|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2019\_1\_java1\_10** | **학번 :** |  | **이름 :** |  |

* **내용 점검**

|  |
| --- |
| 1. 이차원 정형 배열 |
| **public** **class** Exam01 {  **public** **static** **void** main(String[] args) {  **boolean**[][] board = **new** **boolean**[10][10]; //이차원 배열 선언 & 생성  **for** (**int** i = 0; i < 10; i++)  **for** (**int** j = 0; j < 10; j++)  **if**( Math.*random*() < 0.3 )  board[i][j] = **true**; //난수로 초기화  **for** (**int** i = 0; i < 10; i++) {  **for** (**int** j = 0; j < 10; j++)  **if** (board[i][j])  System.***out***.print("# ");  **else**  System.***out***.print(". ");  System.***out***.println();  }  }  } |
| **[실행 결과]** |

|  |
| --- |
| 1. 2차원 비정형 배열 |
| **public** **class** RagArray {  **public** **static** **void** main(String[] args) {  **int** [][] ragged=**new** **int**[5][]; //비정형 배열 선언, 행 크기만 제시  **for**(**int** i=0; i<ragged.length;i++){  ragged[i]=**new** **int**[i+1]; //행별 열크기  }  **for**(**int** i=0; i<ragged.length;i++){ //배열명.length : 행크기  **for**(**int** j=0; j<ragged[i].length;j++) //배열명[행첨자].length : 열크기  ragged[i][j]=j; //초기화  }  **for**(**int**[] row : ragged){  **for**(**int** data : row)  System.***out***.prin(data+ “ “);  System.***out***.println();  }  } |
| **[실행 결과]** |

|  |
| --- |
| 1. 이차원 배열 선언 & 생성 & 초기호 |
| **public** **class** Exam03 {  **public** **static** **void** main(String[] args) {  **int** ary[][] = {{2, 3, 4}, {3, 6, 9}, {4, 5, 7}}; //이차원 배열 선언 & 생성 & 초기화    System.*out*.println("행 크기 : " + ary.length);  System.*out*.println("첫 번째 행의 열 크기 : " + ary[0].length );  //배열 원소출력:for~each  **for** (**int** dim[] : ary) {  **for** (**int** value : dim)  System.*out*.print(value + " ");  System.*out*.println();  }  System.*out*.println("===================");  //배열 원소 출력 : 중첩 for  **for** (**int** i=0; i<ary.length; i++) {  **for** (**int** j=0; j<ary[i].length; j++)  System.*out*.print(ary[i][j] + " ");  System.*out*.println();  }  }  } |
| **[실행 결과]** |

|  |
| --- |
| 1. 오름 차순으로 정렬 |
| **public** **class** Exam04 {  **public** **static** **void** main(String[] args) {  **int** ary[] = {2, 3, 41,13, 66, 19, 4, 95, 37};  **int** index, temp;    **for**(**int** i=0; i<ary.length-1;i++) {  index=i;  **for**(**int** j=i+1; j<ary.length;j++) {  **if**(ary[index] > ary[j])  index=j; //가장 작은 값을 갖는 배열 위치를 index에 저장  }  temp = ary[i];  ary[i]=ary[index];  ary[index] = temp;  }    **for**(**int** data:ary)  System.***out***.print(data+ " ");  }  } |
| **[실행 결과]** |

* **내용 점검**

1. 다음 중에서 올바른 배열 선언을 모두 선택하시오.( )
2. int [] myScores;
3. char [] myChars;
4. int [6] myScores;
5. Dog myDogs [];
6. Dog myDogs [7];
7. 다음 중에서 잘못된 배열 생성 문장을 모두 선택하시오.( )
8. float[ ] f = new float(3);
9. float f2[ ] = new float[ ];
10. float[ ]f1 = new float[3];
11. float f3[ ] = new float[3];
12. float f5[ ] = {1.0f, 2.0f, 2.0f};
13. 다음 문장 중에서 컴파일 오류가 발생하는 것은?( )
14. int[ ] scores = {1, 2, 3};
15. int [ ][ ] scores = {1, 2, 3}, {4, 5, 6};
16. String cats[ ] = {"abc", "def", "hij"};
17. boolean results[ ] = new boolean [] {true, false, true};
18. 다음과 같은 문제를 해결하는 문장을 작성하시오.
19. 30개의 원소를 가지는 int형 배열 studentNumbers 을 선언하고 생성

