**시험문제 해답**

**이름 :**

**1. 다음 알맞은 괄호에 답하세요.**

① 자바에서는 반드시( new )연산자를 사용하여 객체를 생성한다.

② 슈퍼 클래스의 접근제한자( protected )는 패키지 소속과 상관없이 서브 클래스에서

접근이 가능하며 동일한 패키지 내의 클래스에서도 접근이 가능하다.

③ 서브 클래스 객체는 슈퍼 클래스 타입으로 자동 타입 변환이 가능하며 이를( 업캐스팅 )

이라고 하며, 다시 원래의 타입으로 강제 타입 변환하는 것을 다운캐스팅이라고 한다.

④ 슈퍼 클래스에 선언된 메소드를 서브 클래스에서 재정의하는 것을 ( 메소드 오버라이딩 )

이라고 한다.

⑤ 추상 메소드는 실행 코드 없이 원형만 선언된 미완성 메소드이며, 반드시 메소드 이름 앞에

( abstract)를 붙여 선언하여야 한다.

⑥ 인터페이스를 구현할 때 ( implements ) 키워드를 사용하며, 인터페이스의 모든 메소드를

구현한다.

⑦ 추상 클래스와 인터페이스의 ( 객체 )는 생성 할 수 없다.

⑧ 자바에서 인터페이스는 추상 메소드와 ( final 상수 )만을 포함하며, ( interface ) 키워드를

사용하여 선언한다.

**2. 자바의 클래스와 객체에 대한 설명 중 틀린 것은? 4**

① 객체를 인스턴스라고도 부른다.

② 클래스는 객체를 생성하기 위한 틀이다.

③ 클래스는 필드와 메소드로 구성된다.

④ 클래스의 필드들은 private보다는 public으로 선언하는 것이 바람직하다.

**3. 인스턴스 멤버와 정적 멤버에 대한 설명으로 틀린 것은 무엇입니까? 2**

1. 정적 멤버는 static으로 선언된 필드와 메소드를 말한다.
2. 인스턴스 필드는 생성자 및 정적블록에서 초기화 될 수 있다.
3. 정적 필드와 정적 메소드는 객체 생성 없이 클래스를 통해 접근할 수 있다.
4. 인스턴스 필드와 메소드는 객체를 생성하고 사용해야 한다.

**4. 접근 제한자에 대한 설명으로 틀린 것은 무엇입니까? 3**

1. 접근제한자는 클래스, 필드, 생성자, 메소드의 사용을 제한한다.
2. public 접근 제한은 아무런 제한없이 해당 요소를 사용할 수 있게 한다.
3. public 접근 제한은 해당 클래스 내부에서만 사용을 허가한다.
4. 외부에서 접근하지 못하도록 하려면 private 접근 제한을 해야한다.

**5. 필드, 생성자, 메소드에 대한 설명으로 틀린 것은 무엇입니까? 4**

1. 필드는 객체의 데이터를 저장한다.
2. 생성자는 객체의 초기화를 담당한다.
3. 메소드는 객체의 동작부분으로, 실행 코드를 가지고 이는 블록이다.
4. 클래스는 반드시 필드와 메소드를 가져야 한다.

**6. 다음 중 생성자에 대한 설명이 옳지 않은 것은? 4**

1. 모든 생성자의 이름은 클래스의 이름과 동일해야 한다.
2. 클래스에는 생성자가 반드시 하나 이상 있어야 한다.
3. 생성자가 없는 클래스는 컴파일러가 기본 생성자를 추가 한다.
4. 생성자는 오버로딩 할 수 없다.

**7. 생성자에 대한 설명 중 잘못된 것은? 3**

1. 한 클래스에 여러 개의 생성자가 있어도 된다.
2. this()는 생성자에서 다른 생성자를 호출하는 코드이다.
3. 생성자에서는 this 래퍼런스를 사용할 수 없다.
4. 생성자는 객체 당 오직 한번만 호출 된다.

**8. 메소드 오버라이딩에 대한 설명으로 틀린것은? 2**

1. 상속관계에서 발생한다.
2. 메소드의 이름이 동일할 필요는 없다.
3. 슈퍼클래스의 메소드를 서브클래스에서 재작성하는 것이다.
4. 서브클래스에서 슈퍼클래스의 메소드 이름, 매개변수의 개수와 타입, 리턴타입을 동일하게

선언해야 성공한다.

