|  |  |
| --- | --- |
| 1 | 다음 설명 중 맞는 것은?읍다  class StringTest{  public static void main(String[] args){  String s1=new String("korea"); (A)  String s2="korea"; (B)  if(s1==s2){ (C)  System.out.println("같습니다.");  }else{  System.out.println("다릅니다.");  }  }  }  (1) String 은 클래스가 아니다.    (2) String 클래스를 사용하려면 String 클래스가 위치한 package 를 import 해야 한다.    (3) String 이 객체이긴 하지만 (A)와 같이 new 연산자에 의해 인스턴스를 올리는 것은 불가능하다    (4) (C)는 s1과 s2의 내용을 비교하고 있다.    (5) 위 프로그램의 실행 결과는 "같습니다" 가 출력된다. |
| 2 | 다음 설명 중 틀린 것은? 2  (1) String 은 그 내용이 수정될 수 없는 불변(immutable)의 특성이 있다.    (2) String 클래스의 인스턴스를 메모리에 생성하는 방법에는 explicit 생성법과 implicit 생성법이 있는데,  explicit 생성법을 사용하면 메모리의 효율성이 높다.  implicit 생성법이 더 효율성이 높고 많이 사용한다  (3) String 클래스의 암시적(묵시적) 생성법을 제공하는 이유는 프로그램 개발시 문자열의 사용 빈도가 너무 높기 때문이며, 암시적 방법을 이용하면 마치 기본 자료형 데이터를 생성하는 것 처럼 String 객체를 다룰 수 있다    (4) 자바언어에서 객체의 내용을 비교하려면 == 연산자로는 불가능하며 equals() 메서드를 사용해야 한다.  ->==연산자는 주소값을 비교함  (5) String s=new String("java"); 클래스를 new 연산자에 의해 메모리에 올릴 경우 "java" 가 올라가는 메모리 영역은 heap 이다.  ->명시적 생성법을 사용했으므로 heap에 올라가는 것이 맞다 |
| 3 | 아래의 String 클래스를 대상으로 과제를 수행하세요.    String str="korea java";    (1) 문자열에서 4번째 위치한 'e' 문자만 추출하여 출력하세요.  System.out.println(str.charAt(3));  (2) 문장 중 첫번째 a의 index (순번)를 출력하세요.  System.out.println(str.indexOf('a'));  (3) 문장 중 마지막 a의 index 를 출력하세요  System.out.println(str.indexOf(str.charAt(8)));  System.out.println(str.lastIndexOf(‘a’));  (4) 문자열의 총 길이를 출력하세요.  System.out.println(str.length());  (5) korea 를 "코리아"로 대체하여 "코리아 java"를 출력하세요.  String str2="코리아";  System.out.println(str.replace(str.subSequence(0,5),str2.subSequence(0,3)));  (6) 3번째부터 8번째 문자열까지 추출하여 출력하세요.  System.out.println(str.subSequence(2,7));  (7) 대문자로 변환하여 "KOREA JAVA" 를 출력하세요.  System.out.println(str.replace("korea java","KOREA JAVA"));  System.out.println(toUpperCase()); |
| 4 | photo.jpg 라는 문자열을 대상으로 아래의 과제를 수행하세요.    (1) 파일명만 출력  System.out.println(result[0]);    (2) 확장자만 출력  System.out.println(result[1]); |
| 5 | 다음 빈 칸에 적절한 단어를 채우세요.  자바에서는 기본 자료형간 , 형변환을 지원하고 있으며 객체 자료형도 자료형이므로 객체 자료형 간의 형변환도 지원하고 있다. 하지만 기본 자료형과 객체 자료형 간에도 형 변환을 지원하기 위한 클래스들을  제공해주는데, 이러한 목적의 클래스를 가리켜 ( )라 하며 , 이름은 기본 자료형 명칭을 따라간다. |
| 6 | 적절한 클래스 명칭을 작성하세요.   |  |  | | --- | --- | | **설명** | **클래스명** | | byte 형과 객체 자료형간 형변환을 지원하는 클래스 |  | | short 형과 객체 자료형간 형변환을 지원하는 클래스 |  | | int 형과 객체 자료형간 형변환을 지원하는 클래스 |  | | long 형과 객체 자료형간 형변환을 지원하는 클래스 |  | | float 형과 객체 자료형간 형변환을 지원하는 클래스 |  | | double 형과 객체 자료형간 형변환을 지원하는 클래스 |  | |