1번답

	변수	설명
상태(속성)	int year	년도
	int month	월
	int day	일

	메소드 이름	설명
동작(행동)	void setDate(int y, int m, int d)	날짜를 설정
	void printDate()	날짜를 출력

2번답

출력결과 : null

자바에서는 모든 것이 call-by-value 방식으로 전달된다. 따라서 위의 프로그램에서는 참조 변수 x의 값이 참조 변수 y로 복사된다. 참조 변수 y의 값을 변경하여도 변수 x의 값은 변경되지 않는다.

3번답:

```
class Television {
   private String model;

   void setModel(String b) { // 설정자
        model = b;
   }

   String getModel() { // void->String
        return model;
   }
}

public class TelevisionTest {
   public static void main(String[] args) {
        Television t = new Television();      // ()을 붙여주어야 함!
        t.setModel("STV-101");
        String b = t.getModel();    // 객체 참조 변수 t를 적어주어야 함.
   }
}
```

4번답

1) 객체을 생성할대 초출하며, 객체가 생성될 때에 필드에게 초기값을 제공하고 필요한 초기화 절차를 실행한다.

(2) 매개변수의 자료형이나 매개변수 개수로 구별되어 호출

(3) 자기 자신을 참조

(4) 정적변수는 하나의 클래스에 하나만 존재하여 그 클래스의 모든 객체들에 의해 공유되지만 인스턴스 변수는 각 인스턴스마다 별도로 생성된다.

(5) 객체의 참조값이 전달된다.

(6) 정적 메소드는 객체가 생성되지 않은 상태에서 호출되는 메소드이므로 객체 안에서 존재하는 인스턴스 변수들은 사용할 수 없다.

5번답

```
(1)
public class Point {
   private int x, y;
   public void Point(int x, int y) {
         x = x;
         y = y;
   }
}
```

생성자 Point()는 값을 반환하지 않는다 따라서 void를 삭제한다.

```
(2)
public class MyMath {
   public int getRandom() {
         return (int)Math.random();
   }
   public double getRandom() {
         return Math.random();
   }
}
```

메소드 오버로딩은 매개 변수의 개수, 매개 변수의 데이터 타입을 다르게 해야하므로 메소드의 반환형이 다르다고 해서 메소드를 중복시킬 수 있는 것은 아니다.

```
(3)
public class MyClass {
   private String getName() {
         return "MyClass";
   }
   public static String getClassName() {
         return getName();
   }
}
```

정적 메소드 getStringName()에서 인스턴스 메소드 getName()을 호출할 수 없다.

6번답

```
public class Ex06 {

   public static void main(String[] args) {
         // TODO Auto-generated method stub
         System.out.println("제목"+"₩t₩t"+"저자"+"₩t"+"출판사"+"₩t"+"분류");
         Book book1 = new Book();
         book1.setTitle("82년생 김지영");
         book1.setAuthor("조남주");
         book1.setPublisher("민읍사");
         book1.setSort("소설");
         book1.print();

         Book book2 = new Book("파리아파트","₩t"+"기욤 뮈소","밝은세상","소설");
         book2.print();

         Book book3 = new Book("자료구조");
         book3.setAuthor("₩t"+"이자료");
         book3.setPublisher("DB출판사");
         book3.setSort("컴퓨터/IT");
         book3.print();
```

```
    }

}
class Book{
    private String title;
    private String author;
    private String publisher;
    private String sort;

    public Book() {

    }
    public Book(String title, String author,String publisher, String sort) {
        this.title = title;
        this.author = author;
        this.publisher = publisher;
        this.sort = sort;
    }
    public Book(String title) {
        this.title = title;
    }

    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public String getSort() {
        return sort;
    }
    public void setSort(String sort) {
        this.sort = sort;
    }
    public void print() {
        System.out.println(title+"₩t"+author+"₩t"+publisher+"₩t"+sort);
    }
}
```

