

MBT1조

빅데이터 분석가 양성과정 2기 1주차

간단 MBTI 테스트하기



목차



실시간 급상승



- | | | |
|---|---------|------|
| 1 | 프로젝트 설명 | ↑ 56 |
| 2 | 프로젝트 설계 | ↑ 45 |
| 3 | 실제 화면 | ↑ 33 |
| 4 | Q&A | ↑ 19 |



프로젝트 설명

주제 선정 이유



MBTI의 인기

MBTI에 대한 관심은 사그라들 기미가 보이지 않는다. 어색한 첫만남 대화에서 아이스브레이킹을 위한 스몰토크 주제로 활용되는가 하면, 어떤 기업에서는 구인란에 MBTI 항목을 넣기도 한다.



너무 오래 걸리고 귀찮아!

빠르고 간단한 MBTI 테스트!



간단한 MBTI 테스트, MBTI 유형별 설명 & MBTI 유형별 궁합, 수강생 MBTI 궁합

프로젝트 설계





프로젝트 설계



프로그램 기능



순서도



데이터 구성



코드 구성



GUI 흐름도

1

MBTI 테스트

MBTI 테스트 진행

12가지의 간단한 이지선다 질문 답변



MBTI 테스트 결과 출력



유형별 상세설명으로 이동

2

MBTI 상세 설명

MBTI 유형 선택



간단한 설명 출력

3

MBTI 궁합

MBTI 2가지 선택



궁합 출력

MBTI 궁합표 활용

4

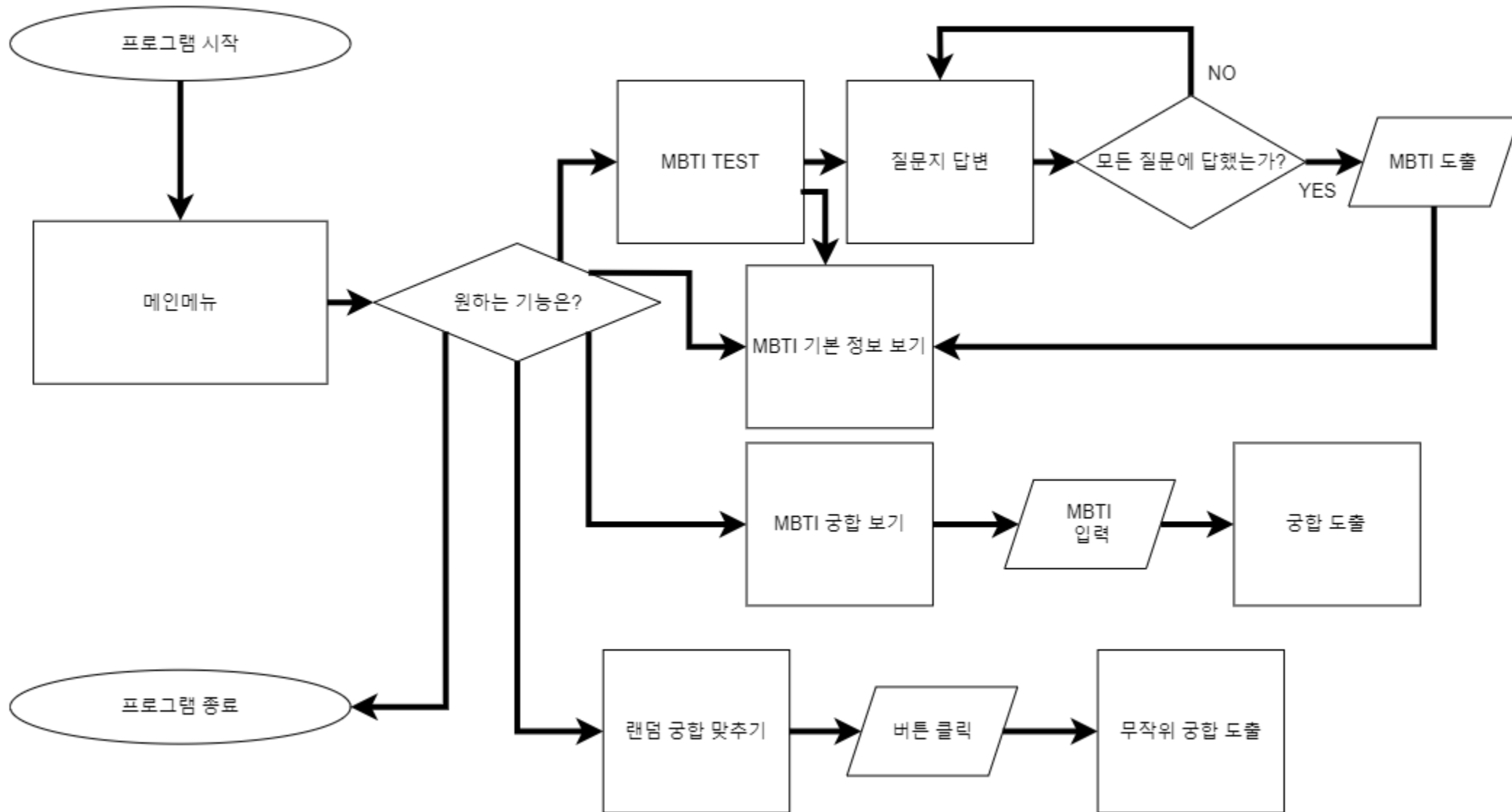
수강생 MBTI 궁합

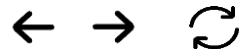
수강생 정보 입력

공란은 random으로 처리



두 사람의 궁합 출력





https://프로그램 데이터 구성.com



MBTI 테스트 질문

유형별 설명

BTI 궁합 점수표

궁합 결과별 사진

강생 MBTI 정보

questions.txt ×

```
1 <title> Extraversion (외향) / Introversion (내향) </title>
2 에너지의 방향
3
4 1. 친구와의 약속이 취소되었을 때 나는?
5 뭐야.. 다른 친구에게 연락해야해서 놀자고 한다. vs 오 ㅋㅋ 혼자서 집에서 뒹굴뒹굴 거린다.
6
7 2. 오랜만에 정말 맛있는 음식점을 찾았을 때 나는?
8 야야, 내가 맛있는 집을 아는데 꼭 한 번 가자! 주변사람들에게 적극적으로 추천한다. vs 오 이집 진짜 맛집이네! 혼자 생각
9
10 3. 조별과제를 할 때 나는?
11 무임승차 못참아! 참여하지 않은 인원을 제명하거나 건의한다. vs 사정이 있겠지... 한 번 참고 봐준다.
12
13
14 <title> Sencing (감각) / INTuituin (직관) 지구멸망?</title>
15 정보를 수집하거나 무엇을 인식하는 부분
16
17 4. 예전에 한 번 먹어본 음식에 대해서 나는?
```

