JAVA 프로그래밍 – 학기말고사 1/2

학년:　　학번:　　이름:　　.

참고 :

1. 개념 작성 문제의 경우 큰 흐름상 이상이 없으면 정답, 단 굵은 글씨체의 키워드 혹은 유사 개념이 포함하여야 함
2. 소스를 작성하는 문제의 경우 참고의 정의가 있는 경우 정확히 기입하여야 정답, 그렇지 않는 경우 개념만 맞으면 정답
3. 소스를 작성하는 문제의 경우 공개된 정답에 의하여서만 채점, 본인 코드가 정확하다고 생각되면 스스로 컴파일하여 동작여부를 증명하여 점수를 획득 할 것

1. 테스트주도개발의 필요성에 대하여 간략히 설명하시오. (10점)

작은 단위의 프로시저 혹은 객체에 대하여 **무 결성을 테스트하여 검증**하여 검증된 요소기술의 집합으로 작동 가능한 최종 결과물을 산출하기 위함

유사의미 정답

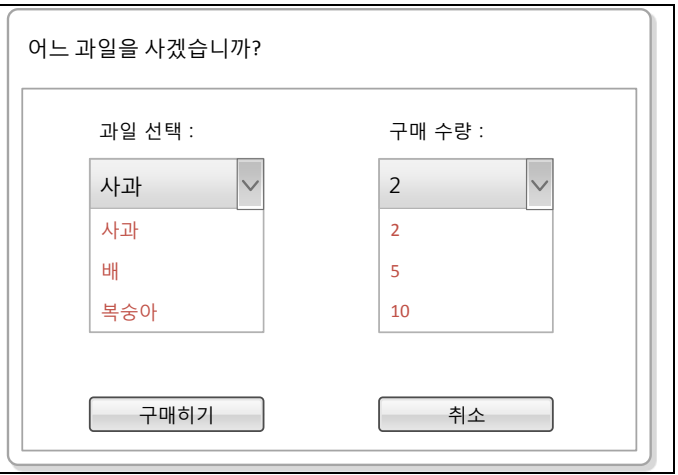
2. 네트워크 통신을 위한 필수 요소를 4가지 이상 제시하고 제시된 각각의 요소에 대하여 간략히 설명하시오. (10점)

프로토콜 : 통신망에 접속된 컴퓨터를 발견하고 정보를 교환하는 데 필요한 통신절차
IP(Internet Protocol) 주소 : 인터넷에서 특정한 컴퓨터를 찾기 위한 주소
도메인 명칭(Domain Name) : 인터넷 주소를 문자의 형태로 표시한 이름(예: klas.khu.ac.kr)
URL(Uniform Resource Locator) 주소 : 통상 웹사이트 주소라고 함
(예: http://www.mycompany.com)
소켓 : 응용 프로그램에 통신기능을 제공하는 라이브러리 프로그램
포트 : 한 IP를 공유하는 소켓의 위치 혹은 주소
등
유사의미 정답

3. 아래에 제시된 과일 구매 프로그램에 대하여 단점을 2가지 이상 제시하고 이를 해결하기 위한 새로운 GUI를 간략한 그림으로 제시하고 동작 과정을 간략히 설명하시오. (10점)

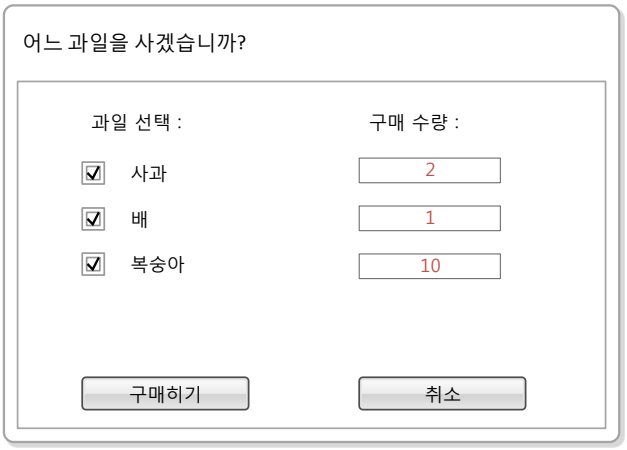
<콤보 박스로 구동되는 GUI>

콤보 박스는 버튼을 클릭하면 그림의 목록이 출현 되고 이 목록을 선택하는 과정으로 UI가 동작됨



정답 :

