

Q1 클래스 MyGeneric<T>는 하나의 제네릭 변수를 갖고 있는 제네릭 클래스이며, 클래스 A와 B는 다음과 같다.

```
class MyGeneric <T> {  
    T t;  
    public T get() {  
        return t;  
    }  
    public void set(T t) {  
        this.t = t;  
    }  
}  
  
class A {  
    @Override  
    public String toString() {  
        return "클래스 A 객체";  
    }  
}  
  
class B {  
    @Override  
    public String toString() {  
        return "클래스 B 객체";  
    }  
}
```

이때 set() 메서드의 매개변수로 A, B 객체를 전달받은 후 get() 메서드를 이용해 각각의 타입을 받을 수 있도록 빈칸을 완성하시오.

```
public static void main(String[] args) {  
  
    A a_in = new A();  
    B b_in = new B();  
  
    MyGeneric<A> mg1 = MyGeneric<A> ()  
    mg1.set(a_in);  
    A a_out = mg1.get();  
    System.out.println(a_out); // 클래스 A 객체  
  
    MyGeneric<B> mg2 = MyGeneric<B> ()  
    mg2.set(b_in);  
    B b_out = mg2.get();  
    System.out.println(b_out); // 클래스 B 객체  
}
```

실행 결과

클래스 A 객체
클래스 B 객체

Q2 클래스 MyClass는 제네릭 메서드 myMethod()를 포함하고 있다.

```
class MyClass {  
    public <T> T myMethod(T t) {  
        return t;  
    }  
}
```

myMethod() 메서드의 입력매개변수로 각각 "감사", 3, 5.8을 넘겨 줘 다음의 실행 결과가 나올 수 있도록 코드를 완성하시오(단, 제네릭 타입을 생략하지 말고 표기).

```
public static void main(String[] args) {  
    // MyClass 객체 생성 + 제네릭 메서드 호출  
    MyClass mc = new MyClass();  
    String out1 = mc.<String>myMethod ("감사");  
    Integer out2 = mc.<Integer>myMethod (3);  
    Double out3 = mc.<Double>myMethod (5.8);  
  
    System.out.println(out1);  
    System.out.println(out2);  
    System.out.println(out3);  
}
```

실행 결과

감사
3
5.8

Q3 다음과 같이 클래스 A와 인터페이스 B가 있을 때, MyGenericA는 'A 또는 A의 자식 클래스'만 제네릭 타입으로 지정할 수 있고, MyGenericB는 'B 또는 B의 자식 클래스'만 제네릭 타입으로 지정할 수 있도록 코드를 완성하시오.

```
class A {}  
interface B {}  
  
// A 또는 A의 자식 클래스만 제네릭 타입으로 지정 가능  
class MyGenericA<T extends A> {  
  
}  
  
// B 또는 B의 자식 클래스만 제네릭 타입으로 지정 가능  
class MyGenericB<T extends B> {  
  
}
```

Q4 클래스 A, B, C, D가 다음과 같은 상속 관계를 갖고 있다.

```
class A {}  
class B extends A {}  
class C extends B {}  
class D extends C {}  
  
class MyClass <T extends C> {  
  
}
```

다음 올바른 코드를 표기하시오(O/X로 표기).

```
public static void main(String[] args) {  
  
    MyClass<A> mca = new MyClass<A>();    X      ( O / X )  
    MyClass<B> mcb = new MyClass<B>();    X      ( O / X )  
    MyClass<C> mcc = new MyClass<C>();    O      ( O / X )  
    MyClass<D> mcd = new MyClass<D>();    O      ( O / X )  
  
}
```

Q5 다음과 같은 상속 구조를 갖는 클래스 A, B, C와 제네릭 클래스 MyGeneric<T>가 있다. 또한 클래스 MyClass 내부에는 제네릭 클래스 객체 타입을 매개변수로 갖는 4개의 메서드가 있다.

```
class A {}  
class B extends A {}  
class C extends B {}  
  
class MyGeneric <T> {  
    T t;  
    public void set(T t) {  
        this.t = t;  
    }  
    public T get() {
```

```

        return t;
    }
}

class MyClass {
    void method1(MyGeneric<B> a) {}
    void method2(MyGeneric<?> a) {}
    void method3(MyGeneric<? extends B> a) {}
    void method4(MyGeneric<? super B> a) {}
}

```

다음 중 올바른 코드를 모두 고르시오(O/X로 표기).

```

public static void main(String[] args) {
    MyClass mc = new MyClass();
    mc.method1(new MyGeneric<A>());      X      ( O / X )
    mc.method1(new MyGeneric<B>());      O      ( O / X )
    mc.method1(new MyGeneric<C>());      X      ( O / X )

    mc.method2(new MyGeneric<A>());      O      ( O / X )
    mc.method2(new MyGeneric<B>());      O      ( O / X )
    mc.method2(new MyGeneric<C>());      O      ( O / X )

    mc.method3(new MyGeneric<A>());      X      ( O / X )
    mc.method3(new MyGeneric<B>());      O      ( O / X )
    mc.method3(new MyGeneric<C>());      O      ( O / X )

    mc.method4(new MyGeneric<A>());      O      ( O / X )
    mc.method4(new MyGeneric<B>());      O      ( O / X )
    mc.method4(new MyGeneric<C>());      X      ( O / X )
}

```

Q6 제네릭 클래스 B는 제네릭 클래스 A를 상속받아 추가로 제네릭 변수를 추가한 것이다. 제네릭 클래스 B의 객체를 생성해 메서드를 호출하는 다음 코드의 빈칸을 완성하고 실행 결과를 쓰시오.

```
class A <K> {
    K k;
    public void setK(K k) {
        this.k = k;
    }
}

class B <T,V> extends A<T> {
    V v;
    public void setV(V v) {
        this.v = v;
    }
    public void print() {
        System.out.println(k + ": " + v);
    }
}

public static void main(String[] args) {
    B <Integer, String> b = new B<>();
    b.setK(1);
    b.setV("감사");
    b.print();
}
```

실행 결과

1: 감사