▼ mbti_info

- ENFJ.txt
- ENFP.txt
- ENTJ.txt
- ENTP.txt
- ESFJ.txt
- ESFP.txt
- ESTJ.txt
- ESTP.txt
- INFJ.txt
- INFP.txt
- INTJ.txt
- INTP.txt
- ISFJ.txt
- ISFP.txt
- ISTJ.txt
- ISTP.txt

ENFJ.txt x

1 “정의로운 사회운동가”
2
3 사회운동가형 사람은 카리스마와 충만한 열정을 지닌 타고난 리더형입니다. 인구의 대략 2%가 이 우
4
5 우리는 대개 강직한 성품을 가진 이에게 마법처럼 끌리곤 합니다. 사회운동가형 사람은 진정으로 타
6

▼ relations

- Relations_0.png
- Relations_1.png
- Relations_2.png
- Relations_3.png
- Relations_-1.png



mbti_mate.csv



mbti_student.csv



mbti_mate.csv X

```
1  "", "ENTJ", "ENTP", "INTJ", "INTP", "ESTJ", "ESFJ", "ISTJ", "ISFJ", "ENFJ", "ENFP", "INFJ", "INFP", "ESTP", "ESFP", "ISTP"
2  "ENTJ", "-1", "0", "1", "1", "0", "-1", "0", "-1", "0", "1", "1", "3", "2", "2", "1", "3"
3  "ENTP", "0", "2", "1", "1", "2", "2", "3", "3", "1", "0", "1", "1", "0", "-1", "0", "-1"
4  "INTJ", "1", "1", "1", "0", "0", "-1", "1", "-1", "1", "1", "1", "2", "3", "3", "1", "2"
5  "INTP", "1", "1", "0", "1", "2", "3", "1", "3", "3", "1", "2", "0", "0", "-1", "0", "-1"
6  "ESTJ", "0", "2", "0", "2", "1", "0", "1", "1", "-1", "2", "-1", "3", "-1", "1", "1", "3"
7  "ESFJ", "-1", "2", "-1", "3", "0", "1", "1", "1", "1", "2", "0", "1", "1", "0", "3", "1"
8  "ISTJ", "0", "3", "1", "1", "1", "1", "1", "0", "-1", "3", "-1", "2", "1", "1", "0", "2"
9  "ISFJ", "-1", "3", "-1", "3", "1", "1", "0", "0", "0", "3", "0", "1", "1", "1", "2", "-1"
10 "ENFJ", "0", "1", "1", "3", "-1", "1", "-1", "0", "2", "0", "1", "1", "2", "2", "3", "1"
11 "ENFP", "1", "0", "1", "1", "2", "2", "3", "3", "0", "1", "1", "0", "-1", "-1", "-1", "1"
12 "INFJ", "1", "1", "1", "2", "-1", "0", "-1", "0", "1", "1", "1", "-1", "3", "3", "3", "1"
13 "INFP", "3", "1", "2", "0", "3", "1", "2", "1", "1", "0", "-1", "1", "-1", "0", "-1", "0"
14 "ESTP", "2", "0", "3", "0", "-1", "1", "1", "1", "2", "-1", "3", "-1", "1", "0", "1", "1"
15 "ESFP", "2", "-1", "3", "-1", "1", "0", "1", "1", "2", "-1", "3", "0", "0", "0", "1", "1"
16 "ISTP", "1", "0", "1", "0", "1", "3", "0", "2", "3", "-1", "3", "-1", "1", "1", "0", "0"
17 "ISFP", "3", "-1", "2", "-1", "3", "1", "2", "-1", "1", "1", "1", "0", "1", "1", "0", "1"
```

mbti_mate.csv
mbti_student.csv

mbti_student.csv	
1	,MBTI
2	단장님,ENTJ
3	남재환,INTJ
4	방영찬,ISFJ
5	김수아,INFJ
6	곽성화,ISFP
7	이재훈,INFJ
8	심정윤,ISTJ
9	김지윤,ESFP
10	우남광,INFP
11	허채범,ISFJ
12	신나령,ISFJ
13	이혜진,ENFP
14	임재원,ISFP
15	곽영호,INFP
16	정한수,ISFJ
17	박병준,ENFP
18	조건영,ISFJ
19	송인욱,INTP
20	김도영,INFP



프로젝트 설계



프로그램 기능



순서도



데이터 구성



코드 구성



GUI 흐름도

코드 구성 : MBTI 상세 설명 - 프로젝트 설계

```
class mbtiTest():
    head_question, MBTI_answers = get_questions()

    def __init__(self):
        self.mbti_type = ''
        self.testing = False
        self.answer = 0
        self.question_num = -1
        self.result = 0
```

```
# 함수 설명 -----
# 함수명 : mbtiTest.mbtiDetail
# 입력값 : mbti_type
# 반환값 : mbti 유형 설명 문자열 리스트
# 기능   : 해당 유형의 자세한 특징 불러오기
# -----

def mbtiDetail(self, mbti_type):
    path = './mbti_info/'+mbti_type+'.txt'
    with open(path, mode='r', encoding='utf-8') as f:
        data = []
        for d in f.readlines():
            try:
                for s in d.split('. '):
                    data.append(s)
            except:
                data.append(d)
    return data
```


MBTI 질문과 이지선다를 가져오는 함수

```
from settings import *

def get_questions():
    with open(QUESTIONS, 'r', encoding='utf-8') as f:
        total = f.read()
    head_question = []
    MBTI_answers = []
    for data in total.split('<title>')[1:]:

        datas = [x for x in data.strip().split('\n') if len(x)][2:]

        for i in range(len(datas)):
            if not i % 2:
                head_question.append(datas[i])
            else:
                MBTI_answers.append(datas[i])

    return head_question, MBTI_answers

class mbtiTest():
    head_question, MBTI_answers = get_questions()

    def __init__(self):
        self.mbti_type = ''
        self.testing = False
        self.answer = 0
        self.question_num = -1
        self.result = 0
```

1. 친구와의 약속이 취소되었을 때 나는?