문제점
과일을 두 가지 이상 선택 불가능하고, 각각에 대하여 구매 수량을 결정할 수 없음,
구매 가능한 과일 목록을 콤보박스 클릭 후에야 확인 가능.
기본 제시된 과일을 구매하는 경우가 아니라면, 구매하기 까지 최소 5번의 마우스 클릭 필요
등등



체크 박스로 각각의 구매 목록을 선택하고 수량을 텍스트 상자에 기입 함으로서 상기 문제를 해결 함
한눈에 구매 가능한 과일 목록 확인 가능
한가지 과일만 살 경우 두 번의 마우스클릭과 한번의 키보드 입력으로 문제의 UI보다 효율 적임

제시된 정답 이외에 개선 효과가 전체 경우의 수가 마우스 클릭 한번이라도 줄일 수 있으면 정답. 이외의 모든 고민 흔적은 5점 이하의 부분 점수 .

4. 아래 소스코드를 참고하여 물음에 답하시오. (10점)

```
1   public class GetMaxNum
2   {
3       public static void main( String[] arg )
4       {
5           int nArray[] = {10,50,60,80,0,70,90,-1};
6           int i    = 0 ;
7           int nMax = 0 ;
8           int nIdx = 0 ;
9           while( nArray[i] != -1 )
10          {
11            if( nArray[i] == 0 )
12               break;
13            if( nMax < nArray[i] )
14            {
15                    nMax = nArray[i];
16                    nIdx = i;
17            }
18          }
19          System.out.println("nArray[" + nIdx + "] = " + nMax );
20      }
21  }
```

a. 소스 코드가 갖는 문제점을 한가지 지적하고, 해결 방법에 대하여 왼쪽의 번호를 참고하여 추가되어야 할 위치와 코드를 제시하라. (5점)

문제점 : while 내에서 i의 증가가 없어 계속 0인상태로 무한 루프를 돌게 됨

해결방법 : 17,18 라인 사이에 다음 코드 추가 : i++;

각각 5점

b. 상기 문제가 해결 된 이후의 예상되는 출력 결과를 제시하라. (5점)

정답 : nArray[3] = 80

오답의 경우 제시 결과에 따라 2점이하

출력 형식이 맞고 정답이 아닌 경우 2점, 인덱스와 값이 맞으나 형식이 틀린 경우 2점

※아래 소스코드를 참고하여 물음에 답하시오. (5번~6번 문제)

```
import java.awt.*;
import java.awt.event.*;

class ButtonHandler implements ①_________________ {
    public void ②_________________{
         System.out.println("버튼의 명칭 : " + e.getActionCommand());
   }
}
```

```
public class ButtonTest extends Frame {
        public static void main(String[] args) {
                Frame f = new Frame("버튼 이벤트 시험");
                Button b1 = new Button("버튼 1");
                Button b2 = new Button("버튼 2");
                Button b3 = new Button("버튼 3");
                Button b4 = new Button("버튼 4");
                ③_____; // 버튼 1 누름을 청취
                ④_____; // 버튼 2 누름을 청취
                ⑤_____; // 버튼 3 누름을 청취
                ⑥_____; // 버튼 4 누름을 청취
                ⑦_____ // 버튼의 배치
                    ……
                ⑦_____ // 버튼의 배치
                ⑧_____;// 프레임 크기 지정 가로=200, 세로=150
                ⑨_____; // 컴포넌트의 화면 출력 명령
        }
}
```

5. 주어진 소스코드에서 버튼 누름 이벤트에 대한 청취가 가능하도록 ①~⑥에 알맞은 코드를 완성하고, "b4" 객체의 버튼을 누를 경우 의 출력결과를 제시 하시오.(10점)

정답

① : ActionListener

② : actionPerformed(ActionEvent e)

③ : b1.addActionListener(new ButtonHandler());

④ : b2.addActionListener(new ButtonHandler());

⑤ : b3.addActionListener(new ButtonHandler());

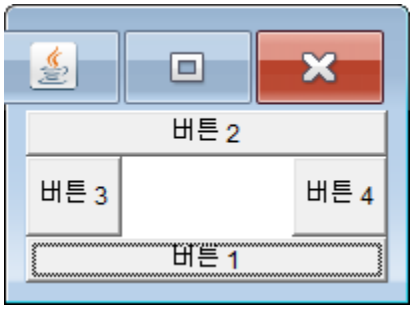
⑥ : b4.addActionListener(new ButtonHandler());