**[풀이]**

1. 1.2, 3.3, 6.7, 4.5, 1.2 원소로 초기화 되는 double형 배열 values

**[풀이]**

1. 크기가 7인 double 형 배열 array를 입력한 값으로 초기화. 단, 입력 객체는 in

**[풀이]**

1. 배열 array의 크기를 출력

**[풀이]**

1. 다음의 배열 선언에서 잘못된 부분을 수정하시오.
2. int[] numbers = new numbers[100];

**[수정]**

1. double rainfalls[100];

**[수정]**

1. char[] ch; //배열 선언 후

ch={‘d’, ‘t’, ‘s’, ‘t’, ‘a’, ‘y’}; //초기화

**[수정]**

1. int[] employees = new int[5];와 같은 배열 선언이 있다고 하자.
2. 유효한 인덱스의 범위는?

**[풀이]**

(2) 만약 employees[5]와 같이 원소를 접근하였다면 어떤 일이 발생하는가?

**[풀이]**

1. 다음 코드는 배열 a를 배열 b로 복사하려는 의도로 작성되었다. 실제 배열이 복사되는가? 만약 올바르지 않다면 올바르게 복사되도록 코드를 수정하라.

int[] a = { 1, 2, 3, 4, 5 };

int[] b = new int[5];

b = a; // 배열 a를 배열 b로 복사

**[수정]**

1. {1.0, 2.0, 3.0, 4.0} 과 같은 초기값을 가지는 double형의 배열을 생성한다. for-each 반복 루프를 이용하여 모든 배열 요소를 출력하는 프로그램을 작성하라.

**[프로그램 소스]**

double[] dim={{1.0, 2.0, 3.0, 4.0}

for(double d : dim)

System.out.println(d);

* **프로그램 과제**

1. 다음과 같은 행렬 합을 계산하여 출력하는 프로그램을 작성하시오.

힌트) sum[i][j]=arr1[i][j]+arr2[i][j];

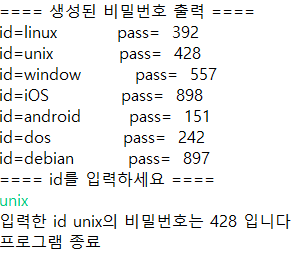
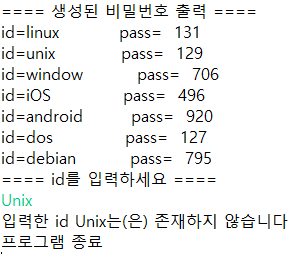
DRW00001b586479



|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

1. 다음과 같이 id를 배열에 저장하고 각각의 id에 대한 비밀번호는 3자리 정수를 생성하여 배열에 저장한다. 제시된 결과처럼 입력된 id에 대한 password를 검색하여 출력하는 프로그램을 작성하시오. 단, id와 password는 서로 동일한 index를 사용한다.

String[] id={“linux”, “unix”, “window”, “iOS”, “android”, “dos”, “debian”};

