

2020 학년도 제 2 학기 (기말고사)시험문제					시험과목	객체지향프로그래밍및실험		담당교수	이명재
전공		학년		학번		성명		감독관 확 인	OK

※ A4용지 1장(앞뒤면사용, 반드시 손으로 작성) 허용

1. 다음은 0~99 사이의 정수를 입력받아, (벡터 numV를 이용하여) 정수에 대한 영어를 출력하는 프로그램이다. 프로그램을 완성하시오.
(주석을 참고하여 작성할 것) (총 14점)

- 1) 프로그램을 완성하시오. (주석을 참고하여 작성할 것) (10점)
- 2) ③과 ④가 있는 부분의 출력문의 결과는 각각 무엇인가? (3점)
- 3) 56이 입력되었다면, ⑥이 있는 부분의 출력문의 결과는 각각 무엇인가? (1점)

```
import java.util.*;
public class NumString {
    public static void main(String[] args) {
        ① numV = new ① (10);
        String numS[] = {"zero", "one", "two", "three", "four",
            /* 중간생략 */
            "ninty seven", "ninty eight", "ninty nine"};
        Scanner scanner = new Scanner(System.in);

        for ( int i = 0; i<numS.length; i++ )
            ② ; // 0~99 까지의 영어를 벡터에 저장함.

        System.out.println("벡터 내의 요소 객체 수 : " + ③ );
        System.out.println("벡터의 현재 용량 : " + ④ );

        System.out.print("정수(-1이 입력될 때까지)입력>> ");
        while(true) {
            int n = scanner.nextInt();
            ⑤ // 입력된 수가 -1이면 프로그램 종료
            break;
            System.out.println("숫자 " + n + "은 " + ⑥ );
        }
    }
}
```

2. 다음은 키보드에서 입력한 내용을 a.txt로 저장하는 프로그램이다. 질문에 답하시오. (총 12점)

- 1) ① ~ ⑦ 에 들어갈 코드를 작성하시오. (10점)
- 2) 입력의 끝을 나타내기 위해서 어떻게 입력해야 될까요? (2점)

```
import java.io.*;
public class CopyConA {
    public static void main(String[] args) {
        InputStreamReader isr;      FileWriter fw = null;
        int c;
        isr = ① // 키보드와 연결된 입력 문자 스트림 생성
        ② {
            fw = ③ // 파일과 연결된 출력 문자 스트림 생성
            while ( ④ ) { // 입력의 끝이 아닌동안
                ⑤ // 키보에에서 입력받은 문자를 파일에 출력
            }
            ⑥ // 닫기
        } ⑦ {
            System.out.println("입출력 오류");
        }
    }
}
```

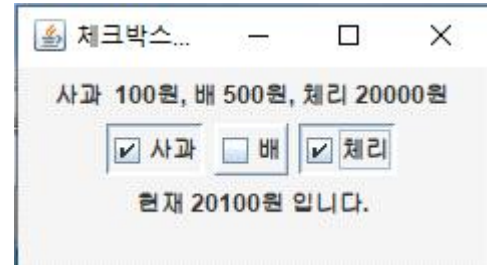
3. 다음은 실행화면의 아무 위치에 마우스 버튼을 누르면 누른 위치로 “Hello” 문자열을 옮기는 프로그램이다. 빈칸을 채우시오. (11점)

```

/* import 문 생략 */
public class MouseEx extends JFrame{
    private JLabel label = new JLabel("Hello");
    public MouseEx() {
        setTitle("Mouse");
        ① : // x 버튼 누르면 종료
        Container c = ② : // 컨테이너 얻어옴.
        ③ { // 익명 리스너 등록
            public void ④ e() { // 리스너 메서드 헤더
                int x = ⑤ : // x 좌표 얻어옴
                int y = ⑥ : // y 좌표 얻어옴
                ⑦ : // 누른 위치에 Hello 작성
            }
        };
        ⑧ : // 배치관리자 설정
        label.setSize(30,25);
        /* 기타 코드 생략 */
    }
}

```

4. 다음은 그림과 같이 세 과일에 대해 각각 체크박스를 만들고 사용자가 과일을 선택하거나 해제하면 바로 선택된 과일들의 합산 가격을 보여주는 프로그램이다. 빈칸을 채우시오. (15점)