→ 6점 , 각각 1점

출력 결과 : "버튼의 명칭 : 버튼 4"

➔ 4점

6. 주어진 소스코드에서 버튼의 위치가 아래와 같도록 배치하고 주어진 크기로 화면에 출력하도록 ⑦~⑨에 알맞은 코드를 완성하시오.(10점)



프레임 크기 → 가로=200, 세로=150

정답

⑦ : f.add(b1, BorderLayout.*SOUTH*); // 버튼의 배치
f.add(b2, BorderLayout.*NORTH*); // 버튼의 배치
f.add(b3, BorderLayout.*WEST*); // 버튼의 배치
f.add(b4, BorderLayout.*EAST*); // 버튼의 배치

⑧ : f.setSize(200, 150);

⑨ : f.setVisible(**true**);

7. 아래 소스코드를 참고하여 물음에 답하시오. (10점)

```
public class socket_client{
        public static final int BUFFER_SIZE = 644096;

        public static void main(String[] args) {

                String strData =
                        "①_____"
                        + "Host: JavaPrograming\n"
                        + "User-Agent: Mozilla/4.0 (compatible: MSIE 7.0; Windows NT 5.1)\n"
                        + "\n";
                byte[] buffer = new byte[BUFFER_SIZE];
                ②_____;//Host의 주소 선언 및 정의
                ③_____;//Host의 접속 포트 선언 및 정의
                Socket soHttp    = null;

                try{
                        ④_____;//Socket의 인스턴트 생성
                        ⑤_____;//OutputStream 획득
                        ⑥_____;//DataOutputStream의 인스턴트 생성
                        ⑦_____;//HTTP를 통해 Host로 데이터 전송

                        InputStream  is = soHttp.getInputStream();
                        DataInputStream dis = new DataInputStream(is);
                        int nOff  = 0;
                        int nRead = 0;

                        do{
                                nRead = dis.read(buffer, nOff, BUFFER_SIZE);
                                if( nRead <= 0 ) break;

                                String strRead = new String(buffer);
                                System.out.print( strRead );
                        }while(nRead == BUFFER_SIZE);

                        ⑧_____;//Socket 닫기
```

```
                }catch(Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

a. ③에 알맞은 아래의 URL의 그림파일을 socket 통신을 이용하여 획득하기 위한 GET방식의 HTTP 요청 문자열을 아래 표를 참고하여 완성하시오. (5점)
그림 파일 url : http://www.khu.ac.kr/upload/mainImg/main_150605a_01.jpg
www.khu.ac.kr의 IP : 163.180.96.211

참고 표)

url	http://klas.khu.ac.kr/
ip	163.180.96.213
http port	80
http request	"GET / HTTP/1.1₩n Host: JavaPrograming₩n User-Agent: Mozilla/4.0 (compatible: MSIE 7.0; Windows NT 5.1)₩n ₩n"

정답 :
`GET /upload/mainImg/main_150605a_01.jpg HTTP/1.1\n`

Char 단위로 틀린 점이 발견 되면 -1점 감점

b. 소켓 통신을 이용하여 데이터를 획득하는 과정에 대한 ②~⑧의 알맞은 코드를 완성하시오.(10점)

아무런 코드가 없으면 0점
타입, 함수 및 파라미터 등의 잘못된 사용이 발견 되면 -2점

정답 :

```
import java.io.*;
import java.net.*;

public class socket_client{
        public static final int BUFFER_SIZE = 644096;

        public static void main(String[] args) {

                String strServer = "163.180.96.211";//Or "klas.khu.ac.kr";//Host 선언 및 정의
                int    nPort     = 80;//Host의 접속 포트 선언 및 정의
```

```
                Socket soHttp    = null;
                String strData =
                        "GET /upload/mainImg/main_150605a_01.jpg HTTP/1.1\n"
                        + "Host: JavaPrograming\n"
                        + "User-Agent: Mozilla/4.0 (compatible: MSIE 7.0; Windows NT 5.1)\n"
                        + "\n";
                byte[] buffer = new byte[BUFFER_SIZE];

                try{
                        soHttp = new Socket( strServer, nPort );//Socket의 인스턴트 생성
                        OutputStream os = soHttp.getOutputStream();//OutputStream 획득
                        DataOutputStream dos = new DataOutputStream(os);//DataOutputStream의
인스턴트 생성