뭐야.. 다른 친구에게 연락해야해서 놀자고 한다.
VS
오 ㅋㅋ 혼자서 집에서 뒹굴뒹굴 거린다.

test start	Left	Right	상세 설명으로
------------	------	-------	---------

2. 오랜만에 정말 맛있는 음식점을 찾았을 때 나는?

야야, 내가 맛있는 집을 아는데 꼭 한 번 가자! 주변사람들에게 적극적으로 추천한다.
VS
오 이집 진짜 맛집이네! 혼자 생각한다.

test start	Left	Right	상세 설명으로
------------	------	-------	---------

questions.txt

<title> Extraversion (외향) / Introversion (내향) </title>

에너지의 방향

1. 친구와의 약속이 취소되었을 때 나는?

뭐야.. 다른 친구에게 연락해야해서 놀자고 한다. vs 오 ㅋㅋ 혼자서 집에서 뒹굴뒹굴 거린다.

2. 오랜만에 정말 맛있는 음식점을 찾았을 때 나는?

야야, 내가 맛있는 집을 아는데 꼭 한 번 가자! 주변사람들에게 적극적으로 추천한다. vs 오 이집 진짜 맛집이네! 혼자 생각한다.

3. 조별과제를 할 때 나는?

무임승차 못참아! 참여하지 않은 인원을 제명하거나 건의한다. vs 사정이 있겠지... 한 번 참고 봐준다.

<title> Sencing (감각) / INTuituin (직관) 지구멸망?</title>

정보를 수집하거나 무엇을 인식하는 부분

4. 예전에 한 번 먹어본 음식에 대해서 나는?

한 번 먹어봤지만 생생하게 기억한다. vs 한 번 먹어봤는데 어떻게 기억해?

5. 내 최애 과자에 대해 설명할 때 나는?

그 과자는 말이야, 짹짹하고 바삭바삭하고 네모모양이고... 구체적으로 설명한다. vs 그냥 존맛탱, 단순하게 설명한다.

6. 친구가 우리 집에 놀러와서 물건을 찾을 때 나는?

그거 왼쪽 서랍 세 번째 칸에 있을거야. 구체적으로 알고 있다. vs 음 책상 쪽에 있을 걸? 한번 잘 봐봐.

```
def L_button_pushed(test, question_label, detail_text):  
    if test.testing:  
        test.answer = 1  
        test.getMBTI_gui(question_label, detail_text)  
  
def R_button_pushed(test, question_label, detail_text):  
    if test.testing:  
        test.answer = 0  
        test.getMBTI_gui(question_label, detail_text)
```

2. 오랜만에 정말 맛있는 음식점을 찾았을 때 나는?

야야, 내가 맛있는 집을 아는데 꼭 한 번 가자! 주변사람들에게 적극적으로 추천한다.

VS

오 이집 진짜 맛집이네! 혼자 생각한다.

test start

Left

Right

상세 설명으로

MBTI 질문에 대답하기

```
# 함수 설명 -----
# 함수명 : getMBTI_gui.getMBTI
# 입력값 : none
# 반환값 : mbti_type
# 기능 : tkinter 사용해 질문에 응답 입력 받고 mbti 유형 추정
# -----
def getMBTI_gui(self, question_label, detail_text):
    self.question_num = self.question_num+1
    if self.question_num == 12 : # 모든 질문 문답 시
        self.testing = False # 검사 종료
        question_label['text'] = '당신의 MBTI 성향은 ' + self.mbti_type + '!' # 검사 결과 출력
        self.show_detail(detail_text, self.mbti_type) # 상세 결과 출력

        self.question_num = -1 # 다음 검사를 위해 초기화
    else:
        kind = MBTI_KIND[self.question_num // 3] # 분류 유형 변환

        self.result = self.result + self.answer

        if self.question_num % 3 == 2:
            if self.result > 1:
                self.choice = 0 # 유형 '문자열' 의 첫 번째 값
            else:
                self.choice = -1 # 유형 '문자열' 의 마지막 값
            self.mbti_type = self.mbti_type+ kind[self.choice] # 뽑아낸 문자를 최종 결과에 추가

            self.result = 0 # 분류 유형 변환시 초기화
```

4. 예전에 한 번 먹어본 음식에 대해서 나는?

한 번 먹어봤지만 생생하게 기억한다.

VS

한 번 먹어봤는데 어떻게 기억해?

test start

Left

Right

상세 설명으로

settings

```
MBTI_MATE = './mbti_mate.csv'
QUESTIONS = './questions.txt'
STUDENTS = './mbti_student.csv'
RELATION_IMG_PATH = './relations/Relations_'

MATE={
    3: '이성적인 관계',
    2: '끈끈한 관계',
    1: '잠재적 관계',
    0: '갈등관계',
    -1: '맞지 않는 관계'}

MBTI_KIND = ['E / I', 'S / N', 'T / F', 'J / P']
```

menu

MBTI TEST

MBTI 상세 설명

MBTI 궁합

수강생 MBTI 궁합

당신의 MBTI 성향은 INFP!

test start

Left

Right

상세 설명으로

“열정적인 중재자”

중재자형 사람은 최악의 상황이나 악한 사람에게서도 좋은 면만을 바라보며 긍정적이고 더 나은 상황을 만들고자 노력하는 진정한 이상주의자입니다.

간혹 침착하고 내성적이며 심지어는 수줍음이 많은 사람처럼 비추어지기도 하지만, 이들 안에는 불만 지피면 활활 타오를 수 있는 열정의 불꽃이 숨어있습니다.

인구의 대략 4%를 차지하는 이들은 간혹 사람들의 오해를 사기도 하지만, 일단 마음이 맞는 사람을 만나면 이들 안에 내재한 총명한 즐거움과 넘치는 영감을 경험할 수 있을 것입니다.

이들은 논리나 단순한 흥미로움, 혹은 인생의 실용적인 부분이 아닌 그들 나름의 원리원칙에 근거하여 사고하고 행동합니다.

