

Java 모의고사_jw

Day01_기초문법

1. 비트 연산자와 관련된 다음 코드의 출력 값을 작성하세요.

```
int value1 = 20;
int value2 = 30;

System.out.println("#1: " + Integer.toBinaryString(value1));
System.out.println("#2: " + Integer.toBinaryString(value2));
System.out.println("#3: " + Integer.toBinaryString(value1 & value2));

// Q. value1과 value2을 OR 연산해 봅시다.
System.out.println("#4: " + Integer.toBinaryString(value1 | value2));

// Q. value1과 value2을 XOR 연산해 봅시다.
System.out.println("#5: " + Integer.toBinaryString(value1 ^ value2));

// Q. value1과 value2을 NOT 연산해 봅시다.
System.out.println("#6 " + Integer.toBinaryString(~value1));
System.out.println("#7: " + ~value2);

// Q. value1를 왼쪽으로 1칸씩 비트를 옮겨봅시다.
System.out.println("#8: " + Integer.toBinaryString(value1 << 1));

// Q. value1를 오른쪽으로 1칸씩 비트를 옮겨봅시다.
System.out.println("#9: " + (value1 >> 1));
```

- #1. _____
- #2. _____
- #3. _____
- #4. _____
- #5. _____
- #6. _____
- #7. _____
- #8. _____
- #9. _____

2. short-circuit evaluation에 대해 설명하세요.

3. 다음 코드의 출력 값을 작성하고, 해당 결과가 나오는 이유를 설명하세요.

```
// Q. value7의 값과 value8의 값을 생각해 봅시다.
int value7 = 10;
int value8 = 20;
System.out.println("결과 1: " + (value7 += 10) > 15 | (value8 -= 10) > 15 );
System.out.println("#1: " + value 7 + ", " + value8);

// Q. value7의 값과 value8의 값을 생각해 봅시다.
value7 = 10;
value8 = 20;
System.out.println("결과 2: " + (value7 += 10) > 15 || (value8 -= 10) > 15
);
System.out.println("#2: " + value 7 + ", " + value8);
```

#결과1: _____

#1: _____, _____

#결과2: _____

#2: _____, _____

- 이유:

4. 아래 코드에서 break문이 들어가야 하는 개수와, 들어가야 하는 위치를 작성하세요.

```
int month = 3;
int day = -1;

switch (month) {    // 1
    case 2:         // 2
        day = 29;   // 3
    case 4:         // 4
    case 6:         // 5
    case 9:         // 6
    case 11:        // 6
        day = 30;   // 7
    default:        // 8
        day = 31;   // 9
}
```

ex. 정답 예시: 1개, 9번 아래

개수:

위치:

5. JVM의 개념과 장점을 설명하세요.

개념:

장점:

6. 자바 애플리케이션의 시작점으로, 자바 애플리케이션을 실행할 때 가장 먼저 호출하는 것은?

정답:

7. 다음 중 Java에서 쓰이는 주석으로 옳지 않은 것을 모두 고르세요.

정답:

1. // 내용
2. /* 내용 */
3. /** 내용 */
4. /** 내용 **/
5. #내용

8. printf문을 사용하여 출력할 때, 자료형에 맞는 포맷 지정자를 작성하세요.

1. 정수:
2. 실수:
3. 문자:
4. 문자열:

9. 각 설명에 대하여 OX로 답하세요.

1. Java에서 메모리의 기본 단위는 1bit이다. (____)
2. 변수명으로 \$\$\$\$가 가능하다.(____)
3. 변수명으로 12_34가 가능하다.(____)
4. 변수명으로 (언더바 4개)가 가능하다.(____)
5. 변수명으로 @@@@가 가능하다.(____)
6. 변수명을 한글로도 작성할 수 있다.(____)
7. byte는 정수형 데이터 타입이다.(____)

8. int a = 3; byte b = a;는 가능하다.(_____)

9. 비트부정연산자를 사용하여 int a = 3; System.out.println(~a);를 실행하면 -3이 나온다.(_____)

10. 객체 타입 비교 연산자를 적으세요.

