Ab  
Bb  
  
Process finished with exit code 0
```



7. 문자열 조작 연습문제 1-1

7. 문자열 조작] 연습문제 1-1

1-1. 문제

- 1부터 100까지의 정수를 콤마로 연결한 다음과 같은 문자열 str을 생성하는 코드를 작성하시오.
1,2,3,4,5 ... 98,99,100,
- 완성된 문자열 str을 콤마를 구분자로 하여 String 배열 a에 대입하시오.

```
public class GameMain {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder();  
        for (int i = 1; i < 101; i++) {  
            sb.append(Integer.toString(i));  
            sb.append(",");  
        }  
        String str = sb.toString();  
        String[] a = str.split("regex: ", "");  
        for (String value : a) {  
            System.out.println(value);  
        }  
    }  
}
```

StringBuilder 클래스의
인스턴스 sb 생성

for 문으로 1~100을
sb.append() 메소드로
정수, 더하기

다 더한 sb를 문자열로 변환

split 메서드로 콤마를 구분자로 하여
String array a에 저장

결과 확인을 위한 for 문

```
GameMain x  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
  
Process finished with exit code 0
```

의도한 대로
결과가 잘
출력됨을
확인함



8. 문자열 조작 연습문제 1-2

8. 문자열 조작] 연습문제 1-2

1-2. 문제

- 폴더명과 파일명을 매개변수로 받아서 연결한 파일 주소를 리턴하는 메서드를 작성하시오
- 폴더명의 경우 마지막에 \기호가 붙어 있는 경우도 있고 아닌 경우도 있으니 두 경우 모두 처리할 수 있도록 하시오

Ex] 폴더명 - C:\dev 또는 C:\dev\, 파일명 - abc.txt 일 경우 ⇒ C:\dev\abc.txt

파일 주소 리턴하는 메서드 fileAddress

```
public class Main {  
    public static String fileAddress(String folderName, String fileName) {  
        StringBuilder sb = new StringBuilder();  
        if (folderName.endsWith("\\")) {  
            sb.append(folderName).append(fileName);  
        } else {  
            sb.append(folderName).append("\\").append(fileName);  
        }  
        String fileAddr = sb.toString();  
        return fileAddr;  
    }  
}
```

결과 확인을 위한 main 메서드

```
public static void main(String[] args) {  
    String folderName = "C:\\dev"; 폴더 이름 마지막에 \가 없는 경우  
    String fileName = "abc.txt";  
    String fileAddr = fileAddress(folderName, fileName);  
  
    System.out.println(fileAddr);  
    System.out.println("=====");  
    folderName = "C:\\dev\\"; 폴더 이름 마지막에 \가 있는 경우  
    fileAddr = fileAddress(folderName, fileName);  
  
    System.out.println(fileAddr);  
}
```

8. 문자열 조작] 연습문제 1-2

```
public class Main {  
    public static String fileAddress(String folderName, String fileName) {  
        StringBuilder sb = new StringBuilder();  
        if (folderName.endsWith("\\")) {  
            sb.append(folderName).append(fileName);  
        } else {  
            sb.append(folderName).append("\\").append(fileName);  
        }  
        String fileAddr = sb.toString();  
        return fileAddr;  
    }  
}
```

```
public static void main(String[] args) {  
    String folderName = "C:\\dev"; 폴더 이름 마지막에 \가 없는 경우  
    String fileName = "abc.txt";  
    String fileAddr = fileAddress(folderName, fileName);  
  
    System.out.println(fileAddr);  
    System.out.println("=====");  
    folderName = "C:\\dev\\"; 폴더 이름 마지막에 \가 있는 경우  
    fileAddr = fileAddress(folderName, fileName);  
  
    System.out.println(fileAddr);  
}
```

```
Main X  
"C:\Program Files\Java\jdk-13.0.2\  
C:\dev\abc.txt  
===== 동일한 결과 값 출력 확인  
C:\dev\abc.txt  
  
Process finished with exit code 0
```




9. 문자열 조작 연습문제 1-3

9. 문자열 조작] 연습문제 1-3

1-3. 문제

- 다음 조건에 맞는 정규표현식을 작성하시오
 1. 모든 문자열
 2. 최초 첫번째 문자는 A, 두 번째 문자는 숫자, 세 번째 문자는 숫자이거나 아무것도 없거나
 3. 최초 첫번째 문자는 U, 2 ~4번째 문자는 영어 대문자

9. 문자열 조작] 연습문제 1-3. #1 풀이

1-3. 문제

1. 모든 문자열

답 : ^.*\$

```
public class Prac1_3Main {  
    public static void main(String[] args) {  
        String[] strList = {"aSEas@31%",  
                             "asd123ASDQW#%^$A78",  
                             "awqe@D#$F^&810"}  
    };  
    for(String str : strList) {  
        System.out.println(str.matches(regex: "^.*$"));  
    }  
}
```

Prac1_3Main ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent  
true  
true  
true
```

Process finished with exit code 0

9. 문자열 조작] 연습문제 1-3. #2 풀이

1-3. 문제

2. 최초 첫번째 문자는 A, 두 번째 문자는 숫자, 세 번째 문자는 숫자이거나 아무것도 없거나
답 : `^A[0-9]{1,2}$`

```
public class Prac1_3Main {  
    public static void main(String[] args) {  
        String[] strList = {"A7", "A78", "A810", "A1k"};  
        for(String str : strList) {  
            System.out.println(str.matches(regex: "^A[0-9]{1,2}$"));  
        }  
    }  
}
```

Prac1_3Main ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Prog

true
true
false
false

Process finished with exit code 0

9. 문자열 조작] 연습문제 1-3. #3 풀이

1-3. 문제

3. 최초 첫번째 문자는 U, 2 ~4번째 문자는 영어 대문자

답 : ^U[A-Z]{3}\$

```
public class Prac1_3Main {  
    public static void main(String[] args) {  
        String[] strList = {"UASX", "UaDA", "U3%d", "UZS", "UAQED", "USA", "UONG"};  
        for(String str : strList) {  
            System.out.println(str.matches(regex: "^U[A-Z]{3}$"));  
        }  
    }  
}
```

Prac1_3Main ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\Jet

true
false
false
false
false
false
true

Process finished with exit code 0