더욱이 성취에 따르는 보상이나 그렇지 못할 경우에 생길 수 있는 불이익 여부에 상관없이 순수한 의도로 인생의 아름다움이나 명예 그리고 도덕적 양심과 미덕을 좇으며 나름의 인생을 설계해 나갑니다.

그리고 그러한 본인들의 생각과 행동에 자부심을 느끼기도 하는데, 이는 마땅한 일입니다.

하지만 모든 사람이 그들의 생각 뒤에 숨은 동기나 의미를 정확히 파악하지는 못하는데, 이

성격유형 관계 - 공합

	분석가				감시자				협상가				장인			
	ENTJ	ENTP	INTJ	INTP	ESTJ	ESFJ	ISTJ	ISFJ	ENFJ	ENFP	INFJ	INFP	ESTP	ESFP	ISTP	ISFP
ENTJ																
ENTP																
INTJ																
INTP																
ESTJ																
ESFJ																
ISTJ																
ISFJ																
ENFJ																
ENFP																
INFJ																
INFP																
ESTP																
ESFP																
ISTP																
ISFP																

	이상적인 관계 - 문제가 잘 해결되고 관계가 자연스럽게 발전됨
	끈끈한 관계 - 어려움이 있을 수 있지만 타협점을 찾음
	잠재적 관계 - 끈끈한 관계를 위해서는 서로의 가치에 변화를 찾아야함
	갈등 관계 - 서로 타협하고 성숙해져야 유지될 수 있음
	맞지않는 관계 - 두 사람 모두가 무조건 양보해야하고 공감 필요함

코드 구성 : MBTI 공합

ENTJ, ENTJ

→ -1, 0, 1, 1, 0, -1, 0, -1, 0, 1, 1, 3, 2, 2, 1, 3,

ENTJ = 0

0, 2, 1, 1, 2, 2, 3, 3, 1, 0, 1, 1, 0, -1, 0, -1,

1, 1, 1, 0, 0, -1, 1, -1, 1, 1, 1, 2, 3, 3, 1, 2,

1, 1, 0, 1, 2, 3, 1, 3, 3, 1, 2, 0, 0, -1, 0, -1,

0, 2, 0, 2, 1, 0, 1, 1, 1, 1, 2, 0, 1, 1, 0, 3, 1,

0, 2, 0, 2, 1, 0, 1, 1, -1, 0, -1, 3, -1, 1, 1, 2,

0, 3, 1, 1, 1, 1, 1, 0, -1, 3, -1, 2, 1, 1, 0, 2,

-1, 3, -1, 3, 1, 1, 0, 0, 0, 3, 0, 1, 1, 1, 2, -1,

0, 1, 1, 3, -1, 1, -1, 0, 2, 0, 1, 1, 2, 2, 3, 1,

1, 0, 1, 1, 2, 2, 3, 3, 0, 1, 1, 0, -1, -1, -1, 1,

1, 1, 1, 2, -1, 0, -1, 0, 1, 1, 1, -1, 3, 3, 3, 1,

3, 1, 2, 0, 3, 1, 2, 1, 1, 0, -1, 1, -1, 0, 0, 0,

2, 0, 3, 0, -1, 1, 1, 1, 2, -1, 3, -1, 1, 0, 1, 1,

2, -1, 3, -1, 1, 9, 1, 1, 2, -1, 3, 0, 0, 0, 1, 1,

1, 0, 1, 0, 1, 3, 0, 2, 3, -1, 3, -1, 1, 1, 0, 0,

3, -1, 2, -1, 3, 1, 2, -1, 1, 1, 1, 0, 1, 1, 0, 1,

코드 구성 :

```
def matchMBTI(mbti01='INFJ', mbti02='INFJ'):
    # MBTI01은 setting 파일의 dic_MBTI 딕셔너리의 해당하는 value 값
    mbtiIndex = dic_MBTI.get(mbti01)
    print(f'(1) : {mbtiIndex}') ### mbtiIndex 값 체크

    # MBT
    filename = 'mbti.txt'
    list_
    with open(filename, mode='r', encoding='utf-8') as file:
        for i
            stat_list = file.read()
            i
            stat_list = stat_list.replace("\n", "")
            stat_list = stat_list.split(",")
            stat_list.pop()
            i += 1
    print(f'(2) : {mbtiIndex}') ### mbtiIndex 값 체크
```

```
filename = 'mbti.txt'
```

```
with open(filename, mode='r', encoding='utf-8') as file:
    dic_MBTI = {'ENTJ' : 0, 'ENTP' : 17, 'INTJ' : 33, 'INTP' : 49,
                'ESTJ' : 65, 'ESFJ' : 81, 'ISTJ' : 97, 'ISFJ' : 113,
                'ENFJ' : 129, 'ENFP' : 145, 'INFJ' : 161, 'INFP' : 177,
                'ESTP' : 193, 'ESFP' : 209, 'ISTP' : 225, 'ISFP' : 241}
    stat_list = file.read()
    stat_list = stat_list.replace("\n", "")
    stat_list = stat_list.split(",")
    stat_list.pop()
```

```
# num_Mate_Rate 는 list_MBTI_MATE의 mbtiIndex 인덱스값
num_Mate_Rate = str(stat_list[int(mbtiIndex)])
```

코드 구성 : MBTI

```
def getMateScore(mbti1='ISTJ', mbti2='INTJ'):
```

```
    mbti1    for i in range(0, 16):
```

```
    mbti2
```

```
        mate_data_0 = []
```

```
        for j in range(0, 16):
```

```
            num = mate_mbti.pop(0)
```

```
            mate_data_0.append(num)
```

```
            mate_data.append(mate_data_0)
```

```
    score = mate_data[mbti1][mbti2]
```

```
    for i in range(0, 16):
```

```
        mate_data_0 = []
```

```
        for j in range(0, 16):
```

```
            mate_data_0.append(num)
```

```
            mate_data.append(mate_data_0)
```

```
    score = mate_data[mbti1][mbti2]
```

```
    ['-1', '0', '1', '1',
```

```
    ['0', '2', '1', '1', '0
```

```
    ['1', '1', '1', '0', '1
```

```
    ['1', '1', '0', '1', '2', '3', '1', '3', '3', '1', '2', '0', '0', '-1', '0', '-1'],
```

```
    , '2', '2', '1', '3'],
```

```
    '0', '-1', '0', '-1'],
```

```
    , '3', '3', '1', '2'],
```

```
    [['-1', '0', '1', '1', '0', '-1', '0', '-1', '0', '1', '1', '3', '2', '2', '1', '3'], ['0', '2', '1', '1', '0', '0', '2', '2', '1', '0', '1', '1', '0', '-1', '0', '-1'], ['1', '1', '1', '0', '1', '1', '0', '1', '1', '0', '1', '1', '0', '1', '1', '0']]
```

```
    ['3', '-1', '2', '-1', '3', '1', '2', '-1', '1', '1', '1', '0', '1', '1', '-1', '1']]
```