정답:

11. 아래 코드는 break, continue문에서 label을 사용하는 방식이다. A와 B에 들어갈 기호를 쓰시오.

```
< 기본 문법 >
EXIT_FOR(____A____): //레이블명
for(int dan = 1; i <= 9; i++) {
    System.out.println(" < " + dan + "단 > ");
    for(int i = 1; i <= 9; i++) {
        if(dan ==5) {
            break EXIT_FOR(____B____)
        } // if문 종료
        System.out.printf("%d * %d = %d\n", dan, i, dan*i);
    } // 안쪽 for문 종료
} // 바깥쪽 for문 종료
```

A: _____

B: _____

Day02_배열

1. 다음 코드의 빈칸을 채우고, 출력을 작성하세요.

```
int[] numberList1 = new int[] {1, 2, 3, 4, 5};
int[] numberList2 = numberList1;

System.out.println("해시코드값 동일 여부: " + (numberList1.____A____ ==
numberList2.____A____));
```

```
numberList1[0] = 100;  
System.out.println("numberList2의 0번 인덱스 값: " + numberList2[0]);
```

해시코드 값을 비교하기 위한 메서드로, A에 공통으로 들어갈 메서드: _____

numberList2의 인덱스 값: _____

2. 배열 안의 요소를 [값1, 값2, ...] 형태로 출력하는 Arrays의 메서드를 작성하세요.

정답: _____

3. API를 이용하여 배열을 복사하려고 합니다. 아래 코드의 빈칸을 작성하세요.

```
int[] numberList1 = new int[] {1, 2, 3, 4, 5};  
int[] numberList2 = new int[5];  
numberList2 = Arrays.copyOf(numberList1, numberList1.length);  
System.out.println("주소 비교: " + (numberList1.hashCode() ==  
numberList2.hashCode()));
```

주소 비교: _____

4. 다음은 첫 줄에 선언된 코드에 이어 추가 작성할 코드들입니다. 1~5번 각 경우에 정상 실행되는지 여부를 OX로 정답을 작성하세요.

```
int[] scores = {90, 80, 100}; // 첫 줄 공통
```

```
// 1번  
scores[3] = 95;
```

```
// 2번  
scores = new int[] {90, 80, 100, 95};
```

```
// 3번
scores = {90, 85, 100, 95};
```

```
// 4번
scores = Arrays.copyOf(scores, 5);
scores[3] = 95;
```

```
// 5번
int scores2 = new int[4];
System.arraycopy(scores, 0, scores2, 0, scores.length);
scores2[3] = 95;
```

1번:

2번:

3번:

4번:

5번:

5. Java에서 배열과 관련된 문장들에 대하여 OX로 참, 거짓을 판별하세요.

1. (____)char 자료형으로 배열을 만들면, 기본값은 \u000이다.
2. (____)새로운 배열을 생성하고 싶을 때, 새로운배열 = Arrays.copyOf() 또는 System.arraycopy()를 사용한다.
3. (____)1차원 배열은 기본형, 다차원배열은 참조형이다.
4. (____) int[][] scores = new int[3][]; 라는 식을 적고 Arrays.toString(scores)를 출력하면 error가 발생한다.

Day03_클래스

1. 객체지향 프로그래밍(OOP)의 정의와 장점에 대해 간단히 설명하세요.

정의:

장점:

2. 현실 세계의 객체를 SW 객체로 설계하는 것을 의미하는 용어를 적어주세요.

정답: _____

3. 객체지향 프로그래밍의 특징(A PIE)의 설명에 해당하는 용어를 작성하세요.

1. _____: 하나의 클래스가 다른 클래스의 속성과 메서드를 물려받는 것
2. _____: 상속 또는 구현 관계에 있을 때 객체들이 서로 다른 방식으로 동작하는 것
3. _____: 객체의 불필요한 세부사항을 숨기고, 필요한 인터페이스만을 제공하여 구현에 의존하지 않는 설계
4. _____: 객체의 데이터와 메서드를 하나로 묶고, 외부로부터 객체의 세부사항을 숨기는 것

4. 클래스를 생성할 때 사용하는 키워드를 작성하세요.

정답: _____

5. 각 설명에 대하여 OX로 답하세요.

1. this는 참조 변수로써 현재 클래스를 가리킨다. (_____)
2. static 메서드 또는 static 블록 내에서는 this를 사용할 수 없다. (_____)

6. 메서드에서 this를 반환하여 메서드 호출을 연쇄적으로 이어나갈 수 있도록 하는 것을 의미하는 용어를 작성하세요.