                        dos.writeBytes(strData);//HTTP를 통해 Host로 데이터 전송

                        InputStream  is = soHttp.getInputStream();
                        DataInputStream dis = new DataInputStream(is);
                        int nOff  = 0;
                        int nRead = 0;

                        do{
                                nRead = dis.read(buffer, nOff, BUFFER_SIZE);
                                if( nRead <= 0 ) break;

                                String strRead = new String(buffer);
                                System.out.print( strRead );
                        }while( (nRead < BUFFER_SIZE)  );

                        soHttp.close();//Socket 닫기

                }catch(Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

8. 아래 소스 코드를 보고 물음에 답하시오.

a. 아래 소스 코드를 컴파일 하여 실행할 경우, 어떤 문제가 발생할 수 있다. 발생할 수 있는 문제가 무엇인지 작성하시오

b. 정상적으로 실행될 수 있도록 소스 코드를 수정하시오.

```
class Library{
  private int totalBooks;
  public Library(int total){
    this.totalBooks = total;
  }

  void ReturnBook(int books, String name){
    totalBooks += books;
    System.out.println("[" + name + "]반납한 책 : " + books + "권");
  }

  void CheckoutBook(int books, String name){
    if((totalBooks - books) > 0){
      totalBooks -= books;
      System.out.println("[" + name + "]대출한 책 : " + books + "권");
    }else
      System.out.println("[" + name + "]더 이상 대출할 수 있는 책이 없습니다.");
```

```
  }

  public void GetBooks(){
    System.out.println("현재 대출 가능한 책은 총 " + totalBooks + "권입니다.");
  }
}

class Student extends Thread{
  boolean flag = false; // 반납/대출
  Library obj;

  public Student(Library obj, String name){
    super(name);
    this.obj = obj;
  }

  public void run(){
    for(int i=0; i<5; i++){
      try{
        sleep(500);
      }
      catch(InterruptedException e){}
      if(flag){
        obj.ReturnBook((int)(Math.random()*10), getName());
        obj.GetBooks();
      }else{
        obj.CheckoutBook((int)(Math.random()*10), getName());
        obj.GetBooks();
      }
      flag = !flag;
    }
  }
}

public class FinalExam1 {
  public static void main(String[] args) {
    Library obj = new Library(10);
    Student stu1 = new Student(obj, "장학생");
    Student stu2 = new Student(obj, "복학생");
    Student stu3 = new Student(obj, "휴학생");

    stu1.start();
    stu2.start();
    stu3.start();
  }
}
```

정답

a. 공용 리소스의 접근 문제 발생 가능(혹은 동기화 문제가 발생 할 수 있음)

b. 아래 표 참고

```
synchronized (obj) {
       if(flag){
              obj.ReturnBook((int)(Math.random()*10), getName());
```

```
                obj.GetBooks();
        }else{
                obj.CheckoutBook((int)(Math.random()*10), getName());
                obj.GetBooks();
        }
}
```

9. 아래 소스코드는 "Data.txt" 파일을 입력 받아 3개의 성적의 평균을 계산하여, 평균 성적이 높은 학생 순으로 출력하는 프로그램의 일부이다. 다음 물음에 답하시오.

a. ①의 영역에 읽어 들인 한 줄 문자열 line에 대하여 탭("₩t")에 의하여 구분하여 정보가 분류되어 입력 아규먼트 vt에 추가 되도록 적절한 코드를 완성하시오.(성적은 평균으로 산출해야 하며, 소수점까지 표현해야 함)(10점)

b. ②의 영역에 평균 성적이 높은 순서로 학생 정보가 출력 벡터 vtOut에 추가되도록 적절한 코드를 완성하시오.(10점)

Data.txt

```
2014103161    홍길동  95    87    60
2014104102    강석진  88    100   65
2014104163    김판수  76    21    78
2015104098    이필립  57    45    84
```