7 번답

```
public class Ex07 {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                BankAccount account= new BankAccount("1000-11111", "홍길동", 100000);
```

```
                BankAccount otherAccount= new BankAccount("1000-22222", "박경미", 100000);
                account.deposit("1000-11111", 10000); // 만원 입금
                account.withdraw("1000-11111", 90000); // 보유량보다 높은 금액 출금
                account.withdraw("1000-11111", 60000);
                account.getBalance("1000-11111"); // 잔액조회
                account.transfer(otherAccount, 10000); // 이체
        }
}

class BankAccount {
        private String accountNumber;
        private String owner;
        private int balance;

        public BankAccount() {

        }

        public BankAccount(String accoutNumber, String owner, int balance) {
                this.accountNumber = accoutNumber;
                this.owner = owner;
                this.balance = balance;
        }

        public void deposit(String accountNumber,  int inputMoney) { // 입금
                if (this.accountNumber.equals(accountNumber)) { // 계좌번호와 이름이 같냐?
같으면 입금!
                        System.out.println("입금 전 잔액 : " + balance);
                        this.balance = this.balance + inputMoney;
                        System.out.println("잔액 후 잔액 : " + balance);
                } else
                        System.out.println("계좌번호가 틀렸습니다.");
        }

        public void withdraw(String accountNumber, int outputMoney) { // 출금
                if (this.accountNumber.equals(accountNumber)) { // 계좌번호와 이름이 같냐?
                        if (balance != 0) {
                                                                                //
잔고가 0원이 아니면(음수는 넣을 일이 없어서 배제)
                                if (outputMoney < balance) {
                                                                        // 출금할 금액이
잔고보다 작냐? 작으면 출금
                                        System.out.println("출금 전 잔액 : " + balance);
                                        balance = balance - outputMoney;
                                        System.out.println("출금액 : " + outputMoney);
                                        System.out.println("잔액 후 잔액 : " + balance);
                                } else
                                        System.out.println("현재 잔액 보다 출금액이 더 큽니다");
                        } else
                                System.out.println("잔액이 0원입니다");
                } else
                        System.out.println("계좌번호가 틀렸습니다.");
        }

        public void getBalance(String accountNumber) { // 잔액조회
                if (this.accountNumber.equals(accountNumber)) {
                        System.out.println("현재 잔액 : " + balance);
                } else
                        System.out.println("계좌번호가 틀렸습니다.");
        }


        public void transfer(BankAccount other, int transMoney) {
                if (this.accountNumber.equals(accountNumber)) {
                        if (balance != 0) {
                                if (transMoney < this.balance) {

                                                        // 맞으면 내 잔고에 마이너스 후 이체!
                                        this.balance -=transMoney;
                                        other.balance+=transMoney;
                                        System.out.println(this.owner+"의 현재 잔액 : " +
balance);
```

```
                                        System.out.println(this.owner+"님의
"+this.accountNumber+"계좌에서" + balance+"이");
                                        System.out.println(other.owner + " 님의 " +
other.accountNumber + " 계좌로 " + transMoney + "만큼 이체 되었습니다.");
                                } else
                                        System.out.println("현재 잔액보다 이체 금액이 더 큽니다.");
                        } else
                                System.out.println("잔액이 0원입니다");
                } else
                        System.out.println("계좌번호나, 이름이 틀렸습니다.");
        }

}
```

8번답

```
class Plane {
        private String company;
        private String model;
        private int max_passenger;
        static int planes=0;

        public String getCompany() {
                return company;
        }

        public void setCompany(String company) {
                this.company = company;
        }

        public String getModel() {
                return model;
        }

        public void setModel(String model) {
                this.model = model;
        }

        public int getMax_passenger() {
                return max_passenger;
        }

        public void setMax_passenger(int max_passenger) {
                this.max_passenger = max_passenger;
        }

        public Plane() {

        }

        public Plane(String company) {
                this.company = company;
        }

        public Plane(String company, String model) {
                this.company = company;
                this.model = model;
        }

        public Plane(String company, String model, int max_passenger) {
                this.company = company;
                this.model = model;
                this.max_passenger = max_passenger;
        }

        public static void setPlanes() {
                planes=planes+1;
        }

        public static int getPlanes() {
                return planes;
```

```
	}

	public void print() {
		System.out.println("회사 : " + company + " 모델명 : " + model + " 최대 수용 승객
: " + max_passenger);
		setPlanes();
	}
}

public class Exam08 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Plane p1 = new Plane();
		p1.setCompany("대한항공");
		p1.setModel("F-16K");
		p1.setMax_passenger(100);
		p1.print();
		Plane p2 = new Plane("아시아나");
		p2.setModel("F-18K");
		p2.setMax_passenger(300);
		p2.print();
		Plane p3 = new Plane("에어부산", "F-15K");
		p3.setMax_passenger(200);
		p3.print();
		Plane p4 = new Plane("진에어", "F-13K", 150);
		p4.print();
		System.out.println("비행기의 갯수 : "+Plane.getPlanes());
	}
}
```