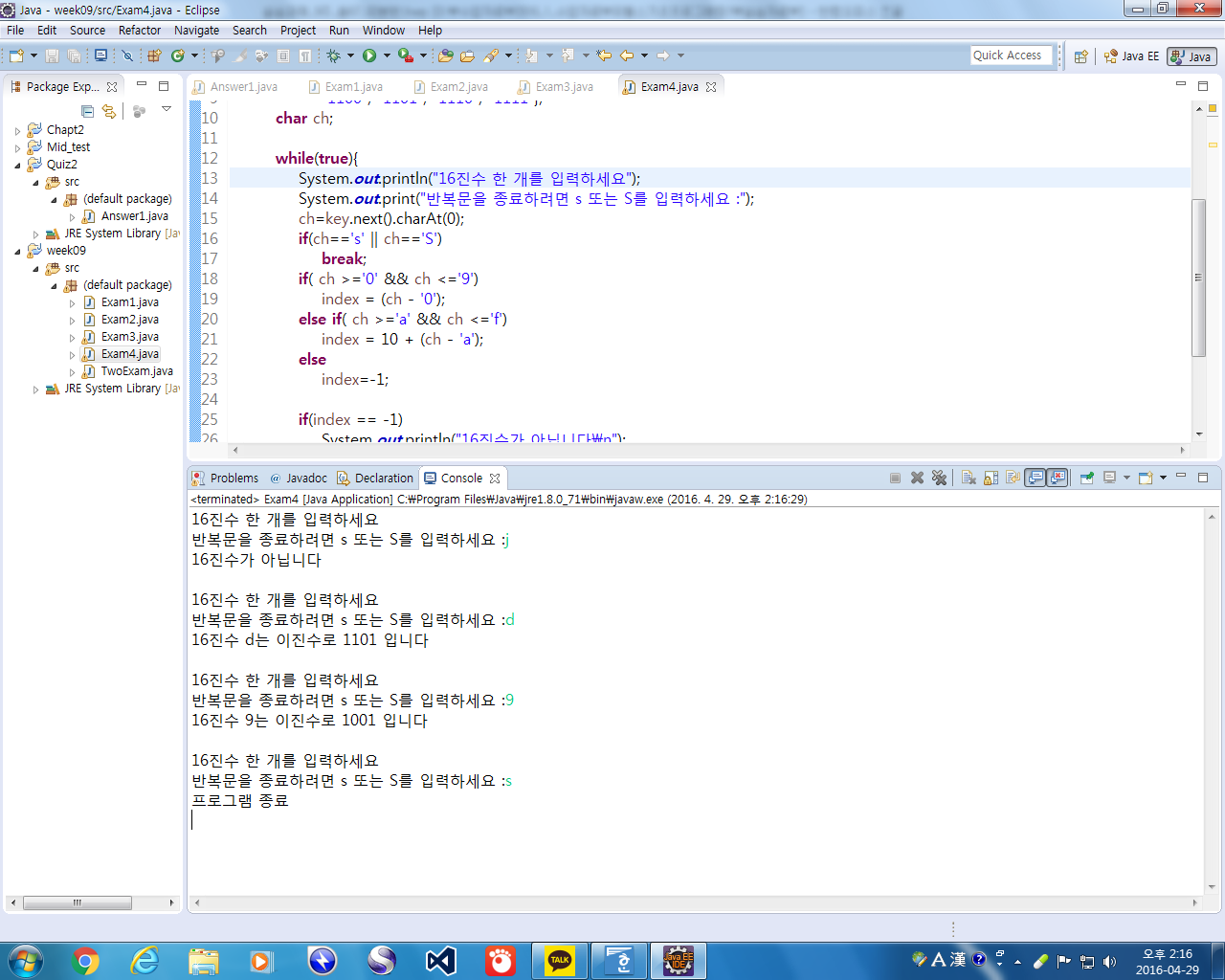
|  |
| --- |
| **[프로그램 소스]** |
| **[실행 결과]** |

1. 16진수를 2진수로 변환하는 프로그램을 작성하시오. 단, 10~15에 해당하는 16진수는 소문자로 입력.

(힌트) 16진수 ‘0’부터 ‘f’에 해당되는 다음과 같은 배열을 정의한 후 사용.

String[] hexa2bin = {"0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111",

"1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111"};