정답: _____

Day04_객체 배열 관리

1. 아래 코드의 옳고 그름을 OX로 답하고, X이면 이유를 설명하세요.

```
// 공통 코드
package pkg1;
public class Person {
    protected String name;
    protected int age;

    Person() {}
}
```

```
/// 1번
package pkg2;
import pkg1.Person;

public class PersonTest extends Person {
    Person p1 = new Person();
}
```

```
/// 2번
package pkg2;
import pkg1.Person;

public class PersonTest extends Person {
    PersonTest p2 = new PersonTest();
}
```

1번:

2번:

2. OX 퀴즈

1. static 영역에서는 non-static 영역을 직접 접근이 불가능하다. (_____)
2. 다른 패키지에 있는 클래스를 사용하기 위해서는 extends 과정이 필요하다. (_____)
3. import를 선언할 때 해당 패키지의 모든 클래스를 포함하고자 한다면, '@'를 사용하기도 한다. (_____)

Day05_상속

1. 다음 코드를 읽고, 주석의 빈칸에 들어갈 알맞은 용어를 작성하세요.

```
class B {

}

class C {
    A a;
```



```
}

class D {

}

class E {

}

class F {
    void method() {
        System.out.println("f");
    }
}

public class A {
    B b;    // A와 B는 ____ (1) ____ 된다.
    C c;    // A와 C는 ____ (2) ____ 된다.
    D[] d;  // A와 D는 ____ (3) ____ 또는 ____ (4) ____ 관계이다.

    void method() {
        F f = new F(); // A는 F에 ____ (5) ____ 한다.
        f.method();
    }
}
```

- 1.
- 2.
- 3.
- 4.
- 5.

2. 'Java의 모든 클래스는 **(A)** 클래스를 상속한다.' (A)에 들어갈 단어를 쓰세요.

- 정답:

3. OX퀴즈

1. 오버라이딩은 재정의의 의미하며, 상속 관계에서만 사용 가능하다.(_____)
2. 메서드 이름과 반환형이 같으면 오버라이딩이 가능하다. (_____)

3. 오버라이딩을 할 때, 하위 클래스의 접근제어자의 범위가 상위 클래스의 접근제어자 범위보다 작거나 같아야 한다. (____)
4. 클래스 앞에도 final을 선언할 수 있다. (____)
5. final 메서드는 오버로딩을 금지하는 역할을 한다. (____)

Day06_다형성, 추상클래스

1. 다형성에 대한 설명에 대해 OX로 참 거짓을 판별하세요.

1. (____) 실제 하나의 객체를 다양한 타입의 참조로 사용하는 것이 다형성이다.
2. (____) Java에는 연산자 오버로딩이 있다.
3. (____) 상속 관계에 있을 때 조상 클래스 타입으로 자손 클래스의 객체를 참조할 수 있다.
4. (____) 제네릭은 매개변수 다형성의 한 종류이다.
5. (____) 다형성을 이용하면 가독성이 개선된다.
6. (____) 참조변수가 참조하고 있는 인스턴스의 타입을 확인하기 위해서 사용하는 연산자는 instanceof 이다.

2. 아래 코드에서 메모리 상에 Student 객체가 생성된다. (O/X)

```
Object obj = new Student();
```

3. 아래 코드를 실행하면 런타임에러가 발생한다. (O/X)

- Student는 Person의 자식 클래스이다.

```
Student st = (Student) new Person();
```

4. 추상클래스에 대한 설명에 대해 OX로 참 거짓을 판별하세요.

1. (____) class 내에 abstract 키워드를 사용한 메서드가 하나라도 있다면 클래스 선언부에 abstract를 추가해주어야 한다.
2. (____) 클래스 선언부에 abstract를 추가하면, 객체를 생성할 수 없는 클래스를 의미한다.
3. (____) abstract 클래스는 구현의 강제를 통해 프로그램의 안정성을 향상시킨다.
4. (____) 멤버 메서드에는 다형성이 적용되고, 멤버 변수에는 다형성이 적용되지 않는다.
5. (____) abstract가 붙어있는 클래스 안에 추상 메서드는 하나도 존재하지 않을 때, 객체를 생성할 수 있다.

6. (____) abstract가 붙어있는 클래스 안에서 객체 배열을 생성할 수 있다.
7. (____) 상위 클래스 타입으로 자식 클래스를 참조할 수 있다.
8. (____) 일반 클래스를 상속받은 자식 클래스를 추상 클래스로 만들 수 있다.