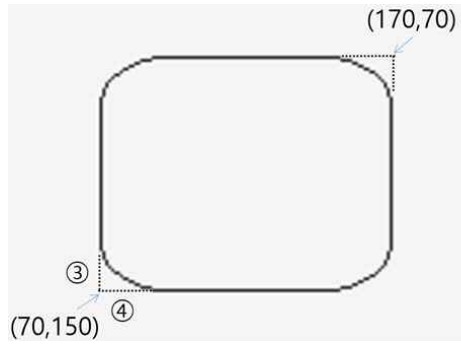
```

/* import 문 생략 */
public class CheckBoxItemEventEx4 extends JFrame {
    JCheckBox [] fruits = new JCheckBox [3]; // 체크박스 배열 0: 사과, 1: 배, 2: 체리
    /* 기타 코드 생략 */
    CheckBoxItemEventEx4() {
        /* 기타 코드 생략 */
        MyItemListener4 listener = new MyItemListener4(); // 리스너 객체 생성
        // 3개의 체크박스 생성, 컨테이너에 삽입, 리스너 등록
        for(int i=0; i<fruits.length; i++) {
            /* 기타 코드 생략 */
            ① : // 체크박스에 리스너 등록
        }
        /* 기타 코드 생략 */
    }
    // 리스너 구현
    class ② { // 리스너 클래스 헤더
        private int sum = 0; // 가격의 합
        public void ③ { // 리스너 메서드, 상태 변했을 때
            int selected=1; // 1이면 선택 상태, -1이면 해제 상태
            // 선택 상태인지 판별
            if ( ④ ) // 선택에 의한 이벤트이면
                ⑤ : // 선택 상태
            else
                ⑥ : // 선택 해제

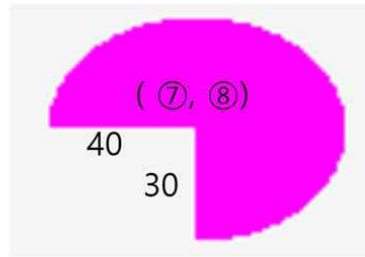
            // 어떤 체크 박스인지 판별
            if( ⑦ ) // 사과 체크박스에서 이벤트가 발생했으면
                sum += ⑧ : // 가격 계산
            /* 배, 체리 선택 체크 등 기타 코드 생략 */
        }
    }
    /* 기타 코드 생략 */
}

```

5. 다음은 Graphics의 그리기 메서드를 이용하여 작성한 프로그램에 대한 화면과 코드의 일부이다. 질문에 답하시오. (총 14점)



<그림 1>



<그림 2>

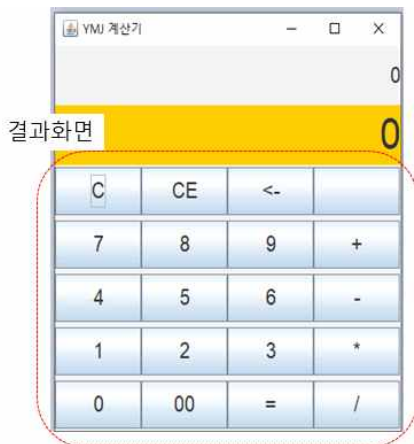
```
g.① ( ② ,50,30); // <그림 1> 관련 문제

int []x ={30,10,30,60};
int []y ={250,275,300,275};
g.drawPolygon(x, y, 3); // 2) 번 문제 관련

g.setColor(Color.MAGENTA);
g.⑤ (70, 170, ⑥ ); // <그림 2> 관련 문제
```

- 1) <그림 1>과 <그림 2>에 관련된 문제입니다. ①~⑥까지 빈칸을 채우세요. (10점)
- 2) 코드에서 `g.drawPolygon(x, y, 3);` 문장에 의해 그려지는 모양을 그리고, 꼭지점들의 좌표값을 표시하세요. (4점)

6. 아래 그림은 계산기에 대한 화면이다. 질문에 답하시오. (총 39점)



<그림 1>



<그림 2> + 버튼이 눌렸을 때

• $136 - 13 + 27$
 op1 op op2
 $123 \quad + 27$
 op1 op op2
 멤버 변수 op1, op2,
 그리고 op의 의미

<그림 3>

- 1) 다음은 결과화면(mainL)과 판넬 p를 <그림 1>과 같이 나타내기 위한 코드의 일부이다. 빈칸을 채우시오. (10점)

```
JLabel mainL; // 결과화면 멤버변수
/* 기타코드 생략 */
mainL.① : // 결과화면 폰트지정 고딕체, 크기 40
mainL.② : // 결과화면 오른쪽 정렬로 지정
mainL.③ : // 결과화면의 배경색이 보이도록 설정
mainL.④ : // 결과화면 배경색을 ORANGE로 설정
/* 기타코드 생략 */

Panel p = new Panel();
grid = ⑤ ; // p를 위한 배치관리자 객체 생성
grid.⑥ : //격자 사이의 수직 간격을 5 픽셀로 설정
⑦ : // grid를 판넬의 배치관리자로 지정
/* 기타코드 생략 */
c.⑧ : // 판넬을 컨테인트팬 c 의 중앙에 배치
```