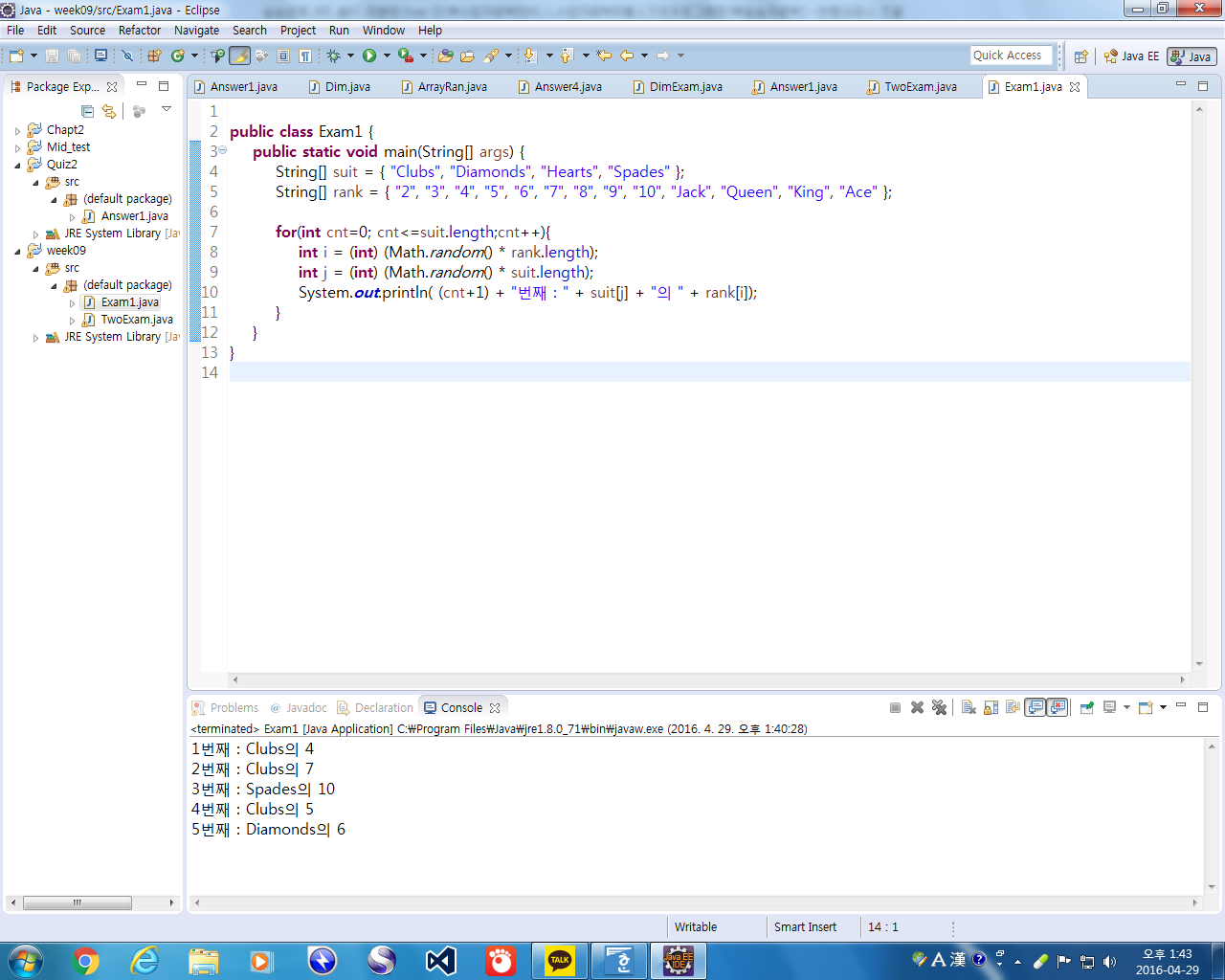


|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

1. 다음과 같은 문자열로 배열을 초기화 한 후 카드를 랜덤하게 선택하여 화면에 출력하는 코드를 작성해보자.

{"Clubs", "Diamonds", "Hearts", "Spades"}

{"2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace" }

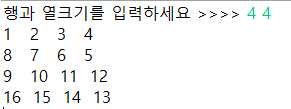


힌트) i = (int) (Math.random() \* rank.length); //i를 배열 index로 사용

|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

1. 입력한 행과 열 크기만큼 이차원 배열을 생성한 후 제시된 결과처럼 초기화 하여 출력하는 프로그램을 작성하시오.

힌트) 행 인덱스가 짝수이면 열 인덱스를 0부터, 홀수이면 열 인덱스를 열 크기-1 부터 초기화. 초기화 할 때 값을 +1 씩 증가



|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |