**객제지향프로그래밍 3장 실습 문제 보고서**

글로벌미디어학부20192722 이지희

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2번 | 4번 | 6번 | 8번 | 10번 | 12번 | 14번 | 16번 |
| O | O | O | O | O | O | O | O |

**2번**

|  |  |
| --- | --- |
| 소스코드 | |
| public class OPP\_3\_2 {  public static void main(String[] args) {  int n[][]={{1},{1,2,3},{1},{1,2,3,4},{1,2}};  for(int i=0;i<n.length;i++){  for(int j=0;j<n[i].length;j++){  System.*out*.print(n[i][j] + " ");  }  System.*out*.println();  }  } }} | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |

**4번**

|  |  |
| --- | --- |
| 소스코드 | |
| import java.util.Scanner;  public class OPP\_3\_4 {  public static void main(String[] args) {  System.*out*.print("소문자 알파벡 하나를 입력하시오.>>");  Scanner scanner = new Scanner(System.*in*);  String s = scanner.next();  char c=s.charAt(0);  for(int i=0;i<c-'a'+1;i++){  for(int j=0;j<c-'a'-i+1;j++){  char outChar =(char)('a' + j);  System.*out*.print(outChar);  }  System.*out*.println();  }  scanner.close();  } } | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 부연 설명 | 힌트와 같이 scanner.next();는 string형 이므로 char형 변수를 만들어서 입력 받은 문자열의 첫 글자를 저장한다. 그 후에는 아스키 코드를 이용하였다.   * 반복문 1 : 몇개의 행을 출력할지 결정   for(int i=0;i<c-'a'+1;i++)  입력 받은 문자가 알파벳의 몇번째 문자열인지 확인하기 위해서 입력 받은 문자의 아스키코드에서 ‘a’의 아스키코드를 빼고 +1을 하였다.   * 반복문 2 : 몇 번째 알파벳까지 출력할지 결정 후 출력   for(int j=0;j<c-'a'-i+1;j++)  첫번째 줄에서는 ‘a’부터 입력 문자까지 출력해야 하고 그 다음 줄은 그 전(입력문자 - 1)까지, 마지막에는 ‘a’만 출력해야 하므로 위의 몇번째 문자열인지 확인한 c-‘a’+1에 i를 빼는 처리를 해주었다. |

**6.**

|  |  |
| --- | --- |
| 소스코드 | |
| import java.util.Scanner;  public class OOP\_3\_6 {  public static void main(String[] args) {  int [] uint={50000,10000,1000,500,100,50,10,1};  int[] count=new int[8];  System.*out*.print("금액을 입력하시오.>>");  Scanner scanner = new Scanner(System.*in*);  int money=scanner.nextInt();  //지폐 세기  for(int i=0;i<uint.length;i++){  count[i]=money/uint[i];  if(count[i]!=0)  System.*out*.println(uint[i]+"원 짜리: "+count[i]);  money%=uint[i];  }  scanner.close();  } } | |
| 실행 화면 |  |
| 부연 설명 | 각 지페 / 동전의 개수를 저장하는 count 배열을 만든 후 크기를 단위의 개수인 8로 초기화 한다.    지폐의 개수는 입력된 금액을 각 단위로 나누었을 때 몫이다. 개수가 0이 아니라면 지폐/동전의 개수가 몇개인지 출력한다. 그 다음 단위에서 지폐를 세기 위해서 money의 값을 단위로 나눈 나머지로 변경한다.  출력시에는 출력 예시와 같이 0개인 경우는 제외하고 출력하였다. |

**8.**

|  |  |
| --- | --- |
| 소스코드 | |
| import java.util.Scanner;  public class OPP {  public static void main(String[] args) {  Scanner scanner=new Scanner(System.*in*);  System.*out*.print("정수 몇개?");  int num= scanner.nextInt();  int[] numArr=new int[num];   for(int i=0;i<num;i++){  numArr[i]=(int)(Math.*random*()\*100+1);  for(int j=0;j<i;j++){ //유일한지 탐색  if(numArr[j]==numArr[i]) {  i--;  break;  }  }}  for(int k=0;k<num;k++){  System.*out*.print(numArr[k]+" ");  if((k+1)%10==0) System.*out*.println();  }   } } | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 부연 설명 | Math.random() 함수의 범위가 0이상 1 미만이기 때문에 100까지의 랜덤한 수를 생성하기 위해서 Math.random()\*100+1로 처리했다. 또한 배열 내에 동일한 수가 있으면 안되기 때문에 생성할 때마다 for()문을 이용하여 이 전까지 같은 수가 있었는지 체크하는 과정을 갖는다.  출력시에는 출력 예시에서 숫자 10개마다 개행을 해주기 때문에 10번째가 될 때 마다 println()으로 개행을 해주었다. |

