
자바 해답 (클래스~API)

해답

1. static 키워드는 제거합니다.
2. price, name의 초기값도 제거합니다.

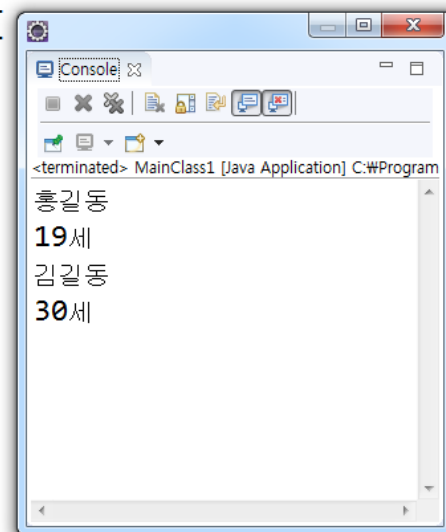
```
public class Bread {  
  
    int price;  
    String name;  
  
    void info() {  
        System.out.println(price + "원");  
        System.out.println(name);  
    }  
}
```

해답

- static 키워드는 제거합니다.
- name, age의 초기값도 제거합니다.
- main에서 2개를 생성해서 값을 지정하고 출력합니다.

```
public class Person {  
  
    String name;  
    int age;  
    void info() {  
        System.out.println(name);  
        System.out.println(age + "세");  
    }  
}
```

```
public class MainClass1 {  
  
    public static void main(String[] args) {  
  
        Person hong = new Person();  
        hong.name = "홍길동";  
        hong.age = 19;  
        hong.info();  
  
        Person kim = new Person();  
        kim.name = "김길동";  
        kim.age = 30;  
        kim.info();  
    }  
}
```



해답

1. 3개의 멤버 변수를 초기화하는 생성자를 만듭니다.
2. 프로그래머가 생성자를 강제로 만든다면, 기본생성자는 없어지기 때문에 기본생성자도 만들어 줍니다.

```
public class Person {  
  
    String name;  
    int age;  
    int tall;  
  
    //기본 생성자  
    Person() {  
  
    }  
    //생성자  
    Person(String pName, int pAge, int pTall) {  
        name = pName;  
        age = pAge;  
        tall = pTall;  
    }  
  
    void info() {  
        System.out.println(name);  
        System.out.println(age + "세");  
        System.out.println("키: " + tall);  
    }  
}
```

해답

```
public class Account {  
  
    //객체의 속성 (멤버변수)  
    String name;  
    String password;  
    int balance;  
  
    //생성자 선언  
    Account(String aName, String pw, int bal) {  
        name = aName;  
        password = pw;  
        balance = bal;  
    }  
  
    //입금 기능  
    void deposit(int money) {  
        balance += money;  
    }  
  
    //출금 기능  
    void withDraw(int money) {  
        balance -= money; //balance = balance - money;  
    }  
  
    //잔액 조회 기능  
    int getBalance() {  
        return balance;  
    }  
}
```

해답

메인화면

```
public class Player {  
    String name;  
    int hp;  
    int mp;  
  
    void info() {  
        System.out.println("캐릭터명:" + name);  
        System.out.println("체력:" + hp);  
        System.out.println("마나:" + mp);  
    }  
}
```

```
public class Wizard extends Player{  
  
    void iceArrow() {  
        System.out.println("얼음화살 사용");  
    }  
}
```

```
public class Warrior extends Player{  
  
    void bash() {  
        System.out.println("때리기 스킬 사용");  
    }  
}
```

```
public class MainClass {  
  
    public static void main(String[] args) {  
        System.out.println("----마법사----");  
  
        Wizard p1 = new Wizard();  
        p1.name = "구르미그린달빛";  
        p1.hp = 500;  
        p1.mp = 1000;  
        p1.info();  
        p1.iceArrow();  
  
        System.out.println("-----전사-----");  
  
        Warrior p2 = new Warrior();  
        p2.name = "강한친구대한전사";  
        p2.hp = 1000;  
        p2.mp = 500;  
        p2.info();  
        p2.bash();  
    }  
}
```

해답

부모클래스

```
public class Player {  
  
    String name;  
    int hp;  
    int mp;  
  
    void info() {  
        System.out.println("캐릭터명:" + name);  
        System.out.println("체력:" + hp);  
        System.out.println("마나:" + mp);  
    }  
  
    void action() {  
  
    }  
}
```

