**자바프로그래밍2 실습과제**

**제2주 객체와 클래스 1**

클래스 선언문 바로 위에 그 클래스에 대한 설명을 주석으로 적어 넣으세요.

**main 메소드 외의 메소드에도 주석을 넣으세요.**

1. [NameCatTest](src/NameCatTest.java) 클래스에는 main 메소드만 들어 있습니다. 이 클래스는 [NameCat](src\NameCat.java) 클래스와 함께 하나의 자바 어플리케이션을 구성합니다. 이렇게 main 메소드**만**을 위한 클래스를 두면 이해는 쉽지만 쓸데없이 클래스 개수가 많아지기도 합니다. 이를 피하기 위해서 main 메소드를 이미 있는 다른 클래스 속에 집어넣어도 됩니다. NameCat 클래스에 NameCatTest 클래스의 main 메소드를 복사해 넣으세요. 그렇게 하면 NameCat 클래스 홀로 하나의 자바 어플리케이션이 됩니다. 이제 main 메소드를 갖게 된 NameCat 클래스를 실행해 보세요. (이클립스 화면 왼쪽에 있는 package explore 구역에 있는 NameCat 클래스를 마우스 오른쪽 버튼 클릭 --> Run As ... --> Java application.)

프로그램의 출력은 아래와 같습니다.



강온실 p1 문제의 답으로 main 메소드를 포함하는 NameCat 클래스를 제출하세요. 강온실에 프로그램을 제출할 때는 패키지 선언(package w2)이 없어야 합니다. public main 메소드를 갖는 NameCat 클래스는 public 클래스로 선언해야 합니다.

1. 이래 지시에 따라 프로그램을 작성하세요.

Human이라는 클래스를 작성하세요. 이 클래스는 String 타입 필드 name과 int 타입 필드 age를 갖도록 하세요. 이 필드들은 모두 private으로 선언하세요. 또 아래와 같은 다섯 개의 메소드를 갖도록 하세요.

public  void  setName(String n) // name을 설정하는 메소드

public  String  getName() // name을 반환하는 메소드

public void setAge(int i) // age를 설정하는 메소드

public int getAge() // age를 반환하는 메소드

public String toString() // Human 객체의 정보를 문자열로 반환하는 메소드

// 반환하는 문자열 형식 – Human [name=홍길동, age=19]

그리고 추가적으로 main 메소드를 작성하세요. main에서는 두 개의 Human 인스턴스들을 만들고 각 인스턴스에게 적당한 이름과 나이를 설정하도록 지시한 후, 다시 각 인스턴스에게 toString 메소드를 호출하여 그 반환된 값들을 화면에 출력합니다.

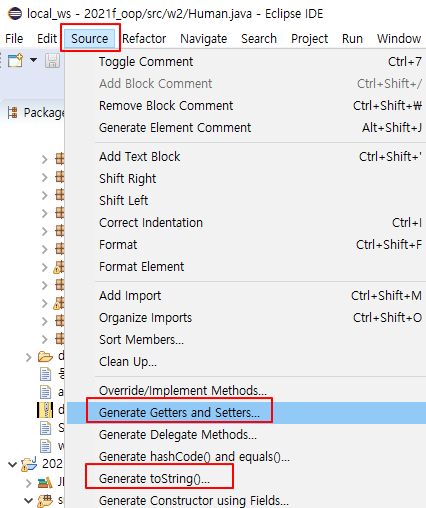
A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

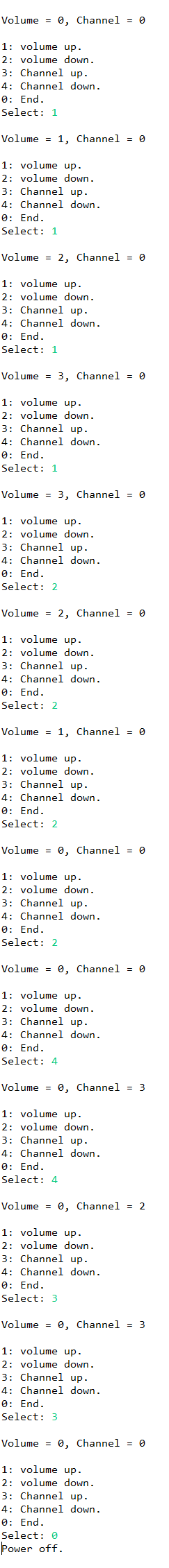
Human.java 파일을 만들고 두 개의 필드를 적어준 후, 이클립스가 제공하는 소스 자동 생성 기능을 사용하면 편리합니다. 단, 주석은 직접 입력해야 합니다.