2) 다음은 숫자 (0~9 와 00) 버튼이 눌러졌을 때 동작하는 리스너에 대한 코드이다. 빈칸을 채우시오. (12점)

```
int num = 0;      // 숫자에 대한 중간결과 보관용 멤버 변수
/* 기타코드 생략 */
class NumActionListener ① { // 숫자 버튼 클릭에 대한 리스너
    public void ② e) { // 리스너 메서드
        JButton b = ③ ; // 눌러진 버튼을 가져옴
        int btnNum = ④ ; // 버튼의 숫자를 정수 btnNum에 저장
        if ( ⑤ ) // 00 버튼이면
            num = ⑥ ; // 현재 숫자 수정
        else // 0~9 버튼 이면
            num = ⑦ ; // 현재 숫자 수정
        mainL.⑧ ; // 결과화면에 숫자 표시
    }
}
```

3) 다음은 연산자 (+-*/) 버튼이 눌러졌을 때 동작하는 리스너에 대한 코드이다. <그림 2> <그림 3>을 참조해서 빈칸을 채우시오. (11점)

```
int op1=0, op2=0; // 숫자1, 숫자2를 나타내는 멤버변수
char op = ' ' ; // 연산자를 나타내는 멤버변수
/* 기타코드 생략 */
/* 리스너 시작 부분 생략 */
JButton b = /* 눌러진 버튼을 가져오는 코드 생략 */
char opChar = ① ; // 버튼의 문자를 문자 opChar에 저장
if ( ② ) // 예) 136 에서 연산자가 눌러졌을 때
    ③ ; // 현재 숫자를 저장
else { // 예) 136 - 13 에서 연산자가 눌러졌을때
    int result = 0;
    ④ ; // 현재 숫자를 저장
    switch(op) {
        case '+': // 더하기 키
            result = ⑤ ;
        case '-': //빼기 키
            /* 기타코드 생략 */
    } // end of switch(op)
    mainL.⑥ ; // 결과화면에 중간 결과 표시
    ⑦ ; // 중간 결과 저장
}
⑧ ; // 눌러진 연산자 저장
⑨ ; // 현재 숫자를 0으로
} // 리스너 끝
```

4) 다음은 버튼이 아닌 키보드로 입력했을때 동작하는 리스너에 대한 코드이다. 빈칸을 채우시오. (6점)

```
class MyKeyListener ① { // 리스너 헤더
    public void ② e) { // 리스너 메서드 헤더, 키가 눌러졌을때
        char keyChar = ③ ; // 눌러진 키의 문자를 가져옴
        if ( ④ ) { // 눌러진 키가 0~9 사이의 숫자키이면
            /* 기타코드 생략 */
        }
    }
}
```

7. 다음은 스레드를 사용하여 빵 공장과 빵 소비자의 동기화문제를 처리하는 프로그램이다. 물음에 답하시오 (총 32 점)

생산자 생산: 빵1, 재고 1
 소비자 소비, 재고 0
 생산자 생산: 빵2, 재고 1
 생산자 생산: 빵3, 재고 2
 생산자 생산: 빵4, 재고 3
 소비자 소비, 재고 2
 생산자 생산: 빵5, 재고 3

생산자 생산: 빵6, 재고 4
 생산자 생산: 빵7, 재고 5
 소비자 소비, 재고 4
 생산자 생산: 빵8, 재고 5
 소비자 소비, 재고 4
 생산자 생산: 빵9, 재고 5
 소비자 소비, 재고 4

생산자 생산: 빵10, 재고 5
 소비자 소비, 재고 4
 소비자 소비, 재고 3
 소비자 소비, 재고 2
 소비자 소비, 재고 1
 소비자 소비, 재고 0

<결과 화면 예>

- 1) 공장은 빵을 생산하고 소비자는 빵을 소비하게 된다. 소비자는 빵이 없으면 공장에서 빵을 생산할 때까지 기다려야 한다. 비슷하게 공장도 빵의 재고가 너무 많이 쌓이면 생산을 일시 중단해야 한다. 빵 생산과 소비에 대한 것이 적절히 이루어지도록 빈칸을 채우시오. (9점)

```
class Bread {
    int stock = 0;    // 빵 재고
    int maxStock = 5;    // 빵 최대 재고

    ① public void put(int num) { // 빵 생산 메서드
        if( ② ) { // 생산이 너무 많아지면
            try {
                ③ ;
            }
            catch ( ④ ) {}
        }
        ⑤ : // 빵 재고 증가
        /* 출력문 생략 */
        ⑥ // 빵 생산 기다리는 소비자 깨움
    }

    ① public void get() { // 빵 소비 메서드
        if( ⑦ ) { // 빵이 없으면
            /* 기타 코드 생략 */
        }
    }
}
```

