Main

package com.ezidayzi;  
  
public class Main {  
  
 public static void main(String[] *args*) {  
 *// 1. 객체지향 프로그램의 시작  
 // (1) 클래스를 만들고 해당 클래스의 인스턴스를 만들어보세요.  
 // 필드와 메소드를 구현하되, 그 중 일부는 접근제한자(private 및 public) 및 겟터/셋터를 사용해 봅니다.* Student kimyoonseo = new Student();  
 kimyoonseo.setName("김윤서");  
 kimyoonseo.setAge(23);  
 kimyoonseo.setGraduation(false);  
 kimyoonseo.setStudentId(20192666);  
 System.*out*.println(kimyoonseo.getName());  
 System.*out*.println(kimyoonseo.getAge());  
 System.*out*.println(kimyoonseo.getStudentId());  
 kimyoonseo.isGraduate();  
 String introduction = kimyoonseo.makeIntroduction();  
 System.*out*.println(introduction);  
 System.*out*.println("==========================================");  
  
 *// 1. 클래스의 상속관계 이해  
 // (1) 부모클래스와 자식클래스를 만들어봅니다. 자식 클래스는 2개 이상 만듭니다.  
 // - 필드와 메소드를 구현하되, 그 중 일부는 접근제한자(private, protected, public) 를 사용해 봅니다.  
 // (1) 부모클래스와 자식클래스의 인스턴스를 생성하고 각 인스턴스에 대하여 소속 클래스에서 정의된 필드와 메소드를 사용해봅니다.  
 // 특히 자식 클래스에서 부모 클래스의 메소드를 오버라이딩해서 구현해 봅니다.  
 // (2) 부모클래스 변수에 자식클래스의 인스턴스를 대입하는 다형성을 체험해봅니다.* Cafe starbucks = new Starbucks();  
 System.*out*.println(starbucks.mainMenu);  
 starbucks.makeCoffee();  
 Cafe twosome = new Twosome();  
 twosome.makeCoffee();  
  
 System.*out*.println("==========================================");  
 *// 추상클래스 사용  
 // (1) 하나의 추상클래스를 만들고 이를 상속받는 2개 이상의 자식클래스를 만듭니다.  
 // (2) 부모클래스 변수에 자식클래스의 인스턴스를 대입해 사용하는 다형성을 체험해봅니다.  
 Animal* dog = new Dog();  
 dog.move();  
 dog.sing();  
  
 *Animal* duck = new Duck();  
 duck.move();  
 duck.sing();  
 System.*out*.println("==========================================");  
 *// 인터페이스 사용  
 // (1) 인터페이스를 정의하고 이를 구현하는 클래스를 만듭니다.  
 // (2) 위 클래스의 인스턴스를 만들고 인터페이스에서 정의한 메소드를 호출해 보세요.  
  
 CarSpeed* korea = new KoreaCarSpeed();  
 Integer koreaCarSpeed = korea.speed(100);  
 System.*out*.println(koreaCarSpeed);  
  
 *CarSpeed* japan = new JapanCarSpeed();  
 Integer japanCarSpeed = japan.speed(150);  
 System.*out*.println(japanCarSpeed);  
 System.*out*.println("==========================================");  
  
 *// 정적 필드와 정적 메소드의 사용  
 // (1) 정적 필드와 정적 메소드를 사용하는 클래스를 만듭니다. 위 문제(1~4)들에서 구현한 클래스들 중 하나를 가져와서 변형해서 사용해도 됩니다.  
 // (2) 정적 필드에 접근하고 정적 메소드를 호출해 봅니다. 이 경우 해당 클래스에 대한 인스턴스를 만들 필요는 없습니다.  
 // 인스턴스 생성은 오직 인스턴스 필드와 인스턴스 메소드를 사용할 때만 필요합니다.* String univInfo = SoongsilStudent.*getSchoolInfo*();  
 System.*out*.println(univInfo);  
  
 String univName = SoongsilStudent.*schoolName*;  
 System.*out*.println(univName);  
  
 }  
}

1)

package com.ezidayzi;  
  
*// 1. 객체지향 프로그램의 시작  
// (1) 클래스를 만들고 해당 클래스의 인스턴스를 만들어보세요.*public class Student {  
 private String name;  
 private Integer age;  
 private Integer studentId;  
 private Boolean graduation;  
  
 public String makeIntroduction() {  
 return "안녕하세요 저는 " + name + " 입니다.";  
 }  
  
 *//Getter* public String getName() {  
 return name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public int getStudentId() {  
 return studentId;  
 }  
  
 public void isGraduate() {  
 if (graduation) {  
 System.*out*.println("졸업생입니다.");  
 } else {  
 System.*out*.println("재학생입니다.");  
 }  
 }  
  