강온실 p2 문제의 답으로 Human 클래스를 제출하세요. 강온실에 프로그램을 제출할 때는 패키지 선언(package w2)이 없어야 합니다. Human 클래스는 public 클래스이어야 합니다.

1. TV를 조정하는 리모콘을 흉내내는 Remocon 클래스를 작성하세요. 이 Remocon은 채널 조정과 음량 조정 기능만 있습니다. 채널은 0번부터 3번까지 있고 3번 채널에 있을 때 채널을 올리면 0번 채널로 바뀝니다. 0번 채널에 있을 때 채널을 내리면 3번 채널로 갑니다. 음량도 0에서 3까지 조정할 수 있습니다. 음량이 3일 때 음량을 더 올리려고 시도해도 3 상태가 그대로 유지됩니다. 음량이 0일 때 음량을 더 내리려고 해도 0 상태가 유지됩니다.

또, Remocon 클래스를 테스트하는 RemoconTest 클래스를 작성하고 이 곳에 main 메소드를 두어 아래와 같이 작동하도록 하세요.



Remocon 클래스 뼈대는 아래와 같습니다. 이 클래스의 인스턴스들을 만들면 각 인스턴스는 각자의 volume, channel 메모리를 가지고 각자의 음량과 채널을 이 메모리에 기억합니다. 각 Remocon 인스턴스에는 아래 그림에 있는 다섯 개의 메소드를 호출할 수 있습니다. 이 클래스에서 음량, 채널의 최대값은 final, 즉 상수로 선언되어 있습니다.

A screenshot of a computer program

Description automatically generated

채널이나 음량을 변경하는 각 메소드(volumeUP, volumeDown, channelUp, channelDown)는 자기의 기본 기능을 수행한 후, 새 상태를 화면에 보여주기 위해 display 메소드를 호출해야 합니다.

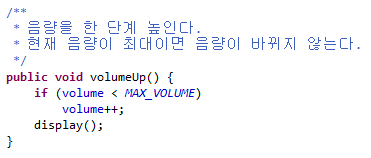
RemoconTest 클래스의 main 메소드는 아래와 같이 구현할 수 있습니다.

A screenshot of a computer program

Description automatically generated

**위 프로그램을 작성할 때 주석을 위에 있는 그대로 입력할 필요는 없지만 현재 주석이 있는 모든 위치에 현재와 비슷한 정도의 상세함을 유지하도록 주석을 달아야 합니다.**

힌트



참고:

아래 선언문은 최대 음량을 의미하는 MAX\_VOLUME의 값이 3이라고 정해 줍니다.

public final static int MAX\_VOLUME = 3; // 최고 음량

이 값은 이 클래스뿐 아니라 다른 클래스에서도 참고할 필요가 있을 수 있으므로 public으로 선언했습니다.

또 한 번 값이 정해지면 바꿀 수 없도록 (즉 상수로) 만들기 위해 final로 선언했습니다.

static 선언은 MAX\_VOLUME 값이 각각의 개별 인스턴스마다 존재하는 대신 Remocon 클래스 전체적으로 하나만 존재하도록 해 줍니다. 상수를 각 인스턴스가 중복해서 갖고 있을 필요가 없기 때문입니다.

강온실에 소스코드를 붙여넣기 할 때는 Remocon과 RemoconTest 등 두 개의 클래스를 붙여 넣어야 합니다. 이 때 main 메소드를 갖고 있는 RemoconTest만 public으로 선언하고, Remocon 클래스는 public으로 선언하지 않아야 합니다. 강온실은 학생이 제출하는 클래스들을 하나의 파일로 저장하고 컴파일/실행합니다. 자바에서 한 파일에 여러 개의 클래스가 있을 수는 있지만 그 중 main 메소드를 갖고 있는 하나의 클래스만 public 클래스이어야 하고 그 클래스의 이름은 파일 이름과 같아야 합니다 강온실의 자바실행기(java interpreter)는 public으로 선언된 클래스의 main 메소드를 호출합니다.

import 문장은 맨 위에 있어야 합니다. Import 문장이 두 클래스 사이에 있으면 안 됩니다.

package 문장은 없어야 합니다.