	ENTJ	ENTP	INTJ	INTP	ESTJ	ESFJ	...	INFJ	INFP	ESTP	ESFP	ISTP	ISFP
ENTJ	-1	0	1	1	0	-1	...	1	3	2	2	1	3
ENTP	0	2	1	1	2	2	...	1	1	0	-1	0	-1
INTJ	1	1	1	0	0	-1	...	1	2	3	3	1	2
INTP	1	1	0	1	2	3	...	2	0	0	-1	0	-1
ESTJ	0	2	0	2	1	0	...	-1	3	-1	1	1	3
ESFJ	-1	2	-1	3	0	1	...	0	1	1	0	3	1
ISTJ	0	3	1	1	1	1	...	-1	2	1	1	0	2
ISFJ	-1	3	-1	3	1	1	...	0	1	1	1	2	-1
ENFJ	0	1	1	3	-1	1	...	1	1	2	2	3	1
ENFP	1	0	1	1	2	2	...	1	0	-1	-1	-1	1
INFJ	1	1	1	2	-1	0	...	1	-1	3	3	3	1
INFP	3	1	2	0	3	1	...	-1	1	-1	0	-1	0
ESTP	2	0	3	0	-1	1	...	3	-1	1	0	1	1
ESFP	2	-1	3	-1	1	0	...	3	0	0	0	1	1
ISTP	1	0	1	0	1	3	...	3	-1	1	1	0	0
ISFP	3	-1	2	-1	3	1	...	1	0	1	1	0	1

코드 구성 : 수강생 MBTI 궁합

- 프로젝트 설계

```
# -----  
# 클래스명 : Student  
# 역할 : mbti찾기, 랜덤대상찾기  
# -----  
class Student:  
    df = pd.read_csv(STUDENTS, index_col=0)  
    names = list(df.index)  
  
    def __init__(self, name):  
        self.name = name  
        self.mbti_type = self.studentMbti()  
  
import random as r  
import pandas as pd  
from settings import *
```

1	, MBTI
2	단장님, ENTJ
3	남재환, INTJ
4	방영찬, ISFJ
5	김수아, INFJ
6	곽성화, ISFP
7	이재훈, INFJ
8	심정윤, ISTJ
9	김지윤, ESFP
10	우남광, INFP
11	허채범, ISFJ
12	신나령, ISFJ
13	이혜진, ENFP
14	임재원, ISFP
15	곽영효, INFP
16	정한수, ISFJ
17	박병준, ENFP
18	조건영, ISFJ
19	송인욱, INTP
20	김도영, INFP

MBTI	
단장님	ENTJ
남재환	INTJ
방영찬	ISFJ
김수아	INFJ
곽성화	ISFP
이재훈	INFJ
심정윤	ISTJ
김지윤	ESFP
우남광	INFP
허채범	ISFJ
신나령	ISFJ
이혜진	ENFP
임재원	ISFP
곽영효	INFP
정한수	ISFJ
박병준	ENFP
조건영	ISFJ
송인욱	INTP
김도영	INFP

['단장님', '남재환', '방영찬', '김수아', '곽성화', '이재훈', '심정윤', '김지윤', '우남광', '허채범', '신나령', '이혜진', '임재원', '곽영효', '정한수', '박병준', '조건영', '송인욱', '김도영']

```
# -----  
# 함수명 : studentMbti  
# 기능 : student 클래스의 이름에 해당하는 mbti 값 찾아줌  
# 파라미터 : 없음  
# 결과값 : str  
# -----  
def studentMbti(self):  
    try:  
        df = pd.read_csv(STUDENTS, index_col=0)  
        studentMbti = df.loc[self.name, 'MBTI']  
    except Exception:  
        studentMbti = 'none'  
  
    return str(studentMbti)
```

```
# -----  
# 함수명 : randomStudent  
# 기능 : 랜덤으로 비교할 대상 찾아줌  
# 파라미터 : 없음  
# 결과값 : randomName, randomMbti  
# -----  
def randomStudent(self, not_include = ''):  
    try:  
        randomName = r.choice(Student.names)  
        while randomName == not_include:  
            randomName = r.choice(Student.names)  
        randomMbti = str(Student.df.at[randomName, 'MBTI'])  
    except Exception:  
        randomName, randomMbti = 'none', 'none'  
  
    return randomName, randomMbti
```