 *// Setter* public void setName(String *name*) {  
 this.name = *name*;  
 }  
  
 public void setAge(int *age*) {  
 this.age = *age*;  
 }  
  
 public void setStudentId(int *studentId*) {  
 this.studentId = *studentId*;  
 }  
  
 public void setGraduation(boolean *graduation*) {  
 this.graduation = *graduation*;  
 }  
}

2)

부모 클래스 – Café

package com.ezidayzi;  
  
public class Cafe {  
 public String cafeName;  
 protected String mainMenu;  
 protected Integer price;  
 private Integer menuCount;  
  
 public void makeCoffee() {  
 System.*out*.println(cafeName+"에서 판매하는 "+mainMenu+"은/는 "+ price+"원 입니다.");  
 }  
}

자식클래스 – Starbucks, Twosome

package com.ezidayzi;  
  
*// 자식 클래스*public class Starbucks extends Cafe {  
 Starbucks () {  
 this.cafeName = "스타벅스";  
 this.mainMenu = "돌체 라떼";  
 this.price = 5000;  
 *// this.menuCount = 3; private 이므로 접근 불가* }  
  
 @Override  
 public void makeCoffee() {  
 System.*out*.println("스타벅스에 오신것을 환영합니다.");  
 super.makeCoffee();  
 }  
}

package com.ezidayzi;  
  
public class Twosome extends Cafe {  
 Twosome () {  
 this.cafeName = "투썸플레이스";  
 this.mainMenu = "스트로베리 초코 생크림 케이크";  
 this.price = 6000;  
 }  
  
 @Override  
 public void makeCoffee() {  
 System.*out*.println("투썸플레이스에 오신것을 환영합니다.");  
 super.makeCoffee();  
 }  
}

3)

추상클래스 Animal

package com.ezidayzi;  
  
public abstract class *Animal* {  
 String name;  
 abstract void sing();  
 abstract void move();  
}

Dog, Duck 구현

package com.ezidayzi;  
  
public class Dog extends *Animal* {  
 Dog() {  
 this.name = "강아지";  
 }  
  
 @Override  
 void sing() {  
 System.*out*.println("멍멍멍");  
 }  
  
 @Override  
 void move() {  
 System.*out*.println("와다다");  
 }  
}

package com.ezidayzi;  
  
public class Duck extends *Animal*{  
 Duck() {  
 this.name = "오리";  
 }  
  
 @Override  
 void sing() {  
 System.*out*.println("꽥꽥꽥");  
 }  
  
 @Override  
 void move() {  
 System.*out*.println("뒤뚱뒤뚱");  
 }  
}

4)

인터페이스

package com.ezidayzi;  
  
public interface *CarSpeed* {  
 int speed(int *speed*);  
}

package com.ezidayzi;  
  
public class JapanCarSpeed implements *CarSpeed* {  
 private static final int *MAX\_SPEED* = 130;  
  
 @Override  
 public int speed(int *speed*) {  
 return Math.*min*(*speed*, *MAX\_SPEED*);  
 }  
}

package com.ezidayzi;  
  
public class KoreaCarSpeed implements *CarSpeed* {  
 private static final int *MAX\_SPEED* = 100;  
  
 @Override  
 public int speed(int *speed*) {  
 return Math.*min*(*speed*, *MAX\_SPEED*);  
 }  
}

5)

package com.ezidayzi;  
  
*// 1. 객체지향 프로그램의 시작  
// (1) 클래스를 만들고 해당 클래스의 인스턴스를 만들어보세요.*public class SoongsilStudent {  
 public static String *schoolName* = "숭실대학교";  
 private static String *schoolAddress* = "서울시 동작구 상도로 369 숭실대학교";  
 private String name;  
 private Integer age;  
 private Integer studentId;  
 private Boolean graduation;  
  
 public static String getSchoolInfo() {  
 return *schoolName* + " " + *schoolAddress*;  
 }  
  
 public String makeIntroduction() {  
 return "안녕하세요 저는 " + name + " 입니다.";  
 }  
  
 *//Getter* public String getName() {  
 return name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public int getStudentId() {  
 return studentId;  
 }  
  
 public void isGraduate() {  
 if (graduation) {  
 System.*out*.println("졸업생입니다.");  
 } else {  
 System.*out*.println("재학생입니다.");  
 }  
 }  
  
 *// Setter* public void setName(String *name*) {  
 this.name = *name*;  
 }  
  
 public void setAge(int *age*) {  
 this.age = *age*;  
 }  
  
 public void setStudentId(int *studentId*) {  
 this.studentId = *studentId*;  
 }  
  
 public void setGraduation(boolean *graduation*) {  
 this.graduation = *graduation*;  
 }  
}