**9. 인터페이스에 대한 설명으로 틀린 것은 무엇입니까? 3**

① 인터페이스는 객체 사용 설명서 역할을 한다.

② 구현 클래스가 인터페이스의 추상 메소드에 대한 실체 메소드를 가지고 있지 않으면

추상 클래스가 된다.

③ 인터페이스는 인스턴스 필드를 가질 수 있다.

1. 구현 객체는 인터페이스 타입으로 자동 변환된다.

**10. 다음 중 인터페이스의 특징이 아닌 것은? 3**

1. 인터페이스의 객체를 생성할 수 없다.
2. 여러 인터페이스를 상속받는 다중 상속을 지원한다.
3. 인터페이스는 추상 메소드, 상수, 필드로 구성된다.
4. 인터페이스는 implements 키워드를 이용하여 구현한다.

**11. 다음 클래스에서 해당 멤버가 필드, 생성자, 메소드 중 어떤 것인지 빈칸을 채우세요.**

|  |
| --- |
| public class Member{  private String name; *(① 필드 )*  public Member(String name) { ...... } *(② 생성자 )*  public void setName(String name) { ..... } *(③ 메소드 )*  } |

**12. 다음 에러를 찾으세요. 6줄**

|  |  |
| --- | --- |
| 1  2  3  4  5  *6*  7 | class Number{  public static int a, b;  public int getA(){ return a; }  public int getB(){ return b; }  public int getC(){ return getA(); }  *public static int getD(){ return getA(); }*  } |

**13. 다음 코드에서 잘못된 문장을 찾으세요. 9줄**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  *9*  10  11  12 | class Number{  public int x;  public static int y;  public static int fun(){  return y;  }}  public class A {  public static void main(String[] args) {  *Number.x=10;*  Number.y=20;  int imsi=Number.fun();  }} |

**14. 다음의 코드에 정의된 변수들을 종류별로 구분해서 적으세요.**

|  |
| --- |
| class Card{  public int kind, num;  public static int width, height;  public void getCard(int k, int n){  kind=k;  num=n  }} |

***① static변수 : width, height ②인스턴스변수(멤버변수) : kind, num ③지역변수 : k, n***

**15. 다음과 같은 멤버 변수를 갖는 Sutdent 클래스를 정의 하세요**

|  |  |  |
| --- | --- | --- |
| 타입 | 필드이름 | 설명 |
| 문자열 | name | 학생이름 |
| 정수 | ban | 반 |
| 정수 | no | 번호 |
| 정수 | kor | 국어점수 |
| 정수 | eng | 영어점수 |
| 정수 | math | 수학점수 |

***[프로그램작성]***

class Student{

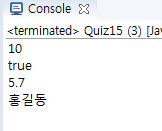
public String name;

public int ban, no, kor, eng, mat;

}

**16. 다음 메인 함수를 보고 클래스를 작성하세요.**

|  |
| --- |
| public class Test16 {  public static void main(String[] args) {  Printer printer=new Printer();  printer.println(10);  printer.println(true);  printer.println(5.7);  printer.println("홍길동");  }  } |



***[프로그램 작성]***

class Printer{

public void println(int x){ System.out.println(x); }

public void println(boolean x){ System.out.println(x); }

public void println(double x){ System.out.println(x); }

public void println(String x){ System.out.println(x); }

}

**17. 다음 main() 보시고 클래스를 작성하세요.**

|  |
| --- |
| public class Test17 {  public static void main(String[] args){  Test t=new Test(10);  Test e=new Test('A');  }} |

***[프로그램 작성]***

class Test{

private int x;

private char y;

public Test(int x){ this.x=x; }

public Test(char y){ this.y=y; }

}

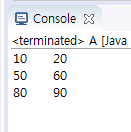
**18. 결과화면을 보시고 프로그램을 완성하세요.**

|  |
| --- |
| class HakJum{  private int jumsu;  private char grade;  ***[생성자 작성]***  public HakJum(int jumsu){  this.jumsu=jumsu;  }  ***[getHakjumg 함수 작성]***  public void getHakjum(){  if(jumsu >= 90)  grade='A';  else if(jumsu >=80)  grade='B';  else if(jumsu >=70)  grade='C';  else if(jumsu >=60)  grade='D';  else if(jumsu < 60)  grade='F';  }  public void disp(){  System.out.println("jumsu:" + jumsu + "\t" + "grade:" + grade);  }  }  public class Test18 {  public static void main(String[] args) {  HakJum h=new HakJum(88);  h.getHakjum();  h.disp();  }} |

**19. 다음 선언을 보시고 배열을 완성하세요**

|  |
| --- |
| class Test{  private int x, y;  public Test(int x, int y){  this.x=x;  this.y=y;  }  public void disp(){  System.out.println(x + "\t" + y);  }}  public class Quiz {  public static void main(String[] ar){  Test[] array=new Test[3];  ***[프로그램작성]***  *array[0]=new Test(10,20);*  *array[1]=new Test(50,60);*  *array[2]=new Test(80,90);*  for(inti=0;i<array.length;i++){  array[i].disp();  }  }} |

[결과화면]



**20. 다음 추상클래스 Study로 부터 상속받아 Study\_Sub 클래스를 정의 하세요**

|  |
| --- |
| abstract class Study{ public abstract void printMsg(); }  ***[프로그램작성]***  class Study\_Sub extends Study{  public void printMsg(){  System.out.println("Java는 참 재미있습니다.");  }  }  public class ABC {  public static void main(String[] args){  Study s=new Study\_Sub();  s.printMsg();  }  } |

[결과 화면] Java는 참 재미있습니다.

**21. 다음의 그림을 보시고 추상 클래스를 선언 하세요.**

abstract class Message{

protected String title;

protected String name;

public abstract void send(String title, String name);

}

**22. 다음과 같은 클래스를 선언해서 상속 하였다. 각각 해당 변수들의 값은?**

|  |
| --- |
| class AA{ protected int x=10; }  class BB extends AA{  public void disp(){  System.out.println(x + "," + this.x + "\t" + super.x);  }} |

***[결과값] x:10 this.x:10 super.x:10***

**23. 다음 결과 값을 보시고 ①번~③번까지 적절한 프로그램을 완성하세요.**

|  |
| --- |
| interface Soundable{ public String sound(); }  class Cat implements Soundable{  @Override  public String sound() { return "야옹"; }  }  class Dog implements Soundable{  @Override  public String sound() { return "멍멍"; }  }  public class Quiz23 {  public static void main(String[] args) {  printSound(① new Cat() );  printSound(② new Dog() );  }  private static void printSound(③ Soundable soundable){  System.out.println(soundable.sound());  }  } |

**24. 메인화면을 보시고 프로그램을 완성하세요.**

|  |
| --- |
| class CDInfo{  protected int managerNum;  public int getManagerNum() {  return managerNum;  }  public void setManagerNum(int managerNum) {  this.managerNum = managerNum;  }  }  class Music extends CDInfo{  protected String artist;  protected String[] song;  ***[프로그램 작성]***  public void setArtist(String artist) {  this.artist = artist;  }  public void setSong(String[] song) {  this.song = song;  }  public void disp(){  System.out.println("관리자 번호:" + this.getManagerNum());  System.out.println("가수:" + artist);  for(int i=0;i<song.length;i++){  System.out.println("힌트송[" + (i+1) + "]:"+ song[i]);  }  }}  public class Quiz30 {  public static void main(String[] args){  Music m=new Music();  m.setManagerNum(123456);  m.setArtist("플라이 투 더 스카이");  String[] song={"너를너를너를", "전화하지 말아요"};  m.setSong(song);    m.disp();  }  } |

**25. 다음 프로그램을 보시고 클래스 다이어그램을 작성하세요.**

|  |
| --- |
| class Person{  protected String name;  protected int kor, eng, math, tot;  protected float avg;  public void setData(String name, int kor, int eng, int math){  this.name=name;  this.kor=kor;  this.eng=eng;  this.math=math;  }  }  class Sungjuk extends Person{  public void total(){  tot=kor+eng+math;  }  public void average(){  avg=(float) tot/3;  }  public void disp(){  System.out.println(name + "\t" + kor + "\t" + eng + "\t"+ math  + "\t" + tot + "\t" + avg);  }  } |