프로젝트 설계



프로그램 기능



순서도



데이터 구성



코드 구성



GUI 흐름도

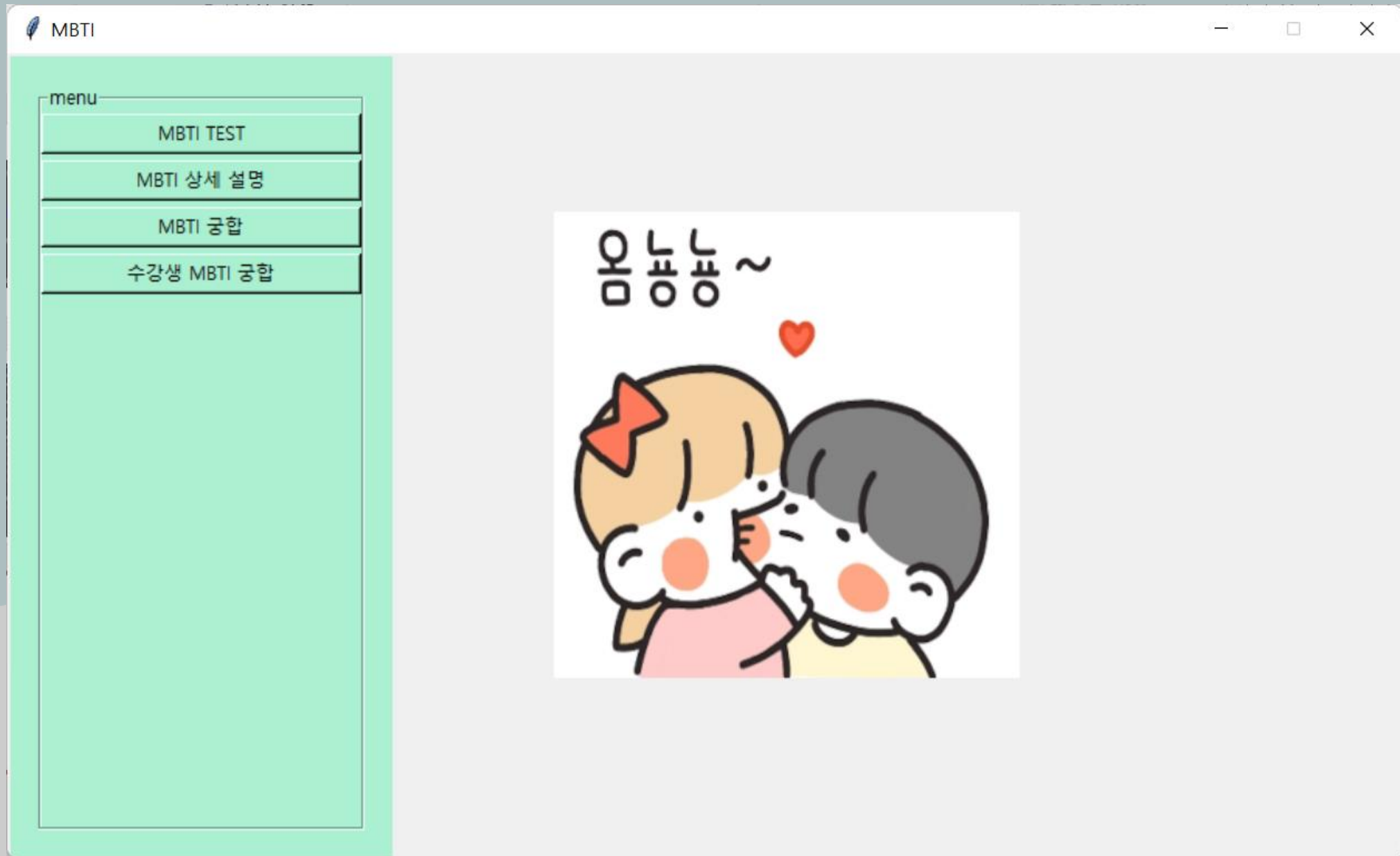
```
menu_frame = tk.LabelFrame(root, text='menu', bg=MENU_COLOR)
menu_frame.place(x=20, y=20, height=HEIGHT - 40, width=MENU_WIDTH - 40)
menu1 = tk.Button(menu_frame, text='MBTI TEST', bg=MENU_COLOR, repeatdelay=1000,
                  command=lambda: manu1_pushed(root))
menu2 = tk.Button(menu_frame, text='MBTI 상세 설명', bg=MENU_COLOR, repeatdelay=1000,
                  command=lambda: manu2_pushed(root))
menu3 = tk.Button(menu_frame, text='MBTI 궁합', bg=MENU_COLOR, repeatdelay=1000,
                  command=lambda: manu3_pushed(root))
menu4 = tk.Button(menu_frame, text='수강생 MBTI 궁합', bg=MENU_COLOR, repeatdelay=1000,
                  command=lambda: manu4_pushed(root))

menu1.pack(fill='both', pady=2)
menu2.pack(fill='both', pady=2)
menu3.pack(fill='both', pady=2)
menu4.pack(fill='both', pady=2)

IMG = tk.PhotoImage(file=RELATION_IMG_PATH + '3.png')
label = tk.Label(root, image=IMG)
label.place(x=MENU_WIDTH + 100, y=100)

root.mainloop()
```

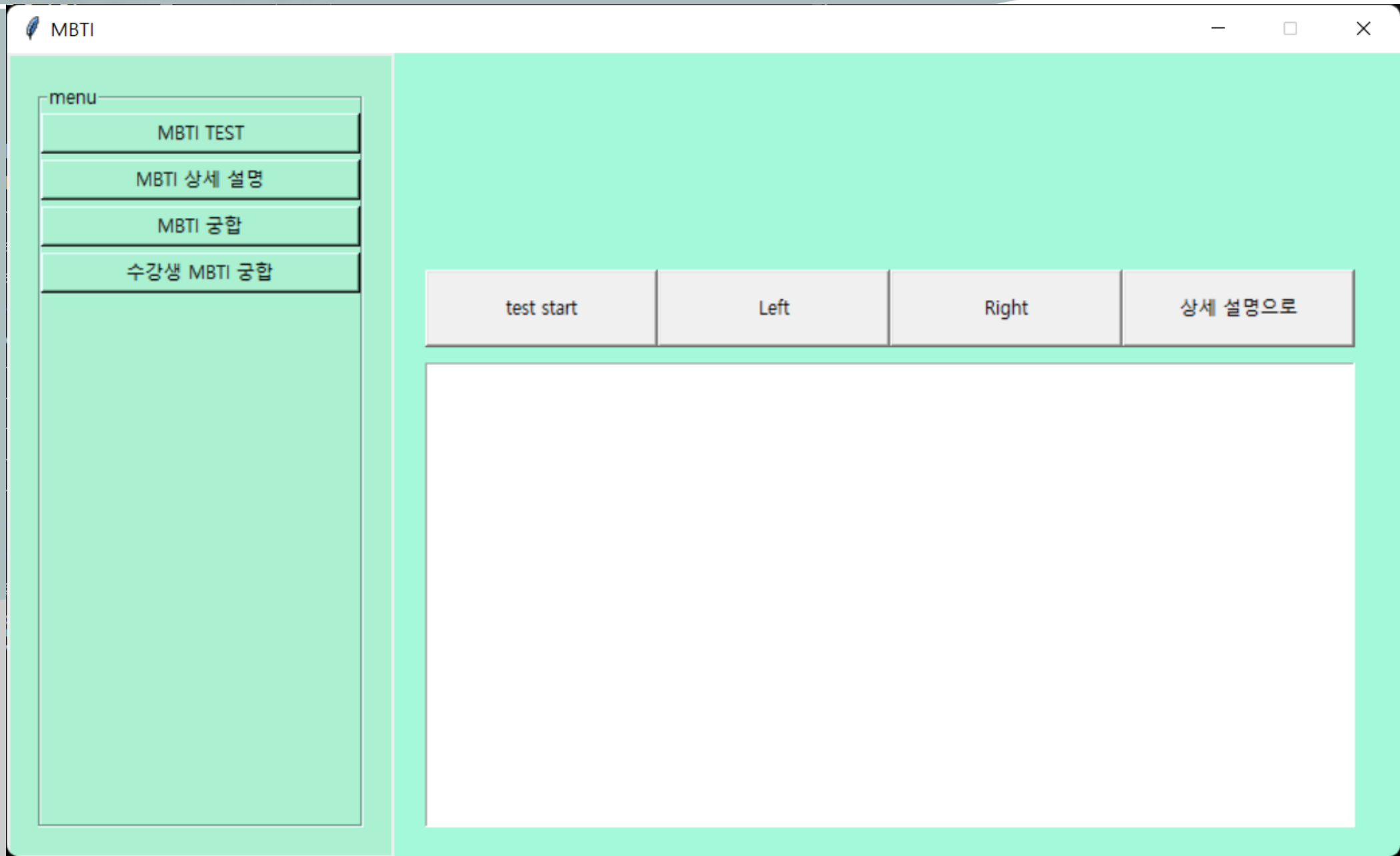

GUI 흐름도 - 프로젝트 설계



```
# -----  
# menu1 버튼이 눌릴 경우 mbti 검사 화면 보이기  
# 입력값 : tkinter.Tk() (화면)  
# 반환값 : Nan  
# -----  
def manu1_pushed(root):...  
# -----  
# menu2 버튼이 눌릴 경우 mbti 상세 설명 화면 보이기  
# 입력값 : tkinter.Tk() (화면)  
# 반환값 : Nan  
# -----  
def manu2_pushed(root, mbti_type='ISFJ'):...
```

```
# -----  
# menu3 버튼이 눌릴 경우 mbti 궁합 파악 화면 보이기  
# 입력값 : tkinter.Tk() (화면)  
# 반환값 : Nan  
# -----  
def manu3_pushed(root):...  
# -----  
# menu4 버튼이 눌릴 경우 수강생 mbti 궁합 화면 보이기  
# 입력값 : tkinter.Tk() (화면)  
# 반환값 : Nan  
# -----  
def manu4_pushed(root):...
```

