|  |  |  |
| --- | --- | --- |
| 2019\_2\_java1\_09 | 학번 : 20195124 | 이름 : 김민석 |

* **실행 결과 제출**

|  |
| --- |
| 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**;  **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 & 초기화 |
| **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 );    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]** |

* **내용 점검**

1. 다음 중에서 올바른 배열 선언을 모두 선택하시오.( 1, 2, 4 ) //Dog 라는 클래스가 만들어져 있을 때
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.( 1, 2, 5 )
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?( 2 )
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] int[] studentNumbers = new int[30];**

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

**[풀이] double[] values = { 1.2, 3.3, 6.7, 4.5, 1.2 };**

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

**[풀이]**

**double[] array = new double[7];**

**for(int i = 0; i < array.length; i++) {**

**array[i] = in.nextDouble();**

**}**

1. 배열 array의 크기를 출력

**[풀이]**

**for(int i = 0; i < array.length; i++) {**

**System.out.println(“array = “ + array[i]);**

**}**

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

**[수정] int[] numbers = new int[100];**

1. double rainfalls[100];

**[수정] double[] rainfalls = new double[100];**

1. char[] ch; //배열 선언 후

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

**[수정]**

**char[] ch;**

**ch = new char[] {‘d’, ‘t’, ‘s’, ‘t’, ‘a’, ‘y’};**

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

**[풀이] 0 ~ 4**

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

**[풀이] 인덱스 초과 오류**

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]**

**for(int i = 0; i < a.length; i++) {**

**b[i] = a[i];**

**}**

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[] d = {1.0, 2.0, 3.0, 4.0};**

**for(double temp : d) {**

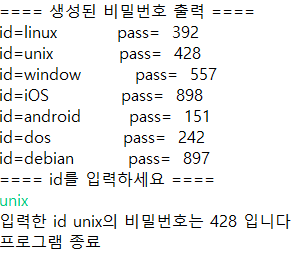
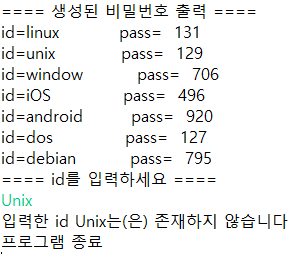
**System.out.println((i+1) + “번째 값 : “ + temp);**

**}**

* **프로그램 과제**

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

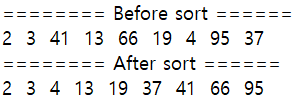
String[] id={“linux”, “unix”, “window”, “iOS”, “android”, “dos”, “debian”};

|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  public class J1\_1025\_HW1 {  public static void main(String[] args) {  Scanner in = new Scanner(System.*in*);  String[] id = {"linux", "unix", "window", "iOS", "android", "dos", "debian"};  int[] password = new int[id.length];  boolean cnd = true;   for(int i = 0; i < id.length; i++) {  password[i] = (int)(Math.*random*() \* 900) + 100;  }   System.*out*.println("==== 생성된 비밀번호 출력 ====");  for(int i = 0; i < id.length; i++) {  System.*out*.printf("id = %6s\t pass = %4d\n", id[i], password[i]);  }  System.*out*.println("==== id 를 입력하세요. ====");  String findId = in.next();   for(int i = 0; i < id.length; i++) {  if(id[i].equals(findId)) {  cnd = false;  System.*out*.println("입력한 id" + findId + "의 비밀번호는 " + password[i] + " 입니다.");  }  }  if(cnd) {  System.*out*.println("입력한 id " + findId + "는(은) 존재하지 않습니다.");  }  } } |
| **[실행결과]** |

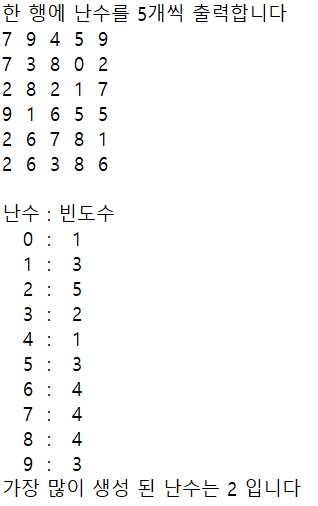
1. 다음과 같이 초기화 된 배열 원소를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오. 단, 배열 원소 출력 시 for-each 반복 루프를 사용하세요

**int** ary[] = {2, 3, 41,13, 66, 19, 4, 95, 37};



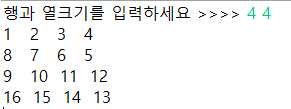
|  |
| --- |
| **[프로그램 소스]**  public class J1\_1025\_HW2 {  public static void main(String[] args) {  int[] ary = { 2, 3, 41, 13, 66, 19, 4, 95, 37};  int min, tmp;   System.*out*.println("======= Before Sort =======");  for(int temp : ary) {  System.*out*.print(temp + " ");  }   System.*out*.println("");  //정렬 구문  for(int i = 0; i < ary.length-1; i++) {  min = i;  for(int j = 1+i; j < ary.length; j++) {  if(ary[j] <= ary[min]) min = j;  }  tmp = ary[i];  ary[i] = ary[min];  ary[min] = tmp;  }   System.*out*.println("====== After Sort ======");  for(int temp : ary) {  System.*out*.print(temp + " ");  }  } } |
| **[실행결과]** |

