**[연습문제]**

1. 다음 중 틀린 것을 모두 골라라.

(1) Java는 C++과 마찬가지로 포인터를 활용할 수 있다.

(2) Java는 다중 상속을 지원한다.

(3) Java는 연산자 오버로딩을 지원한다.

(4) Java는 한번 작성하면 여러 운영체제에서 소스를 그대로 사용할 수 있다.

2. Java에서 사용되는 데이터 형식에 중에서 '정수형'에 대해서 정리해 보자.

Short Int Long

3. 다음은 점수에 따라서 A~F까지 학점을 나뉘는 프로그램이다. 괄호 부분을 채워라.

|  |
| --- |
| public static void main(String args[]) {  int input = 85;  int jumsu = (input / 10) \* 10;   1. switch (jumsu) {   case 100:  case 90: System.out.println("A 학점"); break;  case 80: System.out.println("B 학점"); break;  case 70: System.out.println("C 학점"); break;  case 60: System.out.println("D 학점"); break;  (2) default : System.out.println(“F 학점”); break;  }  } |

4. 다음과 같이 3x3 이차원 배열을 선언하고, for문을 이용해서 값을 대입하는 코드를 작성하라.

힌트) 2중 for문을 사용한다.



Int[][] arr= new int[2][2];

Int num=90;

For(int i=0; i<2;i++){

For(int j=0;j<2;j++){

Arr[i][j]=num+10;

}

}

5. 다음 코드가 출력하는 값을 차례로 써라.

|  |
| --- |
| static int data = 100;  public static void main(String args[]) {  int data = 200;  data ++;  System.out.println(data);  func1(data);  }  static void func1(int data) {  data ++;  System.out.println(data);  } |

201,202

6. 다음은 객체지향과 관련된 내용이다. 틀린 것을 모두 고르고 틀린 이유를 설명하라.

(1) 클래스를 정의할 때, 생성자는 반드시 필요한다.

(2) 메소드 오버로딩(Overloading)은 같은 클래스 내에서 메소드의 이름이 같아도 파라미터의 개수가 다르면 같은 이름의 메소드를 여러 개 선언할 수 있는 것을 말한다.

(3) 정적 필드(Static field)는 앞에 final 키워드를 붙여야 한다.

(4) 정적 메소드(Static Method)는 클래스 변수의 선언이 없이도 사용이 가능하다.

(5) Java는 클래스의 다중 상속이 가능하지만 되도록 사용하지 않는 것이 좋다.

(6) 추상 클래스는 클래스 이름 앞에 abstract를 붙여서 정의한다.

(7) 추상 메소드를 포함하는 클래스는 반드시 추상 클래스로 정의해야 한다.

(8) 추상 메소드도 필요하다면 메소드 본체를 만들고 메소드에서 작동할 내용을 코딩할 수 있다.

(9) 인터페이스에는 일반 변수를 선언할 수는 없지만, 정적 변수는 선언이 가능하다.

(10) 내부 클래스(Inner Class)란 클래스 안에 클래스를 정의하는 것을 말한다.

7. [예제 3-7]의 Car 클래스를 상속받은 Truck 클래스를 다음 조건에 맞춰서 생성하자.

- 배기량 변수 cc를 추가한다 (정수형)

- 트럭의 생산년도 상수 필드 YEAR를 선언하고, 값을 2013으로 한다.

- 배기량 값을 파라미터로 받는 생성자를 추가한다.

- 배기량을 반환하는 getCC() 함수를 추가한다.

- upSpeed(int) 메소드를 최대 속도 150으로 오버라이딩(Overriding)한다.

Public class Truck extends Car{

Int cc;

Static final int YEAR = 2013;

Public Truck(int cc){

This.cc=cc;

}

Public int getCC(){

Return cc;

}

@Override

Public void upSpeed(int){  
 this.speed=150;

}

}

8. 다음 코드는 컴파일 오류는 없으나, 실행시에 오류를 발생된다. 이를 제네릭스로 수정해서 컴파일 시에 오류가 발생하도록 하라.

|  |
| --- |
| public static void main(String args[]) {  ArrayList<Integer> numList = new ArrayList<>();  numList.add(1);  numList.add(2);  numList.add("3");  int hap=0;  for(int i=0; i< numList.size(); i++) {  hap += (int) numList.get(i);  }  System.out.print( hap );  } |

9. 다음은 클래스와 제너릭스를 활용해서, 10+20+30의 합을 구하는 코드다. 빈 칸을 완성해라.

|  |
| --- |
| public static void main(String args[]) {  myClass c1 = new myClass(10);  myClass c2 = new myClass(20);  myClass c3 = new myClass(30);    (1) ArrayList<Integer> myList = new ArrayList<>();  myList.add(c1);  myList.add(c2);  myList.add(c3);  int hap=0;  for(int i=0; i< myList.size(); i++) {  (2) hap +=(int)myList.get(i).getValue; ;  }  System.out.print( hap );  }    static class myClass {  int value;  myClass (int value) {  this.value = value;  }  } |