

|   |   |   |   |
|---|---|---|---|
|   |   | 새 | 내 |
| 아 | 이 | 내 | 비 |
|   | 티 | 기 | 게 |
|   | 엠 |   | 이 |
|   |   |   | 션 |

만  
든  
이

황  
유  
정

추  
유  
림

임  
보  
영

김  
가  
경

입학식과 이튿날로 잡힌 졸업식을 전면 취소

신입생 오리엔테이션 행사 역시 취소

3월2일로 예정된 개강을 2주 미루기로 했다.

## 신종 코로나 바이러스, 대학가 덮치다

개강 연기 줄이고 졸업식 입학식 등도 잇따라 취소

차기태

승인 2020.02.04 17:34

의견 0



청주대학교 전경

[포스트저널=차기태 선임기자] 신종 코로나 바이러스 감염증(우한 폐렴)의 확산세가 드세지면서 강원대를 비롯해 전국의 대학들이 개강을 연기하거나 행사를 잇따라 취소하고 있다.

4일 대학가에 따르면 중앙대학교는 3월2일로 예정된 개강을 2주 미루기로 했다. 신입생 입학식과 오리엔테이션을 취소하는 한편, 14일로 잡혀 있던 졸업식도 8월 하계 졸업식과 통합해 진행하기로 했다.

광운대학교도 17일로 예정됐던 입학식과 이튿날로 잡힌 졸업식을 전면 취소하기로 결정했다. 총장 축하 인사말 등은 홈페이지에 게재할 방침이다.

광운대는 19일부터 2박3일 간 진행될 예정이던 신입생 오리엔테이션 행사 역시 취소했다. 개강도 3월2일에서 일주일 뒤인 같은달 9일로 연기했다.

건국대학교도 2020년도 전기 학위수여식 행사를 8월로 연기하겠다고 밝혔다. 신입생 입학식도 취소했다. 여기에 각 단과대학에 신입생 예비대학 행사 취소를 권고한 한편 신입생 및 편입생 오리엔테이션도 무기한 연기하기로 했다.

광주대학교는 신종 코로나바이러스 확산 방지를 위해 '2019학년도 학위수여식'과 '2020학년도 신입생 입학식'을 취소했다.

| 아이티엠 추천    | 아이티어 추천 |
|------------|---------|
| 학생지원 프로그램  | 학과 행사   |
| 전공 과목      | 학과 장학금  |
| 영역별 교양 추천  | 비 교과 활동 |
| 공인시험 지원금   | 재수강 제도  |
| 해외 연수 프로그램 | 연구 분야   |

# 레이블링 게임

16 Personalities

## 무료 성격유형검사

NERIS Type Explorer®

총 검사 시간은 12분 내외입니다.

혹 질문이 마음에 들지 않더라도 정직하게 답변하십시오.

가능하면 답변 시 '중립'을 선택하지 마십시오.

다른 사람들에게 자신을 소개하는 것을 어려워 합니다.

동의
비동의

0%

## 나만의 꽃 심기

나의 집중 유형을 테스트하고 나만의 꽃을 발견하세요

시작하기

POWERED BY Forest

K-test.exe

File Edit Image Layer Select

+ 자동선택: 레이어

Color Pick

## 퍼스널 컬러 테스트

Personal ColorTest

-성격편-

- 퍼티스
- 로즈버드
- 바닐라\_I
- 스위트\_P
- 네이비\_L
- 원플래밍
- 다우니
- 언덜\_L

- 페리\_60
- 오션딕스
- 오마이스
- 스프라우트
- 킬트스
- 새뮤얼안
- 엘라스\_B
- 오션베이

#FFFFFF  
COLOR 00-000

색상표 01 / 색상표 02 / 색상표 03 / 색상표 04

테스트 하러가기

현재까지 총 1,975,354명이 참여했어요.

친구에게 공유하기

링크 복사

다른 테스트 하러가기

국민악속 | 첫번째 프로젝트

## 호구 성향 테스트

"나는 사기 따위 당하지 않아"

호구가 아니라고 굳게 믿고 있는 당신.

하지만 당신도 모르는 호구 성향이 잠재돼 있었다면?

지금 바로 당신의 호구 성향을 테스트해보세요.

호구 테스트START→

다른 콘텐츠 하기 →

호구 지수

전체유형보기 +

현재 응시자

2417906 명

## MBTI?

마이어스(Myers)와 브릭스(Briggs)가 스위스의 정신분석학자인 카를 융(Carl Jung)의 심리 유형론을 토대로 고안한 자기 보고식 성격 유형 검사 도구



1-1.

ITM에 합격한 사실을 알게 되었다.  
가장 먼저 한 일은?

- (1) "동네 사람들 저 합격했어요!"하고 여기저기 알린다.
- (2) 친한 친구들 혹은 가족들에게 이 사실을 바로 전한다.
- (3) 혼자 조용히 내적 환호를 한다.
- (4) 합격 사실을 알기 전과 감정의 변화가 크지 않다.