**10**

|  |  |
| --- | --- |
| 소스코드 | |
| public class OPP\_3\_10 {  public static void main(String[] args) {  int [][] numArr2d=new int[4][4];  int randompos1, randompos2, n=0;  for(int i=0;i<numArr2d.length;i++){  for(int j=0;j<numArr2d[i].length;j++){  numArr2d[i][j]=(int)(Math.*random*()\*10+1);  }  }  while (n<6){  randompos1=(int)(Math.*random*()\*4);  randompos2=(int)(Math.*random*()\*4);  if(numArr2d[randompos1][randompos2]!=0) {  numArr2d[randompos1][randompos2] = 0;  n++;  }  }  for (int y=0;y<numArr2d.length;y++){  for(int x=0;x<numArr2d[y].length;x++){  System.*out*.print(numArr2d[y][x] + " ");  }  System.*out*.println();  }  } } | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 부연 설명 | 처음에는 4x4 2차원 배열 numArr2d에 모두 10 이하의 정수를 삽입해준 후 4까지의 랜덤한 수를 생성해서 그 자리에 0을 삽입해준다.  랜덤한 자리에 0을 삽입하기 위해서 random()를 사용하는데, random()의 범위가 0이상 1미만이기 때문에 Math.random()\*4 를 하면 범위가 0이상 4미만이 되고 정수로 강제 형변환을 하기 때문에 0,1,2,3 4개 중 하나의 수가 랜덤하게 생성된다. 이를 인덱스로 넣어서 배열에 접근하여 0을 삽입하는 방법을 사용했다. |

**12**

|  |  |
| --- | --- |
| 소스코드 | |
| public class Add {  public static void main(String[] args) {  int sum=0, num;  for (int i=0;i<args.length;i++){  try  {  num=Integer.*parseInt*(args[i]);  sum+=num;  }  catch (NumberFormatException e)  {  }  }  System.*out*.println(sum);  } } | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 부연 설명 | NumberFomatException 예외를 try catch문을 이용하여 처리해서 정수가 아닌 경우에 아무런 처리를 하지 않도록 작성하였다. 숫자인 경우에는 Intger의 parseInt() 메서드를 활용하여 int형으로 만든 후 sum에 더해주었다.  (실행 화면)Mac 터미널에서는 javac 명령어로 Add.java파일을 컴파일 한 후 실행 예시와 같이 실행해주었다. |

**14.**

|  |  |
| --- | --- |
| 소스코드 | |
| import java.util.Scanner;  public class OPP\_3\_14 {  public static void main(String[] args) {  String course[]={"Java","C++","HTML5","컴퓨터구조","안드로이드"};  int score[]={95,88,76,62, 55};  String inString="";   Scanner scanner = new Scanner(System.*in*);  while (true){  int index=-1;  System.*out*.print("과목 이름>>");  inString=scanner.next();  if(inString.equals("그만")) break;  for(int i=0;i<course.length;i++){  if(inString.equals(course[i])) index=i;  }  if(index>-1) System.*out*.println(course[index]+"의 점수는 "+score[index]);  else System.*out*.println("없는 과목입니다. ");   }  scanner.close();  } } | |
| 실행 화면 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 부연 설명 | 우선 과목명을 정확하게 입력한 경우에만 점수가 출력되도록 하였다. 즉, 대소문자를 구분하여 Java가 아닌 java라고 입력하거나 C++이 아닌 c++로 입력한 경우는 없는 과목이라고 생각하고 코드를 작성하였다.  inString이라는 문자열 변수에 과목 이름을 입력받고 for문을 이용하여 입력한 문자열과 course배열에 있는 문자열이 같은 경우가 있는지 탐색하였다. 탐색 후에 있다면 index 변수에 그 문자열이 있는 위치를 저장하여 출력시에 사용하였다. 그래서 인덱스의 초기값인 -1인 경우는 course 배열에 입력한 문자열이 없는 경우 이므로 -1보다 큰 인덱스 값을 갖고 있는 경우에만 과목과 점수를 출력하도록 코드를 작성하였다. |

**16번**

|  |  |
| --- | --- |
| 소스코드 | |
| import java.util.Scanner;  public class OPP\_3\_16 {  public static void main(String[] args) {  String str[]={"가위","바위","보"};  String yourStr;  int comindex;  Scanner scanner=new Scanner(System.*in*);   System.*out*.println("컴퓨터와 가위바위보 게임을 합니다.");  while (true){  System.*out*.print("가위 바위 보!");  yourStr=scanner.next();  if(yourStr.equals("그만")) break;  comindex=(int)(Math.*random*()\*3);  if(yourStr.equals(str[comindex])){  System.*out*.println("사용자: "+yourStr+" 컴퓨터: "+str[comindex]+", 비겼습니다.");  }  else if((yourStr.equals("가위") && str[comindex].equals("보"))  || (yourStr.equals("바위") && str[comindex].equals("가위"))  || (yourStr.equals("보") && str[comindex].equals("바위"))){  System.*out*.println("사용자: "+yourStr+" 컴퓨터: "+str[comindex]+", 사용자가 이겼습니다.");  }  else{  System.*out*.println("사용자: "+yourStr+" 컴퓨터: "+str[comindex]+", 컴퓨터가 이겼습니다.");  }   }  } } | |
| 실행 화면 | **텍스트이(가) 표시된 사진  자동 생성된 설명** |
| 부연 설명 | 사용자 기준으로 가위 바위 보의 모든 경우를 작성하면 아래 표와 같다.   |  |  |  |  | | --- | --- | --- | --- | | 사용자  컴퓨터 | 가위 | 바위 | 보 | | 가위 | 비김 | 이김 | 짐 | | 바위 | 짐 | 비김 | 이김 | | 보 | 이김 | 짐 | 비김 |   따라서 3가지 케이스 ( 사용자가 이기는 경우, 사용자가 지는 경우, 비기는 경우) 로 나누어 if을 이용하여 각각 출력되도록 하였다. 컴퓨터가 낼 것을 고르는 것과, 문자열을 비교하는 것은 문제의 힌트를 이용하여 작성하였다. 또한 while문을 이용하여 무한 루프를 만들고 “그만”이 입력되면 break;를 통해 루프를 빠져나오도록 만들었다. |