5. 아래 코드를 보고 출력값을 작성하세요.

```
class Parent {
    String x = "parent x";

    public void method() {
        System.out.println("parent method");
    }

    public static void static_method() {
        System.out.println("parent static method");
    }
}

class Child extends Parent {
    String x = "child x";

    public void method() {
        System.out.println("child method");
    }

    public static void static_method() {
        System.out.println("child static method");
    }
}

public class BindingTest {
    public static void main(String[] args) {
        Parent p = new Child();
        Child c = new Child();

        System.out.println(p.x);
        System.out.println(c.x);

        p.method();
        c.method();

        p.static_method();
        c.static_method();
    }
}
```

정답: (총 6줄)

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

6. 상속 관계에서 메서드가 중복될 때(오버라이딩), 참조 변수의 타입에 상관없이 항상 실제 인스턴스의 자식 클래스 메서드가 호출되는 것을 뜻하는 용어는?

정답:

7. 아래 코드에서 잘못된 부분을 모두 찾아 쓰세요.

```
class Animal {  
    private String name;  
    private int age;  
  
    public Animal(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    public void eat() {  
        System.out.println("밥을 먹자!");  
    }  
  
    public abstract void speak()  
}
```

정답:

8. 다형성에 대해 한 문장으로 설명하세요.

Day07_인터페이스, 제네릭

1. 인터페이스에 대한 설명에 대해 OX로 참 거짓을 판별하세요.

1. (____) 인터페이스 내에 선언된 메서드는 public static final이 기본으로 생략되어 있다.
2. (____) 인터페이스 내에 정의된 변수는 자동으로 private static final로 간주되며, 생략할 수 있다.

3. (____) 클래스가 인터페이스를 구현할 때 사용하는 키워드는 **implement** 이다.
4. (____) 클래스는 하나의 인터페이스만 구현 가능하다.
5. (____) 인터페이스는 다중 상속이 가능하다.

2. 인터페이스 구현부가 있는 메서드를 작성하기 위해 메서드 앞에 **(A)** 키워드를 작성해야 한다. (A)에 들어갈 키워드는?

정답:

3. 인터페이스 내에 선언된 **(A)** 메서드는 인터페이스 이름으로 메서드에 접근하여 사용하며, **(A)** 메서드는 인터페이스를 구현한 클래스에서 상속되거나 재정의할 수 없다. (A)에 들어갈 키워드는?

정답:

Day08_컬렉션 프레임워크

1. 다음 빈칸에 들어갈 알맞은 인터페이스를 작성하세요.

1. **(A)** : 중복을 허용하지 않는 컬렉션
2. **(B)** : First In First Out 방식의 컬렉션
3. **(C)** : 양쪽 끝에서 요소의 삽입/삭제가 가능한 컬렉션
4. **(D)** : 위 A,B,C등의 컬렉션의 상위 인터페이스
5. **(E)** : 중복을 허용하고 순서가 있는 컬렉션

2. 다음 빈칸에 들어갈 알맞은 유틸리티 클래스를 작성하세요.

1. **(A)** : 컬렉션 객체의 정렬, 검색 등을 위한 정적 메서드를 제공하는 유틸리티 클래스
2. **(B)** : 배열을 다루기 위한 유틸리티 메서드를 제공하는 유틸리티 클래스

Day09_예외처리

1. 런타임 에러에 대해 설명하세요.

2. 다음 빈칸에 들어갈 알맞은 용어를 쓰세요.

1. **(A)** : 실행 시점에서 발생하는 예외로, 컴파일러가 예외처리를 강제하지 않음
2. **(B)** : 프로그래머가 적절한 코드를 통해 대비할 수 있는 오류
3. **(C)** : 발생하면 복구할 수 없는 심각한 오류
4. **(D)** : 예외를 명시적으로 발생시키는 키워드
5. **(E)** : 예외 직접 처리 키워드에서 예외 발생 여부와 상관없이 항상 실행되는 코드 블록의 키워드
6. **(F)** : Exception 인스턴스의 주요 메서드로, 예외 발생 당시의 호출 스택을 출력하는 메서드
7. **(G)** : Exception 인스턴스의 주요 메서드로, 발생한 예외에 대한 구체적인 메시지를 반환하는 메서드

정답

1. (A) :
2. (B) :
3. (C) :
4. (D) :
5. (E) :
6. (F) :
7. (G):