1-2.

학교생활 중 알고 싶은 정보가 생겼다.  
내가 할 행동은?

- (1) 학과 사무실에 전화해서 모르는 정보를 물어보자!
- (2) 교수님이라면 아실 것 같은데 메일을 보내볼까?
- (3) 익명으로 에브리타임에 글을 남겨봐야겠다.
- (4) 학교 홈페이지에 있을 것 같은데 찾아보자!

1-3.

조별 과제를 하게 되었다.  
내가 맡고 싶은 역할은?

- (1) 조별 과제에서는 역시 발표지!
- (2) 발표에 꼭 있어야 하는 PPT 내가 만들어주지!
- (3) 자료 수집이야 말로 내가 해야 할 일!
- (4) 발표만 빠면 뭐든 상관 없어요.

2-1.

동기가 수업 직전 과제를 보여 달라  
할 때, 나의 행동은?

- (1) 뭐? 과제? 나도 안했는데?ㅋㅋ
- (2) 옆에 있던 공부 잘하는 친구에게 물어본다.
- (3) 우리 과제 있었음? 과제가 있는지 진짜 몰랐다.
- (4) 인생에서 과제보다 중요한 것들이 얼마나 많은데!

2-2.

당신은 대학교 4학년, 취준생이다.  
당신의 졸업 후 계획은?

- (1) 나는 아무고토못하는감자...
- (2) 대학원가서 실력을 기른 다음 취업하겠어!
- (3) 인턴십으로 경험을 쌓고 취업해도 늦지 않지ㅎㅎ
- (4) 대학원은 무슨! 취업해서 빨리 돈벌자!

2-3.

ITM전공 1학년들은 교수님과 함께  
영어 튜터링을 한다. 당신이 원하는  
수업은?

- (1) 매주 영단어 시험을 보고 싶어요!
- (2) 넷플릭스 시리즈를 보고 함께 얘기해보는건  
어떨까요?
- (3) 같이 영어로 토론을 해보는건 어떨까요?
- (4) 이번에는 스페인어를 배우는 건 어때요? 라고  
말하고 후회..



## F / T 유형 문항

3-1.

친구가 "이번에 C+ 나올 것 같아.."  
라고 할 때 나의 반응은?

- (1) 이번에 많이 어려웠나보네ㅠㅠ 열심히 했으니까 잘 나올거야!
- (2) 괜찮을 거야!
- (3) 시험 어땠는데?.
- (4) 몇 학점 짜리야? 재수강하자!

3-2.

친구가 요즘 취업 준비에 한창이다.  
친구에게 무슨 말을 할까?

- (1) (자세히 물어보지는 말자!) 요즘에 많이 힘들징..?
- (2) 취준은 잘 하고 있어?
- (3) 취준 어떻게 하고 있어?
- (4) 요즘에 준비 뭐하고 있어? 이력서 넣어 봤어?

3-3.

오늘은 IELTS 결과가 나오는 날!  
친구가 5.5를 못 넘어 슬퍼한다.

- (1) 다음번엔 꼭 넘을거야!!! 슬퍼하지마..
- (2) 괜찮아 다음번에 잘하면 되징!
- (3) 지금 몇 번째 보는 거야?
- (4) 이번이 처음이야? 지원금 또 나오니까 한 번 더 보장~





## P / J 유형 문항

4-1.

ITM에 합격한 사실을 알게 되었다.  
내가 세운 입학 후 계획은?

- (1) 몰라! 새내기니까 계획 없이 놀거야!
- (2) 흠.. 영어 공부나 코딩 공부 좀 할까...?
- (3) 올해 안에 꼭 아이엘츠 시험 봐야지!
- (4) 5월까지 A도 하고 B도 하고 또..이번학기에 O/O까지 끝내고...

4-2.

전공 시험이 벌써 한달 앞으로 다가  
왔다. 내가 선호하는 공부법은?

- (1) 아직 시간 많이 남았네~~ 몰라 일단 놀아 놀아!!
- (2) 한달이면 충분하지! 한 2주 전부터 시작해 볼까나?
- (3) 시험 범위가 총 8차시니까 일주일에 2차시 씩 공부 해야징.
- (4) A파트가 어려웠으니 전 파트들을 10월까지 끝내야겠다.

4-3.

ITM은 다양한 직무로 취업할 수 있다.  
내가 선호하는 스타일의 직업은?

- (1) 업무 시공간을 내 맘대로! 프리랜서
- (2) 자유롭고 유연성을 지닌 개발자 직무
- (3) 정년이 보장되어 안정적인 공무원
- (4) 조직적이고 업무 지향적인 IT 관리자



## 성향 분류 기준

### - 문항별 배점

보기 (1) : 1점 | 보기 (2) : 2점 | 보기 (3) : 3점 | 보기 (4) : 4점

### - 합계 별 분류

| 3 ~ 7점 | 8 ~ 12점 |
|--------|---------|
| 외향 (E) | 내향 (I)  |
| 감각 (S) | 직관 (N)  |
| 감정 (F) | 사고 (T)  |
| 인식 (P) | 판단 (J)  |

📌 전달할 정보 분류

| 아이티엠 추천               | 아이티어 추천           |          |
|-----------------------|-------------------|----------|
| 학생지원 프로그램             | 학과 행사             | E / I 유형 |
| 전공 과목                 | 학과 장학금            | S / N 유형 |
| 영역별 교양 추천<br>공인시험 지원금 | 비 교과 활동<br>재수강 제도 | F / T 유형 |
| 해외 연수 프로그램            | 연구 분야             | P / J 유형 |

## - 활용한 자바 특강 내용

프로그램 GUI : 자바 swing, ArrayList, for 문

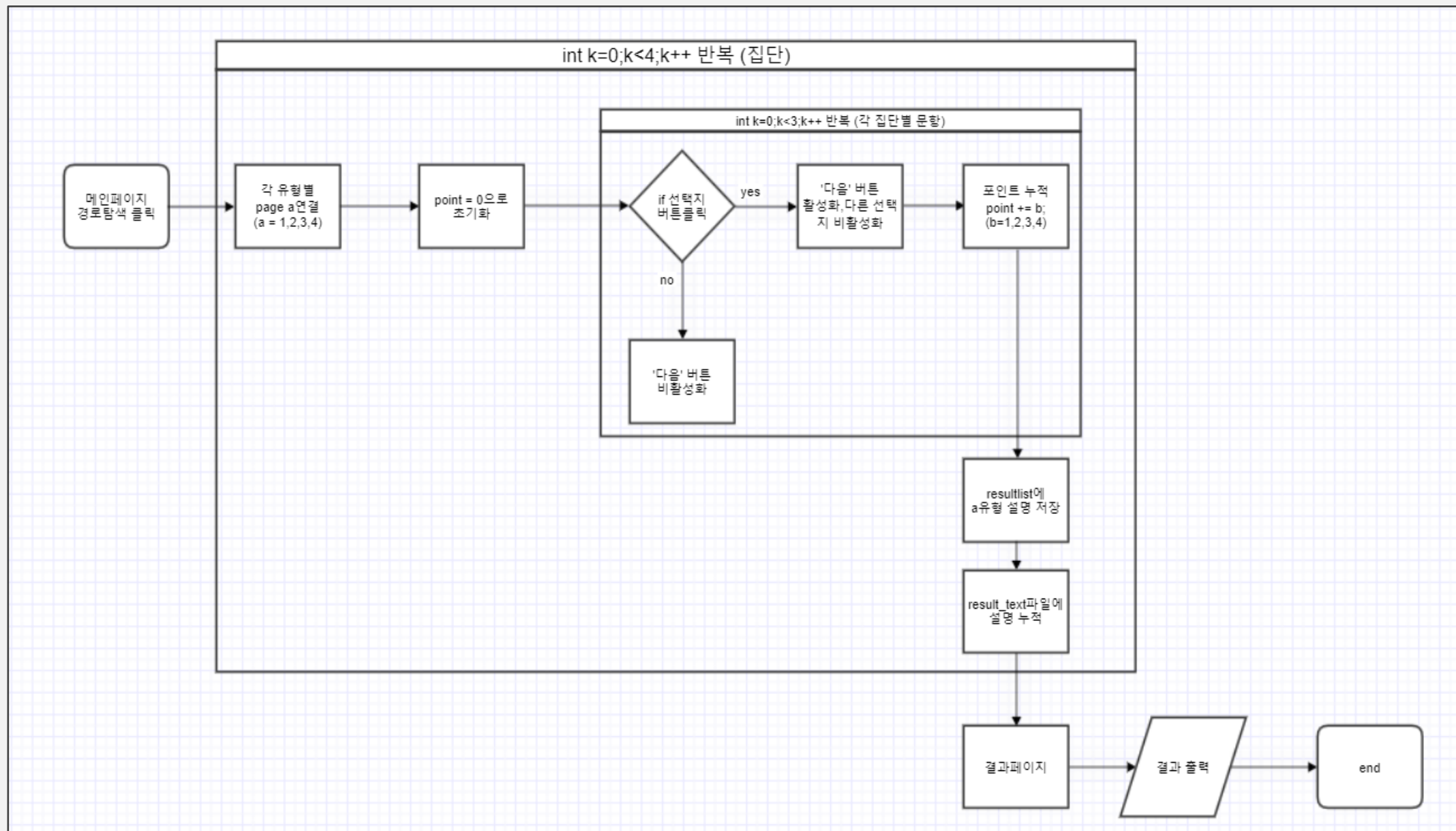
선택하는 문항별 점수 계산 : if 구문

성향에 따른 결과 제공 : 파일 입출력

### ☒ 상세 커리큘럼

| 일차 | 교시 | 내용  | 비고                             |
|----|----|---|--------------------------------|
| 3  | 1  | -조건문 if의 구조와 용법   | -조건문 실행 결과 체크<br>-반복문 실행 결과 체크 |
|    | 2  | -조건문 switch의 구조와 용법   |                                |
|    | 3  | -반복문 for, while의 구조와 용법                                     |                                |
|    | 4  | -조건문의 중첩과 반복문의 중첩   |                                |
| 6  | 1  | -인터페이스를 작성하는 목적<br>-추상 클래스와 인터페이스                           | -인터페이스 구현 결과 체크                |
|    | 2  | -인터페이스를 사용한 다형성 구현<br>-인터페이스를 구현한 클래스                       |                                |
|    | 3  | -패키지의 목적 및 선언방법   |                                |
|    | 4  | -import문을 사용한 패키지 사용  |                                |
| 10 | 1  | -자바 입력과 구조<br>-Stream의 종류                                   | -생성 파일 체크<br>-파일목록 출력 결과 체크    |
|    | 2  | -FileInputStream을 사용한 파일 읽기<br>-FileOutputStream을 사용한 파일 저장 |                                |
|    | 3  | -파일목록과 디렉토리 목록<br>-파일상태값 확인하기                               |                                |
|    | 4  | -파일생성 및 삭제<br>-디렉토리 생성 및 삭제                                 |                                |

## 📌 프로그램 순서도



```
private static int point;{
    point=0;
}
private static String result;{
    result="";
}

private static List<String> resultlist = new ArrayList<String>();//결과 리스트

String path = practice.class.getResource("").getPath();
```

```
File I_result = new File(path + "txtFile/1)I_results.txt");
File E_result = new File(path + "txtFile/1)E_results.txt");
File N_result = new File(path + "txtFile/2)N_results.txt");
File S_result = new File(path + "txtFile/2)S_results.txt");
File F_result = new File(path + "txtFile/3)F_results.txt");
File T_result = new File(path + "txtFile/3)T_results.txt");
File J_result = new File(path + "txtFile/4)J_results.txt");
File P_result = new File(path + "txtFile/4)P_results.txt");
File result_text = new File(path + "txtFile/textresult.txt");
```

각 유형별 설명이  
담긴 텍스트 파일

결과를 입력 받을 txt 파일

```
private void initialize() {  
    frame = new JFrame();  
    frame.setBounds(100, 100, 600, 800);  
    frame.setLocationRelativeTo(null);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.getContentPane().setLayout(null);  
}
```

패널의 배경에 이미지를 넣기 위해  
ImageIcon이용

```
ImageIcon icon;  
icon = new ImageIcon(practice.class.getResource("/image/background1.png"));
```

// 메인페이지


```
JPanel q0 = new JPanel() {  
    public void paintComponent(Graphics g) {  
        g.drawImage(icon.getImage(), 0, 0, null);  
        setOpaque(false);  
        super.paintComponent(g);  
    }  
};
```

패널의 배경에 이미지 생성


```
q0.setBackground(Color.BLACK);  
q0.setBounds(0, 0, 600, 800);  
frame.getContentPane().add(q0);
```

유형 : 4개  
 각 유형별 문제 수 : 3개  
 각 문제 선지 수 : 4개

|       | 1번 선지 | 2번 선지 | 3번 선지 | 4번 선지 |
|-------|-------|-------|-------|-------|
| 1번 문항 | 1-1   | 1-2   | 1-3   | 1-4   |
| 2번 문항 | 2-1   | 2-2   | 2-3   | 2-4   |
| 3번 문항 | 3-1   | 3-2   | 3-3   | 3-4   |

 Button[3][4]

```
JButton[][] btnNewButton_1 = new JButton[3][4];
```

 각 문제들의 선지(?) 버튼들을 배열로 선언

```
btnNewButton_1[0][0] = new JButton("(1) “ 동네 사람들 저 합격했어요! ” 하고 여기저기 알린다.");
```

```


btnNewButton_1[0][0].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnNewButton_1[0][0])
        {
            for(int k=0;k<4;k++) {
                btnNewButton_1[0][k].setEnabled(false);
            }
        }
    }
});

```

```

q1_1.setLayout(null);
q1_1.add(btnNewButton_1[0][0]);

```

 선지를 중복 선택하는 경우를 막기 위해 선지 하나를 선택하면 나머지 버튼들은 비활성화가 되도록 함.



```

txtItm = new JTextField();
txtItm.setOpaque(false);
txtItm.setBounds(37, 209, 506, 57);
txtItm.setForeground(Color.WHITE);
txtItm.setBackground(null);
txtItm.setFont(new Font("Mapo꽃섬", Font.BOLD, 28));
txtItm.setHorizontalAlignment(SwingConstants.LEFT);
txtItm.setText("가장 먼저 한 일은?");
q1_1.add(txtItm);
txtItm.setColumns(10);
txtItm.setBorder(javax.swing.BorderFactory.createEmptyBorder());

textField_2 = new JTextField();
textField_2.setOpaque(false);
textField_2.setBounds(37, 105, 506, 57);
textField_2.setForeground(Color.WHITE);
textField_2.setBackground(null);
textField_2.setText("1-1.");
textField_2.setHorizontalAlignment(SwingConstants.LEFT);
textField_2.setFont(new Font("Mapo꽃섬", Font.BOLD, 28));
textField_2.setColumns(10);
q1_1.add(textField_2);
textField_2.setBorder(javax.swing.BorderFactory.createEmptyBorder());

txtItm_1 = new JTextField();
txtItm_1.setBounds(37, 156, 516, 57);
txtItm_1.setOpaque(false);
txtItm_1.setForeground(Color.WHITE);
txtItm_1.setBackground(null);
txtItm_1.setText("ITM에 합격한 사실을 알게 되었다.");
txtItm_1.setHorizontalAlignment(SwingConstants.LEFT);
txtItm_1.setFont(new Font("Mapo꽃섬", Font.BOLD, 28));
txtItm_1.setColumns(10);
q1_1.add(txtItm_1);
txtItm_1.setBorder(javax.swing.BorderFactory.createEmptyBorder());

```

1-1.

ITM에 합격한 사실을 알게 되었다.

가장 먼저 한 일은?

(1) "동네 사람들 저 합격했어요!"하고 여기저기 알린다.

(2) 친한 친구들 혹은 가족들에게 이 사실을 바로 전한다.

(3) 혼자 조용히 내적 환호를 한다.

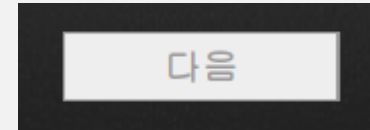
(4) 합격 사실을 알기 전과 감정의 변화가 크지 않다.

다음

```
//버튼
JButton btnNewButton_1_1_6 = new JButton("다음");
btnNewButton_1_1_6.setEnabled(false);
btnNewButton_1_1_6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        q1_2.setVisible(true);
        q1_1.setVisible(false);
    }
});
btnNewButton_1_1_6.setBounds(259, 680, 91, 23);
q1_1.add(btnNewButton_1_1_6);
```

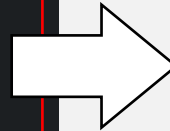


- 선지를 선택하기 전에는 비활성화 상태가 되도록 설정함.(setEnabled)
- 다음 버튼을 클릭하면 화면이 전환됨.



```
JButton btnNewButton_1_2_6 = new JButton("다음");
btnNewButton_1_2_6.setEnabled(false);
btnNewButton_1_2_6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        q1_3.setVisible(true);
        q1_2.setVisible(false);
    }
});
btnNewButton_1_2_6.setBounds(259, 680, 91, 23);
q1_2.add(btnNewButton_1_2_6);
```

```
for(int p=0;p<4;p++) {
    btnNewButton_1[0][p].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            btnNewButton_1_1_6.setEnabled(true);
        }
    });
    btnNewButton_1[1][p].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            btnNewButton_1_2_6.setEnabled(true);
        }
    });
}
```



- 선지를 선택하면 다음 버튼이 활성화됨
- 반복문을 활용하여 각 문제의 4개의 선지들을 설정함.



## 사용자 유형 결정

```
JButton next2_3 = new JButton("\uB2E4\uC74C");
next2_3.setEnabled(false);
next2_3.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        if (point >= 8) {
            try (BufferedReader br = new BufferedReader(new FileReader(N_result))){
                String line;
                while((line=br.readLine()) != null) {
                    System.out.println(line);
                    resultlist.add(line);
                }
            }catch (IOException a){
                a.printStackTrace();
            }
            result += "당신은 N입니다.\n";
        }
        else {
            result += "당신은 S입니다.\n";
            try (BufferedReader br = new BufferedReader(new FileReader(S_result))){
                String line;
                while((line=br.readLine()) != null) {
                    System.out.println(line);
                    resultlist.add(line);
                }
            }catch (IOException a){
                a.printStackTrace();
            }
        }
        q3_1.setVisible(true);
        q2_3.setVisible(false);
        point=0;
    }
});
```

다른 유형으로 넘어갈 때  
(ex : A집단(I vs E)->B집단(N vs S))  
포인트를 초기화 함.

Point 값에 따라 유형이 결정 됨.  
이때, bufferedReader를 활용해  
각 유형별 설명이 담긴 텍스트 파일을 불러와  
resultlist에 arraylist 형식으로 저장함

```
private static int point;{
    point=0;
}
private static String result;{
    result="";
}
```

point 변수, result 변수 설정 및 초기화

```
for(int i =0;i<3;i++){
    btnNewButton_2[i][0].setBounds(60, 338, 480, 60);
    btnNewButton_2[i][0].setFont(new Font("Mapo꽃섬", Font.PLAIN, 17));
    btnNewButton_2[i][0].setForeground(Color.WHITE);
    btnNewButton_2[i][0].setContentAreaFilled(false);
    btnNewButton_2[i][0].setBorder(new LineBorder(Color.WHITE, 1));
    btnNewButton_2[i][0].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 1;
            System.out.println(point);
            btnNewButton_2[0][0].setBackground(Color.yellow);
        }
    });

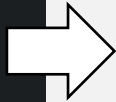
    btnNewButton_2[i][1].setBounds(60, 418, 480, 60);
    btnNewButton_2[i][1].setFont(new Font("Mapo꽃섬", Font.PLAIN, 17));
    btnNewButton_2[i][1].setForeground(Color.WHITE);
    btnNewButton_2[i][1].setContentAreaFilled(false);
    btnNewButton_2[i][1].setBorder(new LineBorder(Color.WHITE, 1));
    btnNewButton_2[i][1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 2;
            System.out.println(point);
        }
    });

    btnNewButton_2[i][2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 3;
            System.out.println(point);
        }
    });

    btnNewButton_2[i][2].setBounds(60, 499, 480, 60);
    btnNewButton_2[i][2].setFont(new Font("Mapo꽃섬", Font.PLAIN, 17));
    btnNewButton_2[i][2].setForeground(Color.WHITE);
    btnNewButton_2[i][2].setContentAreaFilled(false);
    btnNewButton_2[i][2].setBorder(new LineBorder(Color.WHITE, 1));

    btnNewButton_2[i][3].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 4;
            System.out.println(point);
        }
    });

    btnNewButton_2[i][3].setBounds(60, 579, 480, 60);
    btnNewButton_2[i][3].setFont(new Font("Mapo꽃섬", Font.PLAIN, 17));
    btnNewButton_2[i][3].setForeground(Color.WHITE);
    btnNewButton_2[i][3].setContentAreaFilled(false);
    btnNewButton_2[i][3].setBorder(new LineBorder(Color.WHITE, 1));
}
```



각 버튼들의 공통되는 속성들(위치, 폰트, 클릭 이벤트, 색상 등등)을 반복문을 활용하여 설정함.

|     |     |     |     |
|-----|-----|-----|-----|
| 1-1 | 1-2 | 1-3 | 1-4 |
| 2-1 | 2-2 | 2-3 | 2-4 |
| 3-1 | 3-2 | 3-3 | 3-4 |

보기를 같은 번호끼리 묶어 각 선지 별 배점을 할당함.

이 점수는 사용자의 유형을 결정할 때 쓰임.



```
for(int i=0;i<3;i++){
    btnNewButton_2[i][0].setBounds(60, 338, 480, 600);
    btnNewButton_2[i][0].setFont(new Font("Mapo꽃섬"));
    btnNewButton_2[i][0].setForeground(Color.WHITE);
    btnNewButton_2[i][0].setContentAreaFilled(false);
    btnNewButton_2[i][0].setBorder(new LineBorder(Color.BLACK));
    btnNewButton_2[i][0].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 1;
            System.out.println(point);
            btnNewButton_2[0][0].setBackground(Color.GREEN);
        }
    });

    btnNewButton_2[i][1].setBounds(60, 418, 480, 600);
    btnNewButton_2[i][1].setFont(new Font("Mapo꽃섬"));
    btnNewButton_2[i][1].setForeground(Color.WHITE);
    btnNewButton_2[i][1].setContentAreaFilled(false);
    btnNewButton_2[i][1].setBorder(new LineBorder(Color.BLACK));
    btnNewButton_2[i][1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 2;
            System.out.println(point);
        }
    });

    btnNewButton_2[i][2].setBounds(60, 499, 480, 600);
    btnNewButton_2[i][2].setFont(new Font("Mapo꽃섬"));
    btnNewButton_2[i][2].setForeground(Color.WHITE);
    btnNewButton_2[i][2].setContentAreaFilled(false);
    btnNewButton_2[i][2].setBorder(new LineBorder(Color.BLACK));
    btnNewButton_2[i][2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 3;
            System.out.println(point);
        }
    });

    btnNewButton_2[i][3].setBounds(60, 579, 480, 600);
    btnNewButton_2[i][3].setFont(new Font("Mapo꽃섬"));
    btnNewButton_2[i][3].setForeground(Color.WHITE);
    btnNewButton_2[i][3].setContentAreaFilled(false);
    btnNewButton_2[i][3].setBorder(new LineBorder(Color.BLACK));
    btnNewButton_2[i][3].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            point += 4;
            System.out.println(point);
        }
    });
}
```

1-1.

ITM에 합격한 사실을 알게 되었다.

가장 먼저 한 일은?

(1) “동네 사람들 저 합격했어요!”하고 여기저기 알린다.

(2) 친한 친구들 혹은 가족들에게 이 사실을 바로 전한다.

(3) 혼자 조용히 내적 환호를 한다.

(4) 합격 사실을 알기 전과 감정의 변화가 크지 않다.

위치, 폰트, 클릭 이벤트, 색상 등등)을

|  |     |     |
|--|-----|-----|
|  | 1-3 | 1-4 |
|  | 2-3 | 2-4 |
|  | 3-3 | 3-4 |

각 선지 별 배점을 할당함.

결정할 때 쓰임.



다음



결과창으로 넘어가는 다음 버튼

```
JButton next4_3 = new JButton("\uB2E4\uC74C");
next4_3.setEnabled(false);
next4_3.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        if (point >= 8) {
            result += "당신은 J입니다.\n";
            try (BufferedReader br = new BufferedReader(new FileReader(J_result))){
                String line;
                while((line=br.readLine()) != null) {
                    System.out.println(line);
                    resultlist.add(line);
                }
            }catch (IOException a){
                a.printStackTrace();
            }
        }
        else {
            result += "당신은 P입니다.\n";
            try (BufferedReader br = new BufferedReader(new FileReader(P_result))){
                String line;
                while((line=br.readLine()) != null) {
                    System.out.println(line);
                    resultlist.add(line);
                }
            }catch (IOException a){
                a.printStackTrace();
            }
        }
        System.out.println(result);
        q4_3.setVisible(false);
        System.out.println(resultlist.get(8));
        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(result_text));
            for(int b=0;b<resultlist.size();b++) {
                writer.write(resultlist.get(b));
                writer.write("\r\n");
            }
            writer.close();
        } catch (IOException k) {
            k.printStackTrace();
        }
        q5_1.setVisible(true);
        //result_screen
```

결과창으로 넘어갈 때 BufferedWriter를 통해  
resultlist에 저장한 각 유형별 설명들을  
result\_text 파일(textresult.txt)에 저장함!  
이때, 줄바꿈(WrWn) 하면서 파일을 작성함.

```
JTextArea txtItm_r = new JTextArea();
txtItm_r.setLineWrap(true);
txtItm_r.setCaretPosition(txtItm_r.getDocument().getLength());
txtItm_r.setOpaque(false);
txtItm_r.setBounds(37, 130, 516, 900);
txtItm_r.setForeground(Color.WHITE);
txtItm_r.setBackground(null);

try (BufferedReader br = new BufferedReader(new FileReader(result_text))) {
    String str;
    while((str=br.readLine()) != null) {
        txtItm_r.append(str+"\r\n");
        txtItm_r.append("\r\n");
    }
    br.close();
} catch (IOException a) {
    a.printStackTrace();
}

txtItm_r.setFont(new Font("Mapo꽃섬", Font.PLAIN, 17));
txtItm_r.setColumns(10);
q5_1.add(txtItm_r);
txtItm_r.setBorder(javax.swing.BorderFactory.createEmptyBorder());
// 결과창

q5_1.setLayout(null);
q5_1.add(txtItm_r);

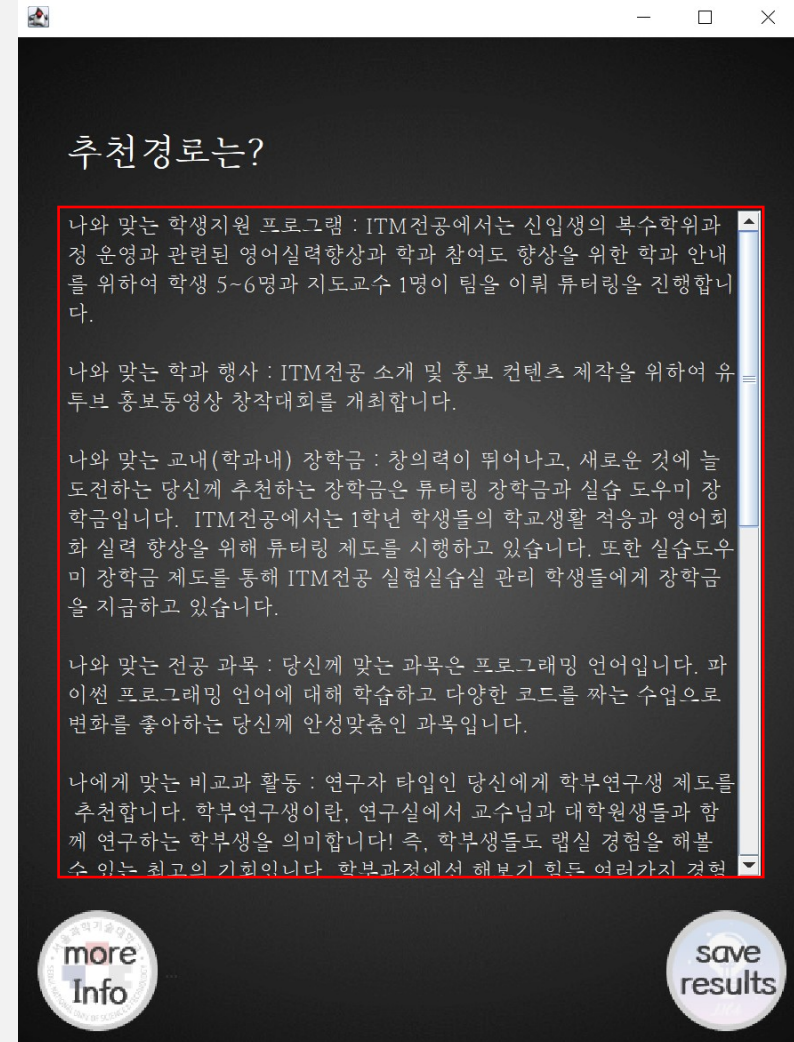
txtItm_1 = new JTextField();
txtItm_1.setOpaque(false);
txtItm_1.setBounds(37, 57, 516, 57);
txtItm_1.setForeground(Color.WHITE);
txtItm_1.setBackground(null);
txtItm_1.setText("추천경로는?");
txtItm_1.setHorizontalAlignment(SwingConstants.LEFT);
txtItm_1.setFont(new Font("Mapo꽃섬", Font.BOLD, 28));
txtItm_1.setColumns(10);
q5_1.add(txtItm_1);
txtItm_1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
}
```