자식클래스

```
public class Wizard extends Player{  
  
    void iceArrow() {  
        System.out.println("얼음화살 사용");  
    }  
  
    void action() {  
        System.out.println("마법사는 귀여운 동작을 합니다.");  
    }  
}
```

자식클래스

```
public class Warrior extends Player{  
  
    void bash() {  
        System.out.println("때리기 스킬 사용");  
    }  
  
    void action() {  
        System.out.println("전사는 멋진 동작을 합니다.");  
    }  
}
```

해답

```
public class Calculator {  
    int result; //누적 시킬 멤버변수  
    int add(int a) {  
        result +=a;  
        return result;  
    }  
    int add(int a, int b) {  
        result += a + b; // result = result + a + b;  
        return result;  
    }  
    int add(int a, int b, int c) {  
        result += a + b + c;  
        return result;  
    }  
    int add(int a, int b, int c, int d) {  
        result += a + b + c + d;  
        return result;  
    }  
}
```


해답

```
public class ArrayPrint {  
    void printArray(String[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
    void printArray(int[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
    void printArray(char[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
}
```

해답

```
public class Wizard extends Player{

    void iceArrow() {
        System.out.println("얼음화살 사용");
    }

    //생성자
    public Wizard(String name) {
        this.name = name;
        this.hp = 500;
        this.mp = 1000;
    }

    void action() {
        System.out.println("마법사는 귀여운 동작을 합니다.");
    }

}
```

```
public class Warrior extends Player{

    void bash() {
        System.out.println("때리기 스킬 사용");
    }

    //생성자
    public Warrior(String name) {
        this.name = name;
        this.hp = 1000;
        this.mp = 500;
    }

    void action() {
        System.out.println("전사는 멋진 동작을 합니다.");
    }

}
```

해답

해답

1. bash스킬의 매개변수로 Player클래스를 받을 수 있도록 선언합니다.
2. Player의 자식 클래스들이 모두 전달 될 수 있습니다.
3. 전달 된 클래스의 hp - 100 하여 저장합니다.

```
public class Warrior extends Player{  
  
    //생성자  
    public Warrior(String name) {  
        this.name = name;  
        this.hp = 1000;  
        this.mp = 500;  
    }  
    //bash스킬  
    void bash(Player player) {  
        System.out.println(player.name + "님 배워 스킬 적중\n");  
        player.hp = player.hp-100;  
    }  
  
    void action() {  
        System.out.println("전사는 멋진 동작을 합니다.");  
    }  
}
```

해답

1. instanceof 를 사용하여 객체의 종류에 따라 hp 감소를 다르게 처리합니다.

```
public class Warrior extends Player{

    public Warrior(String name){
        this.name = name;
        this.hp = 1000;
        this.mp = 500;
    }

    public void bash(Player p) {

        if(this.mp <= 0) {
            System.out.println("스킬을 사용할 수 없습니다");
            return;
        }

        System.out.println(p.name + "스킬적중");
        this.mp -= 100; //나의 mp 100 감소

        if(p instanceof Warrior) {
            p.hp -= 100;
        } else if(p instanceof Wizard) {
            p.hp -= 200;
        }

    }

}
```

해답

```
public class Wizard extends Player {  
    Wizard(String name) {  
        this.name = name;  
        this.hp = 500;  
        this.mp = 1000;  
    }  
    void iceArrow() {  
        System.out.println("얼음화살 사용");  
    }  
    public void blizzard(Player[] players) {  
        System.out.println("-----");  
        System.out.println(this.name + "님의 눈보라 시전!");  
        for(Player p : players) {  
            int damage = (int)(Math.random()*6) + 10;  
            p.hp -= damage;  
  
            System.out.println(p.name + "님이 " + damage + "피해를 입었습니다");  
        }  
    }  
}
```

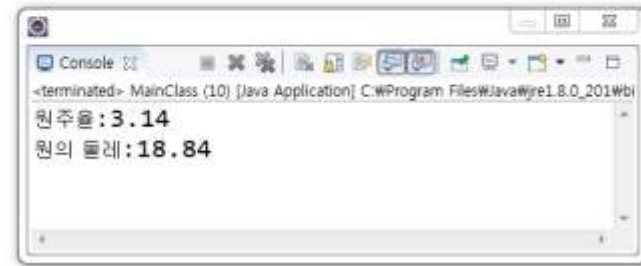
```
public class MainClass {  
    public static void main(String[] args) {  
        Warrior w1 = new Warrior("강한친구");  
        Wizard w2 = new Wizard("구르미");  
        Warrior me = new Warrior("나의캐릭터");  
  
        //w2의 블리자드를 5번 실행시켜주세요  
        Player[] arr = {w1, me};  
  
        w2.blizzard(arr);  
        w2.blizzard(arr);  
        w2.blizzard(arr);  
        w2.blizzard(arr);  
        w2.blizzard(arr);  
    }  
}
```