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.Vector;

class StuInfo{
	String stuName;
	int stuNum;
	double average;
	StuInfo(String stuName, int stuNum, double average){
		this.stuName = stuName;
		this.stuNum = stuNum;
		this.average = average;
	}
	StuInfo(StuInfo stu){
		this.stuName = stu.stuName;
		this.stuNum = stu.stuNum;
		this.average = stu.average;
	}
}

class FileControl{
	public void Write(Vector<StuInfo> vt){
		// 소스코드 생략
	}
	public void Read(Vector<StuInfo> vt){
		BufferedReader reader = null;
		try{
			reader = new BufferedReader(new
FileReader("C:/Users/Hong/Documents/data.txt"));
			String line = reader.readLine();
			while( line != null){
		// ① 한 줄 문자열을 탭("\t")에 의하여 구분하여 정보 분류
			}
```

```
                } catch(IOException e){
                        System.out.println(e.getMessage());
                } finally{
                        try{ if(reader != null) reader.close(); } catch(Exception
e){}
                        }
                }
}

public class FinalExam3 {
        public static void main(String[] args){
                FileControl fc = new FileControl();
                Vector<StuInfo> vtIn = new Vector<StuInfo>();
                Vector<StuInfo> vtOut = new Vector<StuInfo>();

                fc.Read(vtIn);

                // ② 평균 성적을 기준으로 학생들을 내림차순 정렬

                fc.Write(vtOut);
        }
}
```

정답

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.Vector;


class StuInfo{
	String stuName;
	int stuNum;
	double average;
	StuInfo(String stuName, int stuNum, double average){
		this.stuName = stuName;
		this.stuNum = stuNum;
		this.average = average;
	}
	StuInfo(StuInfo stu){
		this.stuName = stu.stuName;
		this.stuNum = stu.stuNum;
		this.average = stu.average;
	}
}

class FileControl{
	public void Write(Vector<StuInfo> vt){
		try{
			FileWriter fw = new
FileWriter("C:/Users/Hong/Documents/result.txt");
			int nSize = vt.size();
			for(int i=0; i<nSize; i++)
				fw.write(vt.get(i).stuNum + "\t" +
vt.get(i).stuName + "\t" + vt.get(i).average + "\r\n");
			fw.close();
		}catch(IOException e){
			System.out.println(e.getMessage());
		}finally{}
	}
	public void Read(Vector<StuInfo> vt){
		BufferedReader reader = null;
		try{
			reader = new BufferedReader(new
FileReader("C:/Users/Hong/Documents/data.txt"));
			String line = reader.readLine();
			while( line != null){
				StringTokenizer st = new StringTokenizer(line,
"\t");
				while(st.hasMoreTokens()){
					int stuNum =
Integer.parseInt(st.nextToken());
					String stuName = st.nextToken();
					int a = Integer.parseInt(st.nextToken());
					int b = Integer.parseInt(st.nextToken());
					int c = Integer.parseInt(st.nextToken());
					double avg = (a+b+c)/3.;
					System.out.println(stuNum + " " + stuName +
" " + avg);
```

```
                                        StuInfo stu = new StuInfo(stuName, stuNum,
avg);
                                        vt.add(stu);
                                }

                                line = reader.readLine();
                        }
                } catch(IOException e){
                        System.out.println(e.getMessage());
                } finally{
                        try{ if(reader != null) reader.close(); } catch(Exception
e){}
                        }
                }
}

public class FinalExam3 {
        public static void main(String[] args){
                FileControl fc = new FileControl();
                Vector<StuInfo> vtIn = new Vector<StuInfo>();
                Vector<StuInfo> vtOut = new Vector<StuInfo>();

                fc.Read(vtIn);

                // 평균 성적을 기준으로 학생들을 내림차순 정렬
                int nSize = vtIn.size();
                while( nSize != 0 ){
                        int maxIndex = 0;
                        for(int i=1; i<vtIn.size(); i++){
                                if( vtIn.get(maxIndex).average <
vtIn.get(i).average )
                                        maxIndex = i;
                        }
                        StuInfo stuInfo = new StuInfo(vtIn.get(maxIndex));
                        vtOut.add(stuInfo);
                        vtIn.remove(maxIndex);
                        nSize = vtIn.size();
                }

                fc.Write(vtOut);
        }
}
```

JAVA 프로그래밍 – 학기말고사 "참고"

학년: 학번: 이름: .

http://docs.oracle.com/javase/7/docs/api/java/의 문서를 참고하였습니다.