BufferedReader를 이용해 textresult.txt(result\_text)파일에 저장된 결과 텍스트를 불러와서 textarea인 txtItm\_r에 입력함.

```
1 JScrollPane scrollPane_1 = new JScrollPane(txtItm_r,  
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);  
scrollPane_1.setBounds(37, 130, 520, 500);  
scrollPane_1.setVisible(true);  
2 scrollPane_1.setOpaque(false);  
scrollPane_1.getViewport().setOpaque(false);  
3 scrollPane_1.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
q5_1.add(scrollPane_1);
```



- 1 결과가 출력되는 textarea에 수직 스크롤을 추가함
- 2 scrollPane\_1의 테두리와 배경을 투명하게 설정함
- 3 scrollPane\_1을 결과창(q5\_1)에 추가함





```
JButton web = new JButton("website");
web.setIcon(new ImageIcon(practice.class.getResource("/image/학과.png")));
web.setBounds(11, 654, 112, 92);
web.setBorderPainted(false);
web.setContentAreaFilled(false);
web.setFocusPainted(false);
web.setOpaque(false);
web.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{
            Desktop.getDesktop().browse(new URI("https://itm.seoultech.ac.kr/"));
        }catch (IOException a) {
            a.printStackTrace();
        }
        catch (URISyntaxException a) {
            a.printStackTrace();
        }
    }
});

q5_1.add(web);
```



를 버튼으로 추가하고

클릭할 경우 ITM 학과 홈페이지로  
연결되도록 함

```
String saveFilePath = "D:/";
String saveFileName = "your_results";
String saveTxtFileExt = ".txt";

Dimension size = q5_1.getSize();
BufferedImage image = new BufferedImage(
    size.width, size.height,
    BufferedImage.TYPE_INT_RGB);
Graphics2D g2 = image.createGraphics();
q5_1.paint(g2);

JButton save = new JButton("saveResults");
save.setIcon(new ImageIcon(practice.class.getResource("/image/결과.png")));
save.setBounds(483, 654, 112, 92);
save.setBorderPainted(false);
save.setContentAreaFilled(false);
save.setFocusPainted(false);
save.setOpaque(false);

save.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            File saveresult = new File(saveFilePath+saveFileName+"."+saveTxtFileExt);
            BufferedWriter writer = new BufferedWriter(new FileWriter(saveresult));
            for(int b=0;b<resultlist.size();b++) {
                writer.write(resultlist.get(b));
                writer.write("\r\n");
            }
            writer.close();
        } catch (IOException k) {
            k.printStackTrace();
        }
    }
});

q5_1.add(save);
```



를 버튼으로 추가하고

클릭할 경우 테스트 결과가  
txt 파일로 사용자 컴퓨터에  
저장되도록 함



## 프로그램 시연

실제 작동 모습



## 기대 효과 및 활용 방안

ITM에 입학하는 새내기들에게 학과에 대한 중요 정보들을 공유하고 성향에 따라 과 내 활동 방향을 추천해줌으로써, 입학생들의 적성에 대한 불안감을 해소하고 ITM에 대한 정보 전달이 가능하다. 또한 학과 이해도가 높아짐