- 2) 다음은 공장과 소비자를 스레드로 구현한 코드이다. 기본적으로 공장은 200 ms 마다 빵을 생산하고 소비자는 0~999 ms 사이의 임의의 시간마다 빵을 소비한다. 빈칸을 채우시오. (10점)

```
class Producer extends ① { // 빵 공장 스레드,
    private Bread b;
    public Producer(Bread blank) { b = blank; }
    public void ② { // 스레드 코드
        for (int i = 1; i <= 10; i++)
            try{
                ③ : // 현재 실행 중인 스레드 양보
                ④ : // 빵 생산
            }
        /* 기타 코드 생략 */
    } // 클래스 끝

    class Consumer implements ⑤ { // 빵 소비 스레드,
        private Bread b;
        public Consumer(Bread blank) { b = blank; }
        public void ② { // 스레드 코드
            for (int i = 1; i <= 10; i++ )
                try{
                    ⑥ : // 빵 소비
                    int x = (int) ( ⑦ ); // 0~999 사이의 임의의 수 생성
                    ⑧ : // 임의의 시간 만큼 휴식
                }
            /* 기타 코드 생략 */
        } // 클래스 끝
```

(3) 다음은 빵 공장과 소비자에 대한 main 프로그램이다. 빈칸을 채우시오. (5점)

```
public class ProducerConsumer {  
    public static void main(String args[]) {  
        Bread bread = new Bread();  
        /* 공급자 관련 코드 생략 */  
        ① // 소비자 스레드 객체 c1 생성  
        ② // 소비자 스레드 c1 시작  
    }  
}
```

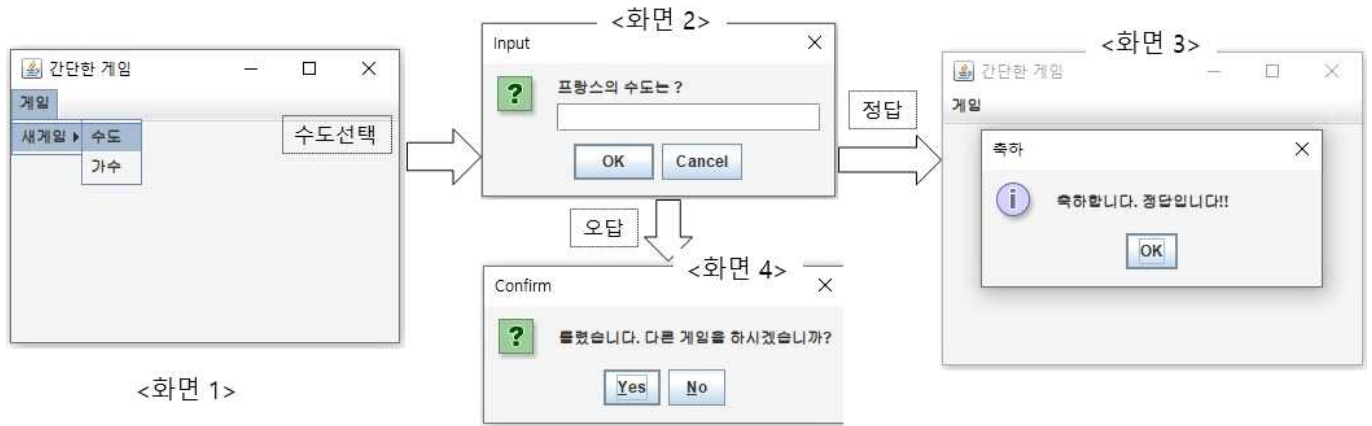
(4) 스레드는 생명 주기에 따라 6(+1)가지 상태를 갖는다. 7번 문제 코드를 사용하여 6(+1)가지 상태를 표시하여라. (8점)

(작성 예) 문제 2) ⑦ 에 의해 상태 A로, 상태 X는 문제 코드에 없음. 등

* 상태 별로 1개만 작성하면 됨.