GUI 흐름도 - 프로젝트 설계

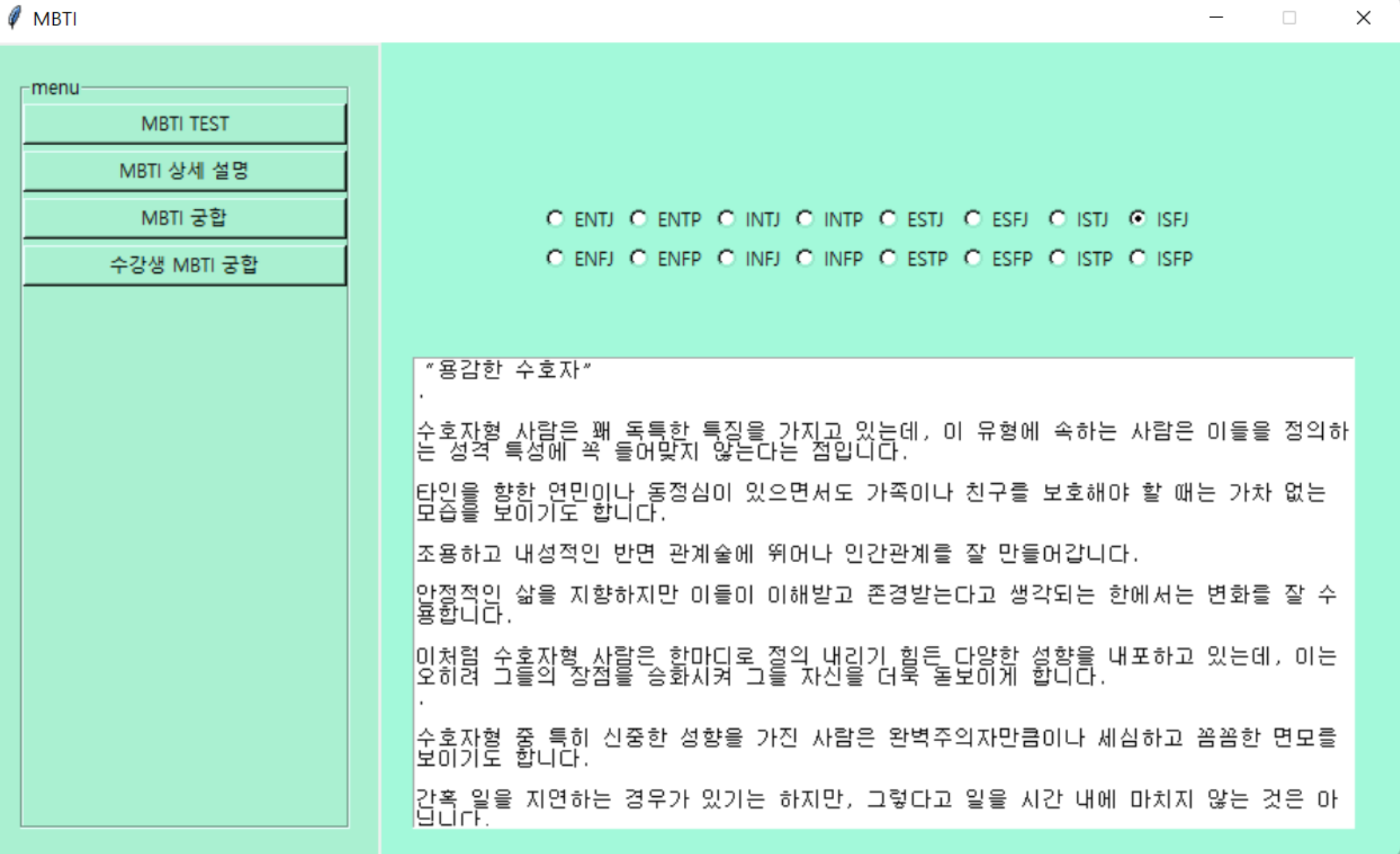


GUI 흐름도 - 프

```
# test_start 버튼이 눌릴 경우 mbti 검사 시작 및 질문 출력
# 입력값 : mbtiTest 객체, question_label (질문 칸) , detail_text (검사 결과 화면)
# 반환값 : Nan
# -----
def test_button_pushed(test, question_label, detail_text):
    if not test.testing:
        test.mbti_type = ''
        test.getMBTI_gui(question_label, detail_text)
        test.testing = True

# -----
# Left, Right 버튼이 눌릴 경우 결과 반영 및 다음 질문 출력
# 입력값 : mbtiTest 객체, question_label (질문 칸) , detail_text (검사 결과 화면)
# 반환값 : Nan
# -----
def L_button_pushed(test, question_label, detail_text):
    if test.testing:
        test.answer = 1
        test.getMBTI_gui(question_label, detail_text)

def R_button_pushed(test, question_label, detail_text):
    if test.testing:
        test.answer = 0
        test.getMBTI_gui(question_label, detail_text)
```



GUI 흐름도

- 프로젝트 설계

```
# 상세 설명에서 라디오 버튼이 눌릴 경우 버튼에 저장된 index에 따른 해당 유형의 설명 출력 함수 호출
# 입력값 : detail_text (상세 설명 화면), mbti_index (라디오 버튼에 포함된 변수), mbti_type (menu1에서 넘어올시 작동)
# 반환값 : Nan
]# -----

def radio_choice(detail_text, mbti_index, mbti_type=''):
    print(mbti[mbti_index.get()])
    test0 = mbtiTest()
    test0.show_detail(detail_text, mbti[mbti_index.get()])
    if len(mbti_type):
        test0.show_detail(detail_text, mbti_type)

]# -----

# 궁합에서 라디오 버튼이 눌릴 경우 버튼에 저장된 index에 따른 해당 유형 출력
# 입력값 : mate_text (타입 출력 화면), mbti_index (라디오 버튼에 포함된 변수)
# 반환값 : Nan
]# -----

def mate_choice(mate_text, mbti_index):
    print(mbti[mbti_index.get()])
    mate_text.config(state='normal')
    mate_text.delete('1.0', 'end')
    mate_text.insert('current', mbti[mbti_index.get()])
```



MBTI TEST

MBTI 상세 설명

MBTI 궁합

수강생 MBTI 궁합

MBTI 1

☒ ENTJ
 ☐ ENTP
 ☐ INTJ
 ☐ INTP
 ☐ ESTJ
 ☐ ESFJ
 ☐ ISTJ
 ☐ ISFJ
☐ ENFJ
 ☐ ENFP
 ☐ INFJ
 ☐ INFP
 ☐ ESTP
 ☐ ESFP
 ☐ ISTP
 ☐ ISFP

MBTI 2

☒ ENTJ
 ☐ ENTP
 ☐ INTJ
 ☐ INTP
 ☐ ESTJ
 ☐ ESFJ
 ☐ ISTJ
 ☐ ISFJ
☐ ENFJ
 ☐ ENFP
 ☐ INFJ
 ☐ INFP
 ☐ ESTP
 ☐ ESFP
 ☐ ISTP
 ☐ ISFP

MBTI 궁합 계산

[illegible]

GUI 흐름도

프로젝트 소개

```
# -----  
# 궁합에서 라디오 버튼으로 선택한 유형들의 궁합 결과를 출력  
# 입력값 : heart_text (하트 그릴 화면), result_label (궁합 결과 출력 화면),  
#         mate_text1 (첫 번째 유형이 입력된 text), mate_text2 (두 번째 유형이 입력된 text)  
# 반환값 : Nan  
# -----  
def mate_pushed(heart_text, result_label, mate_text1, mate_text2):  
    mbti_type1 = mate_text1.get("1.0", "end").strip()  
    mbti_type2 = mate_text2.get("1.0", "end").strip()  
  
    result_label['text'], score = getMateScore(mbti_type1, mbti_type2)  
  
    heart = returnHeart(score)  
    if score == 3:  
        heart_text['fg'] = 'red'  
    else:  
        heart_text['fg'] = 'black'  
  
    heart_text.config(state='normal')  
    heart_text.delete('1.0', 'end')  
    heart_text.insert('current', heart)  
    heart_text.config(state='disabled')  
    print(score)
```


MBTI

menu

MBTI TEST

MBTI 상세 설명

MBTI 궁합

수강생 MBTI 궁합

이름 :

이름 :

MBTI 궁합 계산

초기화

GUI 흐름도 - 프로젝트 설계

```
# 수강생 메뉴에서 입력한 수강생들의 유형과 공합 결과를 출력, 미입력시 랜덤으로 출력
# 입력값 : heart_text (하트 그릴 화면), result_label (공합 결과 출력 화면),
#         name_text1 (첫 번째 수강생), name_text2 (두 번째 수강생)
#         mate_text1 (첫 번째 유형이 입력될 text), mate_text2 (두 번째 유형이 입력될 text)
# 반환값 : Nan
# -----
def student_mate_pushed(heart_text, result_label, name_text1, name_text2, mate_text1, mate_text2):
    name1 = name_text1.get("1.0", "end").strip()
    name2 = name_text2.get("1.0", "end").strip()

    st0 = Student('0')
    if not len(name1):
        if len(name2):
            name1, mbti_type1 = st0.randomStudent(not_include=name2)
        else:
            name1, mbti_type1 = st0.randomStudent()
        name_text1.insert('current', name1)

    else:
        st1 = Student(name1)
        mbti_type1 = st1.mbti_type

    mate_text1.config(state='normal')
    mate_text1.delete('1.0', 'end')
    mate_text1.insert('current', mbti_type1)
    mate_text1.config(state='disabled')
```

```
if not len(name2):
    name2, mbti_type2 = st0.randomStudent(not_include=name1)
    name_text2.insert('current', name2)

else:
    st2 = Student(name2)
    mbti_type2 = st2.mbti_type

mate_text2.config(state='normal')
mate_text2.delete('1.0', 'end')
mate_text2.insert('current', mbti_type2)
mate_text2.config(state='disabled')

# 만들어둔 결과 출력 함수 사용
mate_pushed(heart_text, result_label, mate_text1, mate_text2)
```



프로그램 실행

들어주셔서 감사합니다!



Q&A