1. 아래와 같은 뼈대를 갖는 Counter 클래스를 작성하고 이 클래스의 작동을 테스트하는 CouterTest 클래스를 작성하세요.

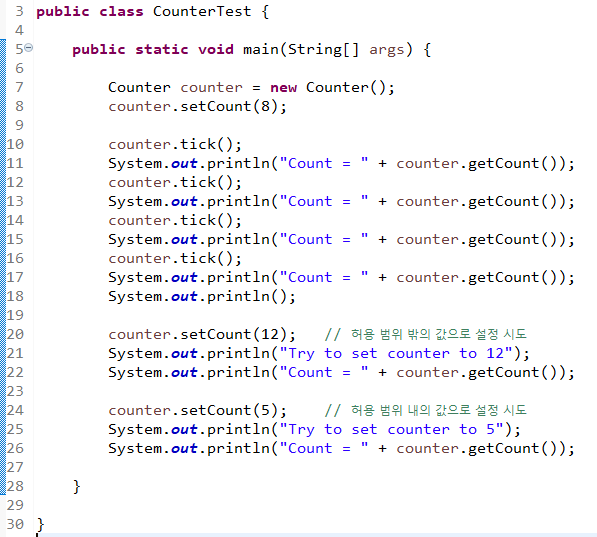


위 코드에서 마지막 메소드인 isInputValid 메소드는 외부의 사용자를 위한 메소드가 아니고 클래스 내부적으로 필요해서 만든 보조적인 메소드(helper method)입니다. 즉 외부 사용자를 위한 메소드가 아닙니다. 그래서 이 메소드는 private으로 선언했습니다. 주어진 수가 유효한 수이면 true를, 그렇지 않으면 false를 반환합니다.

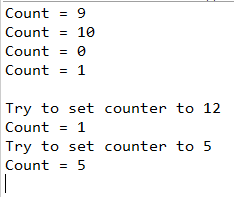
참고:

private 메소드인 isInputValid 메소드에 주석을 달 때는 javadoc에서 통용되는 주석 형식을 따르지 않고 프로그래머의 편의에 따라 주석을 달았습니다. /\*\* 주석 \*/ 형식의 주석은 api 문서에 나타나는 반면 /\* 주석 \*/ 형태의 주석은 api 문서에 나타나지 않습니다. 위 프로그램에서 두 가지 형태의 주석이 다른 색깔로 보임에 유의하세요.

아래는 main 메소드 예입니다.



이 main 메소드를 실행한 결과는 아래와 같습니다.



강온실에 소스코드를 붙여넣기 할 때는 Counter와 CounterTest 등 두 개의 클래스를 붙여넣어야 합니다. 이 때 main 메소드를 갖고 있는 CounterTest만 public으로 선언하고, Counter 클래스는 public으로 선언하지 않아야 합니다.

1. 위 4번에서 작성한 Counter 클래스에서는 최대 수가 상수로 고정되어 있습니다. 이렇게 하면 10까지 세는 카운터를 이용하다가 12까지 세는 카운터가 필요하면 소스 코드를 수정하고 다시 컴파일해야 합니다. 다시 컴파일하는 게 개발 단계에서는 별 일 아니지만 실제로 현장에 설치되어 운영되고 있는 상황에서는 별 (심각한) 일이 됩니다. 이 문제를 해결하려면 최대값을 상수로 선언하는 대신 변수로 선언하고 이 변수를 임의로 설정하는 public 메소드를 두면 됩니다.

상수 MAX 대신 변수 max를 선언하고 setMax 메소드를 추가하세요. 클래스 이름은 Counter2로 지으세요. 또 두 개의 Counter2 인스턴스를 구성한 후 하나는 0부터 12까지 세도록 하고 다른 하나는 0부터 60까지 세도록 설정한 후 이들에게 적절한 메소드들을 호출하여 기능을 테스트하는 main 메소드를 Counter2 안에 작성해 넣으세요.