java.net

Class Socket

Constructors :	
Socket()	Creates an unconnected socket, with the system-default type of SocketImpl.
Socket(InetAddress address, int port)	Creates a stream socket and connects it to the specified port number at the specified IP address.
Socket(InetAddress address, int port, InetAddress localAddr, int localPort)	Creates a socket and connects it to the specified remote address on the specified remote port.
Socket(Proxy proxy)	Creates an unconnected socket, specifying the type of proxy, if any, that should be used regardless of any other settings.
Socket(SocketImpl impl)	Creates an unconnected Socket with a user-specified SocketImpl.
Socket(String host, int port)	Creates a stream socket and connects it to the specified port number on the named host.
Socket(String host, int port, boolean stream)	Deprecated. Use DatagramSocket instead for UDP transport.
Methods :	
void close()	Closes this socket.
OutputStream getOutputStream()	Returns an output stream for this socket.
InputStream getInputStream()	Returns an input stream for this socket.

java.io

Class OutputStream

Constructors :	
OutputStream()	
Methods :	
void close()	Closes this output stream and releases any system resources associated with this stream.
void flush()	Flushes this output stream and forces any buffered output bytes to be written out.
void write(byte[] b)	Writes b.length bytes from the specified byte array to this output stream.
void write(byte[] b, int off, int len)	Writes len bytes from the specified byte array starting at offset off to this output stream.
abstract void write(int b)	Writes the specified byte to this output stream.

java.io

Class DataOutputStream

Constructors :	
DataOutputStream(OutputStream out)	Creates a new data output stream to write data to the specified underlying output stream.
Methods :	
void flush()	Flushes this data output stream.
int size()	Returns the current value of the counter written, the number of bytes written to this data output stream so far.
void write(byte[] b, int off, int len)	Writes len bytes from the specified byte array starting at offset off to the underlying output stream.
void write(int b)	Writes the specified byte (the low eight bits of the argument b) to the underlying output stream.
void writeBoolean(boolean v)	Writes a boolean to the underlying output stream as a 1-byte value.
void writeByte(int v)	Writes out a byte to the underlying output stream as a 1-byte value.
void writeBytes(String s)	Writes out the string to the underlying output stream as a sequence of bytes.
void writeChar(int v)	Writes a char to the underlying output stream as a 2-byte value, high byte first.
void writeChars(String s)	Writes a string to the underlying output stream as a sequence of characters.
void writeInt(int v)	Writes an int to the underlying output stream as four bytes, high byte first.

java.io

Class DataInputStream

Constructors :	
DataInputStream(InputStream in)	Creates a DataInputStream that uses the specified underlying InputStream.
Methods :	
int read(byte[] b)	Reads some number of bytes from the contained input stream and stores them into the buffer array b.
int read(byte[] b, int off, int len)	Reads up to len bytes of data from the contained input stream into an array of bytes.
boolean readBoolean()	See the general contract of the readBoolean method of DataInput.
byte readByte()	See the general contract of the readByte method of DataInput.
char readChar()	See the general contract of the readChar method of DataInput.
double readDouble()	See the general contract of the readDouble method of DataInput.
float readFloat()	See the general contract of the readFloat method of DataInput.

java.util

Class StringTokenizer

Constructors :	
StringTokenizer(String str)	Constructs a string tokenizer for the specified string.
StringTokenizer(String str, String delim)	Constructs a string tokenizer for the specified string.
Methods :	
String nextToken()	Returns the next token from this string tokenizer.
boolean hasMoreTokens()	Tests if there are more tokens available from this tokenizer's string.

java.util

Class Random

Constructors :	
Random()	Creates a new random number generator.
Methods :	
int nextInt(int n)	Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.

java.util

Class Vector<E>

Constructors :	
Vector()	Constructs an empty vector so that its internal data array has size 10 and its standard capacity increment is zero.
Methods :	
boolean add(E e)	Appends the specified element to the end of this Vector.
E get(int index)	Returns the element at the specified position in this Vector.
E remove(int index)	Removes the element at the specified position in this Vector.

java.awt

Class BorderLayout

Constructors :	
BorderLayout()	Constructs a new border layout with no gaps between components
BorderLayout(int hgap, int vgap)	Constructs a border layout with the specified gaps between components.
Methods :	
int getHgap()	Returns the horizontal gap between components.
int getVgap()	Returns the vertical gap between components.
void setHgap(int hgap)	Sets the horizontal gap between components.
void setVgap(int vgap)	Sets the vertical gap between components.
Fields :	
CENTER	The center layout constraint (middle of container).
NORTH	The north layout constraint (top of container).
WEST	The west layout constraint (left side of container).
EAST	The east layout constraint (right side of container).
SOUTH	The south layout constraint (bottom of container).