해답

1. static메서드, 변수 는 객체생성 없이 ★클래스명.이름 으로 바로 사용가능

```
public class Calculator {  
  
    int result; //누적 시킬 멤버변수  
  
    //객체 생성없이 사용, 객체마다 같은 값을 갖는 pi  
    public static double pi = 3.14;  
  
    //객체 생성없이 사용가능한 circle  
    public static double circle(int r) {  
        return 2* r * pi;  
    }  
}
```

```
public class MainClass {  
  
    public static void main(String[] args) {  
  
        System.out.println("원주율:" + Calculator.pi);  
  
        //원의 둘레공식 : 2 * pi * 반지름  
        System.out.println("원의 둘레:" + Calculator.circle(3)); //매개변수 3전달  
  
    }  
}
```



해답

1. static메서드, 변수 는 객체생성 없이 ★클래스명.이름 으로 바로 사용가능

```
public class ArrayPrint {  
    //생성자를 private으로 처리  
    private ArrayPrint() {  
    }  
    //printArray의 오버로딩  
    public static void printArray(String[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
    //printArray의 오버로딩  
    public static void printArray(int[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
    //printArray의 오버로딩  
    public static void printArray(char[] arr) {  
        for(int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println();  
    }  
}
```


해답

```
public class MainClass {  
    public static void main(String[] args) {  
        //nextLine() 을 써서 공백도 입력할 수 있도록하세요.  
        Scanner scan = new Scanner(System.in);  
        while(true) {  
            System.out.print("아이디 등록: ");  
            String id = scan.nextLine();  
            id = id.trim();  
  
            if(id.length() < 5) {  
                System.out.println("아이디는 5글자 이상으로 등록하세요.");  
            }else {  
                System.out.println("아이디: " + id + "가 등록되었습니다.");  
                break;  
            }  
        }  
        scan.close();  
    }  
}
```

해답

```
public class StringQuiz02 {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
  
        System.out.print("주민번호 13자리를 입력하세요: ");  
        String ssn = scan.next();  
        ssn = ssn.replace("-", ""); // "-" 를 모두 공백으로 치환  
  
        if(ssn.length() != 13) { //입력받은 문자열 길이 확인  
            System.out.println("주민번호 13자리를 입력하세요.");  
        } else {  
            switch(ssn.charAt(6)) { //6번째 인덱스 값을 잘라서 확인  
                case '1':  
                    System.out.println("남성");  
                    break;  
                case '2':  
                    System.out.println("여성");  
                    break;  
            }  
        }  
        scan.close();  
    }  
}
```

해답

for문을 사용하는 방법

```

public class Palindrome {

    public static String palindromeCheck(String word) {
        String result = null;

        //if(word.replace(" ", "").length() % 2 == 1) { //문장의 길이가 홀수일 경우만 실행해주는 if문

            word = word.replace(" ", "");
            int mid = word.length() / 2 ;

            String front = word.substring(0, mid);
            String back = word.substring(mid+1, word.length());

            String str1 = "";
            for(int i = front.length()-1 ; i>=0 ; i-- ) { //앞 문자열을 뒤에서 부터 한글자의 길라 다시 붙임.
                str1 += front.charAt(i);
            }

            if(str1.equals(back)) {
                result = "회문입니다.";
            } else {
                result = "회문이 아닙니다.";
            }

        //} else {
        //    result="문장을 홀수로 입력하세요.";
        //}
        return result;
    }
}

```

스트링빌더를 사용하는 방법

```

public class Palindrome {

    public static String palindromeCheck(String word) {
        String result = null;

        //if(word.replace(" ", "").length() % 2 == 1) { //문장의 길이가 홀수일 경우만 실행해주는 if문

            word = word.replace(" ", "");
            int mid = word.length() / 2 ;

            String front = word.substring(0, mid);
            String back = word.substring(mid+1, word.length());

            StringBuilder sb = new StringBuilder(back); //빌더 생성

            if (sb.reverse().toString().equals(front) ) { //sb를 거꾸로 뒤집고 -> 문자열로 변경하고 -> 같은지 확인
                result = "회문입니다.";
            } else {
                result = "회문이 아닙니다.";
            }

        //} else {
        //    result="문장을 홀수로 입력하세요.";
        //}
        return result;
    }
}

```