* 문제 x) ⑦ 대신 코드를 직접 써도 됨 (예) xyz에 의해 상태 A로

8. 아래 그림은 "국가의 수도" 또는 "노래의 가수"를 맞추는 간단한 게임에 대한 화면이다. 질문에 답하시오. (총 50점)



1) readNations() 메소드는 국가와 수도가 저장되어 있는 파일 nation.txt에서 데이터를 읽어, 국가를 key, 수도를 value로 해서 멤버 변수 nations에 저장한다. readNations() 메소드를 작성하여라. (한글 인코딩은 UTF-8 사용으로 가정함) (12점)

<nation.txt 데이터 예>

대한민국 서울

프랑스 파리

미국 워싱턴

```

① nations = new ① (); // 멤버변수 nations 선언, 생성
void readNations() {
    BufferedReader br;
    InputStreamReader isr = null;
    FileInputStream fis = null;
    String input;
    // 국가와 수도가 저장되어 있는 파일 nation.txt에서 데이터를 읽어 nations에 저장하는 코드 작성
    // 위의 지역변수를 모두 사용하지 않아도 됨.
    ②
}
    
```

2) 아래 코드는 <화면 1> 과 같은 메뉴를 구성하는 코드이다. 빈칸을 채우시오. (8점)

```

void createMenu() {
    JMenuBar mb = new JMenuBar(); // 메뉴바 생성
    JMenu gameMenu = new JMenu("게임"); // 게임 메뉴 생성
    ① // 게임 메뉴 삽입
    ② // 새게임 메뉴 생성
    ③ // 수도 메뉴 생성
    ④ // 수도 메뉴 삽입
    /* 가수 메뉴 생성, 삽입 코드 생략 */
    /* 수도, 가수 메뉴에 리스너 등록 코드 생략 */
    ⑤ // 새게임 메뉴 삽입
    ⑥ // 메뉴바를 프레임에 삽입
}
    
```

3) 수도, 가수 메뉴에 등록된 리스너는 선택된 메뉴에 따라 문제 유형(1: 수도, 2: 가수)을 정하여 문제를 출제하는 setQuestion(int) 메서드를 호출한다. 리스너 코드의 빈칸을 채우시오. (5점)

```
// 아래가 리스너 코드 부분임
int type = 0; // 1: 수도, 2: 가수
String cmd = ① // 선택된 메뉴를 가져옴
② // 수도메뉴를 선택했으면,
③ // 수도 문제 유형 지정
/* 가수메뉴 선택 등의 작업 생략 */
④ // 문제 출제 메서드 호출
```

4) rightAnswer() 메서드는 사용자가 정답을 입력했을 때, <화면 3>과 같은 메시지 창을 이 게임 화면의 중앙에 출력한다. 빈칸을 채우시오. (5점)

```
void rightAnswer() { // 정답을 입력했을 때 메시지 창 출력
    ① ( ② , ③ , ④, ⑤);
}
```

5) wrongAnswer() 메서드는 사용자가 오답을 입력했을 때, <화면 4>와 같은 창을 출력한다. 빈칸을 채우시오. (4점)

```
void wrongAnswer() {
    int result = ① (null, /* 생략 */, ② );
    if ( ③ ) // 사용자가 Yes 버튼을 클릭했으면
        /* 기타 코드 생략 */
}
```

6) 다음은 문제를 출제하는 setQuestion(int) 메서드이다. 이 메소드는 문제 유형에 따라 국가-수도 문제라면, getNation() 메서드를 호출하여 문제에 사용할 국가 이름을 얻어온 후에 <화면 2>와 같이 문제를 출제한다. 사용자가 입력을 하면, 국가 이름과 사용자 입력 값으로 checkAnswer() 메서드를 사용하여 정답을 판별한다. 빈칸을 채우시오. (10점)

```
void setQuestion(int type){ // 1: 수도, 2: 가수
    String n ;
    ① { // 수도(국가) 문제이면
        ② // 문제에 사용할 국가 이름 가져옴
        String userAnswer = ③ // <화면 2>와 같이 문제 화면 표시
        /* 기타 코드 생략 */
        if ( ④ ) // 사용자 답이 정답이면
            ⑤
        else
            ⑥
    } // 기타 코드 생략 */
}
```

7) checkAnswer() 메서드는 국가 이름과 수도(사용자 입력 값임)를 매개변수로 받아 수도가 올바르게 true, 아니면 false를 리턴한다. 빈칸을 채우시오. (6점)

```
① { // checkAnswer 메서드 헤더, nation:국가, capital: 수도
    String n = ② // 국가의 수도 이름 알아내기
    if ( ③ ) // 수도가 올바르게
        return true;
    else
        return false;
}
```

/* 한 학기 동안 수고 많이 했습니다.^^

건강하고 즐겁고 의미 있는 겨울방학 보내기기를 바랍니다. */