java.awt

Class GridLayout

Constructors :	
GridLayout()	Creates a grid layout with a default of one column per component, in a single row.
GridLayout(int rows, int cols)	Creates a grid layout with the specified number of rows and columns.
GridLayout(int rows, int cols, int hgap, int vgap)	Creates a grid layout with the specified number of rows and columns.
Methods:	
int getHgap()	Returns the horizontal gap between components.
int getVgap()	Returns the vertical gap between components.
void setHgap(int hgap)	Sets the horizontal gap between components.
void setVgap(int vgap)	Sets the vertical gap between components.

java.awt

Class Button

Constructors :	
Button()	Constructs a button with an empty string for its label.
Button(String label)	Constructs a button with the specified label.
Methods:	
String getLabel()	Gets the label of this button.
void setLable(String label)	Sets the button's label to be the specified string.
void addActionListener(ActionListener l)	Adds the specified action listener to receive action events from this button.

java.awt

Class Checkbox

Constructors :	
Checkbox()	Creates a check box with an empty string for its label.
Checkbox(String label)	Creates a check box with the specified label.
Checkbox(String label, boolean state)	Creates a check box with the specified label and sets the specified state.
Methods:	
String getLabel()	Gets the label of this button.
void setLable(String label)	Sets the button's label to be the specified string.
boolean getState()	Determines whether this check box is in the "on" or "off" state.
void setState(boolean state)	Sets the state of this check box to the specified state.

java.awt

Class CheckboxGroup

Constructors :	
CheckboxGroup()	Creates a new instance of CheckboxGroup.
Methods:	

Checkbox getSelectedCheckbox()	Gets the current choice from this check box group.
void setSelectedCheckbox(Checkbox box)	Sets the currently selected check box in this group to be the specified check box.

java.awt

Class Panel

Constructors :	
Panel()	Creates a new panel using the default layout manager.
Panel(LayoutManager layout)	Creates a new panel with the specified layout manager.
Methods:	
Component add(Component comp)	Appends the specified component to the end of this container. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).
Component add(Component comp, int index)	Adds the specified component to this container at the given position. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).

java.awt

Class Container

Constructors :	
Container()	Constructs a new Container.
Methods:	
void setLayout(LayoutManager mgr)	Sets the layout manager for this container.
void setVisible(boolean b)	Shows or hides this component depending on the value of parameter b.
void setSize(int width, int height)	Resizes this component so that it has width width and height height.
Component add(Component comp)	Appends the specified component to the end of this container. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).
Component add(Component comp, int index)	Adds the specified component to this container at the given position. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).

java.awt

java.lang.Object -> java.awt.Component -> java.awt.Container -> java.awt.Window-> java.awt.Frame

Class Frame

Constructors :	
Frame()	Constructs a new instance of Frame that is initially invisible.
Frame(String title)	Constructs a new, initially invisible Frame object with the specified title.
Methods:	
void setLayout(LayoutManager mgr)	Sets the layout manager for this container.
void setVisible(boolean b)	Shows or hides this component depending on the value of parameter b.
void setSize(int width, int height)	Resizes this component so that it has width width and height height.
Component add(Component comp)	Appends the specified component to the end of this container. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).
Component add(Component comp, int index)	Adds the specified component to this container at the given position. This is a convenience method for addImpl(java.awt.Component, java.lang.Object, int).

이벤트 모델 - 주요 Event의 리스너 메소드

Linstener	메소드
ActionListener	actionPerformed(ActionEvent)

ItemListener	itemStateChanged(ItemEvent)
TextListener	textValueChanged(TextEvent)
MouseMotionListener	mouseDragged(MouseEvent), mouseMoved(MouseEvent)
MouseListener	mousePressed(MouseEvent), mouseReleased(MouseEvent) mouseEntered(MouseEvent), mouseExited(MouseEvent)
KeyListener	keyPresses(KeyEvent), keyReleased(KeyEvent) keyTyped(KeyEvent)
FocusListener	focusGained(FocusEvent), focusLost(FocusEvent)
AdjustmentListener	adjustmentValueChanged(AdjustmentEvent)
ComponentListener	componentMoved(ComponentEvent) componentHidden(ComponentEvent) componentResized(ComponentEvent) componentShown(ComponentEvent)