Examples

Week 13

클래스 상속

```
class Point {
   private int x, y;
   public void set(int x, int y) {
      this.x = x; this.y = y;
   }
   public void showPoint() {
      System.out.println(
               "(" + x + "," + y + ")");
   }
}

class ColorPoint extends Point {
   private String color;
   public void setColor(String color) {
      this.color = color;
   }
   public void showColorPoint() {
      System.out.print(color);
      showPoint();
   }
}
```

```
public class ColorPointEx {
   public static void main(String[] args)
  {
      Point p = new Point();
      p.set(1, 2);
      p.showPoint();

      ColorPoint cp = new ColorPoint();
      cp.set(3, 4);
      cp.setColor("red");
      cp.showColorPoint();
   }
}
```

```
(1,2)
red(3,4)
```

상속과 접근지정자

```
class Person
{
   private int weight;
   int age;
   protected int height;
   public String name;

   public void setWeight(int weight) {
      this.weight = weight;
   }

   public int getWeight() {
      return weight;
   }
}
```

```
InheritanceEx.java:20: error: weight
has private access in Person
                weight = 99;
                ^
1 error
```

```
class Student extends Person
{
   public void set() {
      age = 30;
      name = "James";
      height = 175;
      weight = 99;    // error
      setWeight(99);
   }
}

public class InheritanceEx
{
   public static void main(String[] args)
   {
      Student s = new Student();
      s.set();
   }
}
```

상속과 생성자 호출 (super() 호출 자동 삽입)

```
class A {
    public A() {

        System.out.println("A");
    }
}

class B extends A {
    public B() {

        System.out.println("B");
    }
}

class C extends B {
    public C() {

        System.out.println("C");
    }
}
```

```
public class ConstructorEx {
   public static void main(String[] args)
   {
      C c;
      c = new C();
   }
}
```

```
A
B
C
```

super() 호출 자동 삽입-1

```
class A {
    public A() {

        System.out.println("A");
    }

    public A(int x) {

        System.out.println("A-2");
    }
}

class B extends A {
    public B() {

        System.out.println("B");
    }
}
```

```
public class ConstructorEx2 {
   public static void main(String[] args)
   {
      B b;
      b = new B();
   }
}
```

```
A
B
```

super() 호출 자동 삽입-2

```
class A {
    //public A() {
    //
    //    System.out.println("A");
    //}

    public A(int x) {

        System.out.println("A-2");
    }
}

class B extends A {
    public B() {

        System.out.println("B");
    }
}
```

```
public class ConstructorEx2 {
   public static void main(String[] args)
   {
      B b;
      b = new B();
   }
}
```

```
ConstructorEx2.java:12: error: constructor
A in class A cannot be applied to given
types;
        public B() {
                   ^
  required: int
  found: no arguments
  reason: actual and formal argument lists
differ in length
1 error
```

super() 호출 자동 삽입-3

```
class A {
    public A() {

        System.out.println("A");
    }

    public A(int x) {

        System.out.println("A-2");
    }
}

class B extends A {
    public B() {

        System.out.println("B");
    }

    public B(int x) {

        System.out.println(“B-2");
    }
}
```

```
public class ConstructorEx3 {
   public static void main(String[] args)
   {
      B b;
      b = new B(5);
   }
}
```

```
A
B-2
```

명시적 super() 호출-1

```
class A {
    public A() {

        System.out.println("A");
    }

    public A(int x) {

        System.out.println("A“ + x);
    }
}

class B extends A {
    public B() {

        System.out.println("B");
    }

    public B(int x) {
        super(x);
        System.out.println(“B“ + x);
    }
}
```

```
public class ConstructorEx4 {
    public static void main(String[] args)
    {
        B b;
        b = new B(5);
    }
}
```

```
A5
B5
```

명시적 super() 호출-2

```
class Point {
    private int x, y;
    public Point() {
        this.x = this.y = 0;
    }
    public Point(int x, int y) {
        this.x = x; this.y = y;
    }
    public void showPoint() {
        System.out.println(
                "(" + x + "," + y + ")");
    }
}

class ColorPoint extends Point {
    private String color;
    public ColorPoint(int x, int y,
                           String color)
    {
        super(x, y);
        this.color = color;
    }
    public void showColorPoint() {
        System.out.print(color);
        showPoint();
    }
}
```

```
public class SuperEx {
    public static void main(String [] args)
    {
         ColorPoint cp =
               new  ColorPoint(5,6,"blue");
        cp.showColorPoint();
    }
}
```

```
blue(5,6)
```

this() 호출과 super() 호출

```
class A {
    public A() {
        this(0);
        System.out.println("A");
    }
    public A(int x)
    {
        System.out.println("A" + x);
    }
}

class B extends A {
    public B() {
        this(0);
        System.out.println("B");
    }
    public B(int x) {
        super(x);
        System.out.println("B" + x);
    }
}
```

```
public class ThisAndSuper {
    public static void main(String[] args)
    {
        B b;
        b = new B();
    }
}
```

```
A0
B0
B
```