1. 0~9사이의 난수를 30번 생성한 후 다음과 같이 빈도수를 계산하여 출력하는 프로그램을 작성하시오. 또한 가장 많이 생성된 난수도 함께 출력하시오.



|  |
| --- |
| **[프로그램 소스]**  public class J1\_1025\_HW3 {  public static void main(String[] args) {  int rnd, maxIndex = 0;  int[] bindosu = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  System.*out*.print("한 행에 난수를 5 개씩 출력합니다.");  for(int i = 0; i < 30; i++) {  rnd = (int)(Math.*random*() \* 10);  if(i%5 == 0) System.*out*.println("");  System.*out*.print(rnd + " ");  bindosu[rnd]++;  }   System.*out*.printf("\n난수 : 빈도수\n");  for(int i = 0; i < bindosu.length; i++) {  System.*out*.println(" " + i + " : " + bindosu[i]);  if(bindosu[maxIndex] <= bindosu[i]) maxIndex = i;  }  System.*out*.println("가장 많이 생성 된 난수는 " + maxIndex + " 입니다.");  } } |
| **[실행결과]** |

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



|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  public class J1\_1025\_HW4 {  public static void main(String[] args) {  Scanner in = new Scanner(System.*in*);  int row, column, plusNum = 0, i, j;   System.*out*.print("행 크기를 입력하세요 >>>> ");  row = in.nextInt();  System.*out*.print("열 크기를 입력하세요 >>>> ");  column = in.nextInt();   int[][] matrix = new int[row][column];   for(i = 0; i < row; i++) {  if (i % 2 == 0)  for (j = 0; j < column; j++) {  plusNum++;  matrix[i][j] = plusNum;  }  else  for (j = 4; j >= 0; j--) {  plusNum++;  matrix[i][j] = plusNum;  }  }   for(i = 0; i < row; i++) {  for(j = 0; j < column; j++) {  System.*out*.printf("%3d", matrix[i][j]);  }  System.*out*.println("");  }  } } |
| **[실행결과]** |

1. 인원수를 입력하면 알파고가 선택한 메뉴대로 주문해야 하는 카페가 있다. 제시된 결과처럼 실행될 수 있도록 프로그램을 작성하세요.[18점]

조건1) 무한 반복문 사용 금지

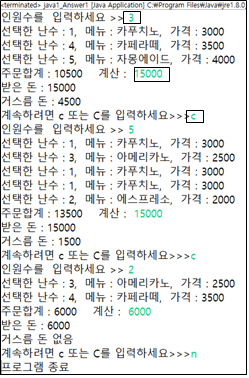
조건2) if문 사용 금지

조건3) 메뉴는 1~5 범위에서 생성된 난수 값에 따라 다음과 같이 결정 된다

|  |  |  |
| --- | --- | --- |
| 난수 | 메뉴 | 가격 |
| 1 | 카푸치노 | 3000 |
| 2 | 에스프레소 | 2000 |
| 3 | 아메리카노 | 2500 |
| 4 | 카페라떼 | 3500 |
| 5 | 자몽에이드 | 4000 |

조건 4) 박스로 표시된 부분은 입력을 받는다

조건 5) 계속하려면 문자 ‘c’ 또는 ‘C’를 입력 받는다



|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  public class J1\_1025\_HW5 {  public static void main(String[] args) {  Scanner in = new Scanner(System.*in*);  int i, rnd, pay, personNum = 0, hap = 0, returnMoney = 0;  char cnd = 'a';  String[][] menu = {{"카푸치노", "3000"},  {"에스프레소", "2000"},  {"아메리카노", "2500"},  {"카페라떼", "3500"},  {"자몽에이드", "4000"}};   do {  System.*out*.print("인원 수를 입력하세요 >> ");  personNum = in.nextInt();   for (i = 0; i < personNum; i++) {  rnd = (int)(Math.*random*() \* 5) + 1;  System.*out*.println("선택한 난수 : " + rnd + ", 메뉴 : " + menu[rnd-1][0] + ", 가격 : " + menu[rnd-1][1]);  hap += Integer.*parseInt*(menu[rnd-1][1]);  }   System.*out*.print("주문합계 : " + hap + " 계산 : ");  pay = in.nextInt();  System.*out*.println("받은 돈 : " + pay);   returnMoney = pay - hap;  System.*out*.println((returnMoney > 0)? "거스름 돈 : " + returnMoney : (returnMoney == 0)? "거스름 돈 없음 " : " 계산 불가능! ");  hap = 0;   System.*out*.print("계속하려면 c 또는 C를 입력하세요 >>> ");  cnd = in.next().charAt(0);  } while(cnd == 'c' || cnd == 'C');  System.*out*.println("프로그램 종료");  } } |
| **[실행결과]** |