<프로그램 예시>

Graphical user interface, text, application

Description automatically generated

<예시 프로그램>의 실행 결과



위 예시처럼 작동하는 프로그램을 강온실에 제출하세요.

생성자 ==================================

1. 위 p2번 문제의 Human 클래스를 복사하여 Human2라는 이름으로 붙여 넣으세요. 이클립스 화면 왼쪽에 있는 패키지 익스플로러 구역에서 Human 위에서 마우스 우클릭하여 Copy한 후, 다시 Human 위에서 마우스 우클릭하여 Paste하면 됩니다. 그리고 Human2 클래스에 name과 age를 파라미터로 갖는 생성자를 넣고 이를 활용하도록 main 메소드를 수정하세요. 클래스 이름은 Human2로 하세요. 출력은 2번 문제와 같게 하세요.
2. 위 p1번 문제에서 작성한 NameCat을 복사하여 NameCat2라는 이름으로 붙여넣기 한 후, 세 개의 생성자를 추가하고 이를 활용하는 main 메소드를 아래 그림과 같이 작성하세요.

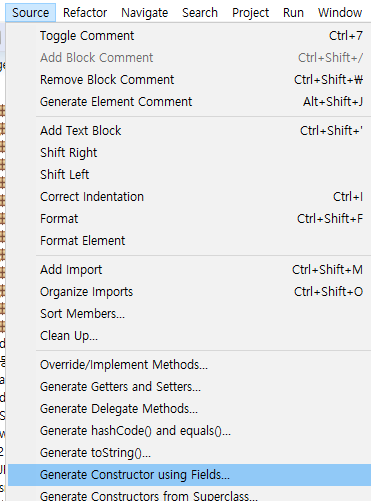


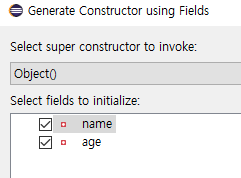
이 프로그램을 실행하면 아래와 같은 출력이 나와야 합니다. 위 main 메소드에서는 세 마리의 NameCat2들을 구성하고 이들에게 적절한 메소드를 호출하여 각각의 필드들을 설정한 후 이 세 마리를 print했습니다. 객체에 print 메소드를 호출하면 그 객체에 toString 메소드가 호출되고 그 반환값이 출력됩니다. 세 마리의 NameCat2들을 구성할 때 다중정의(overloading)된 세 가지의 생성자를 이용했습니다.

A screenshot of a computer

Description automatically generated

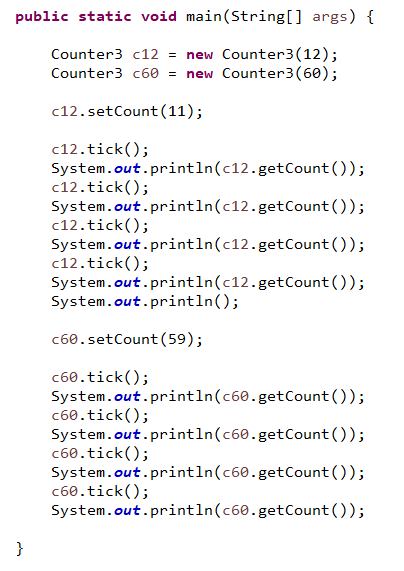
생성자를 추가할 때는 이클립스가 제공하는 소스자동생성기능을 사용하면 편리합니다.





두 개의 필드를 모두 체크하면 두 개의 파라미터를 갖는 생성자가 만들어집니다. 한 개만 선택하면 한 개의 파라미터를 갖는 생성자가 만들어집니다. 아무것도 선택하지 않으면 파라미터 없는 생성자가 만들어집니다.

1. 위 5번 문제의 Counter2 클래스를 Counter3로 복사하고 Counter3에 생성자를 하나 추가하세요. 이 생성자는 Counter3 인스턴스가 만들어질 때 최대값을 지정해 줍니다. 이렇게 하면 main에서 Counter3 인스턴스들을 구성한 후 최대값을 설정하는 메소드를 호출해 주지 않아도 됩니다. 프로그램 실행 결과는 위 p5번 문제와 동일합니다.



끝.