- ✓ this() 호출은 생성자의 첫번째 코드
- ✓ super() 호출은 생성자의 첫번째 코드

상속을 사용하지 않은 CarEx.java

```
class Convertible
{
   String color;
   int speed;

   public Convertible(String color) {
      this.color = color;
      speed = 0;
   }

   public void engineStart() {
      System.out.println("engine start");
   }

   public void engineStop() {
      System.out.println("engine stop");
   }
```

```
   public void speedUp() {
      speed++;
      System.out.println(
               "speed up(" + speed + ")");
   }

   public void speedDown() {
      if (speed > 0)
         speed--;
      System.out.println(
             "speed down(" + speed + ")");
   }

   public void openRoof() {
      System.out.println("open roof");
   }

   public void closeRoof() {
      System.out.println("close roof");
   }
}
```

상속을 사용하지 않은 CarEx.java

```
class DumpTruck
{
   String color;
   int speed;
   int cargoSize;

   public DumpTruck(String color,
                        int cargoSize) {
      this.color = color;
      speed = 0;
      this.cargoSize = cargoSize;
   }

   public void engineStart() {
      System.out.println("engine start");
   }

   public void engineStop() {
      System.out.println("engine stop");
   }
```

```

   public void speedUp() {
      speed++;
      System.out.println(
               "speed up(" + speed + ")");
   }

   public void speedDown() {
      if (speed > 0)
         speed--;
      System.out.println(
             "speed down(" + speed + ")");
   }

   public void dump() {
      System.out.println("dump");
   }
}
```

상속을 사용하지 않은 CarEx.java

```
class FreezerTruck
{
   String color;
   int speed;
   int cargoSize;
   int temperature;

   public FreezerTruck(String color,
               int cargoSize, int temp) {
      this.color = color;
      speed = 0;
      this.cargoSize = cargoSize;
      temperature = temp;
   }

   public void engineStart() {
      System.out.println("engine start");
   }

   public void engineStop() {
      System.out.println("engine stop");
   }
```

```
   public void speedUp() {
      speed++;
      System.out.println(
               "speed up(" + speed + ")");
   }

   public void speedDown() {
      if (speed > 0)
         speed--;
      System.out.println(
             "speed down(" + speed + ")");
   }

   public void setTemperature(int temp) {
      temperature = temp;
      System.out.println(
   "set temperature(" + temperature + ")");
   }
}
```

상속을 사용하지 않은 CarEx.java

```
public class CarEx
{
   public static void main(String[] args)
   {
      Convertible c =
                    new Convertible("red");
      System.out.println("Convertible: ");
      c.engineStart();
      c.speedUp();
      c.openRoof();
      c.closeRoof();
      c.speedDown();
      c.engineStop();
      System.out.println();

      DumpTruck d =
                new DumpTruck("blue", 10);
      System.out.println("Dump truck: ");
      d.engineStart();
      d.speedUp();
      d.speedDown();
      d.dump();
      d.engineStop();
      System.out.println();
```

```
Convertible:
engine start
speed up(1)
open roof
close roof
speed down(0)
engine stop
```

```
Dump truck:
engine start
speed up(1)
speed down(0)
dump
engine stop
```

```
      FreezerTruck f =
           new FreezerTruck("blue", 10, 0);
      System.out.println("Freezer truck: ");
      f.engineStart();
      f.setTemperature(-20);
      f.speedUp();
      f.speedDown();
      f.engineStop();
      System.out.println();
   }
}
```

```
Freezer truck:
engine start
set temperature(-20)
speed up(1)
speed down(0)
engine stop
```

상속을 사용하는 CarEx.java

- 상속을 사용하도록 CarEx.java를 다시 작성하시오.
  - 필요하면 새로운 클래스를 추가하시오.
  - 코드 중복을 최소화 하시오.
  - CarEx 클래스(main 메소드)는 변경하지 마시오.
  - 출력 결과는 동일해야 함.