

한국사능력검정시험 가채점 프로그램

Korean history test scoring app

2018320185 김한주

프로젝트 개요

a) 내가 겪고 있는 문제 상황 제시

지난 여름방학에 “시간이 남아돌 때 뭐 라도 해 두자” 하는 생각에 한국사능력검정시험을 응시하였다. 일주일간 가볍게 공부를 하고 시험을 보러 갔다 왔고, 시험이 종료된 지 몇 시간 후에 공식 홈페이지에 정답이 올라왔다. 작년에 수능을 봤던 나는, 한국사능력검정시험 역시 가채점 서비스를 제공하는 곳이 있을 거라고 생각했다.

수능 같은 경우는 인터넷 강의 사이트(이하 인강 사이트)에서 수능 및 모의고사의 가채점 서비스를 제공한다. 수능 시험은 시험 종료 후 문제지를 제출해야 하기 때문에 가채점을 하려면 답안을 적어 나와야 한다. 수험생들은 자신이 제출할 답안을 수험표 뒷면과 같은 빈 종이에 답안을 적어와 이러한 가채점 서비스에 입력한다. 그러면 가채점 서비스를 통해 자동으로 채점되어 점수와 예상 등급을 제공받는다. 수험생의 성적을 조사하여, 모의고사를 분석하고 자체적인 등급컷을 제공하는 등 인강 사이트는 수험생의 성적을 수집하여 얻는 이점이 많다. 그래서 인강 사이트에서는 다양한 이벤트를 통해 많은 수험생들을 회원으로 모집하고 성적을 입력하도록 유도한다. 수험생 역시 편리하게 가채점을 할 수 있고 부족한 부분을 조언받을 수 있어 이러한 서비스를 대부분 이용한다.

하지만 한국사능력검정시험은 인강 사이트는 여럿 있지만, 등급이 없는 절대 평가이고, 시험지를 들고 나올 수 있기 때문인지 이러한 가채점 서비스를 제공하지 않았다. 시험을 보고 습관적으로 가채점을 위해 답안을 한 줄로 적어온 나는 답과 하나하나 비교하며 채점할 수밖에 없었다. 문제마다 배점도 다르다 보니 틀리더라도 몇 점이 깎였는 지 바로 계산할 수가 없었다. 그래서 가채점 프로그램을 직접 만든다면 편리하게 사용할 수 있겠다고 생각했다.

b) 문제 상황을 해결할 수 있는 방법(2가지 이상)

아무래도 이 프로그램이 유용하게 사용되려면 가장 최근 회차 시험의 답안이 실시간으로 반영되어야 할 것이다. 그래서 이 프로그램의 기본은 ‘어떻게 정답을 데이터베이스화 시킬 것인가’가 될 것이다. 방법은 여러가지가 있다.

1. 공식 홈페이지의 웹 데이터를 받아와 정답과 배점을 추출하기
2. 공식 홈페이지의 답안을 소스코드에 입력하기

3. 공식 홈페이지의 답안을 실행 과정에 입력하기

c) 여러가지 해결 방법 중 내가 선택한 방법

1번 방법이 가장 사용하기에도 편하고 실시간으로 반영되어 가장 좋긴 하지만, 한국사능력검정시험 홈페이지의 소스에서 정답 데이터에 접근하기 어려워 웹 데이터를 받아온다고 해서 정답을 추출하기가 힘들 것 같아 구현하기 어려울 것 같았다. 그래서 2번과 3번 방식을 고려해 보았다. 2번은 실시간으로 반영할 수만 있다면 빠르고 정확한 방법이지만, 시험이 시행될 때마다 프로그램을 업데이트해야 한다는 단점이 있다. 그래서 3번 방법으로 실행하는 과정도 제공한다면 그 단점을 어느정도 해소할 수 있다고 생각해서 2번과 3번 방법을 합쳐서 개발하기로 했다.

d) 예상되는 모습, 기대하는 효과

프로그램은 최근 회차의 시험을 채점하는 기능과 지난 기출문제를 채점하는 기능을 제공한다. 최근 회차 채점 기능은 프로그램의 자체 데이터베이스에서 최근 시험 정답을 추출하여 입력된 답안과 비교하는 방식과 사용자가 직접 정답을 입력한 뒤에 답안을 입력하는 방식을 제공하고, 사용자가 직접 정답을 입력할 때는 복사+붙여넣기 방식으로 쉽게 입력할 수 있게 하여 편리하게 이용할 수 있도록 한다. 답안 입력 방식 역시 1문제씩 / 5문제씩 / 한번에 입력 등 다양한 방식을 제공한다. 채점 후에는 점수와 시험 합격 여부, 그리고 어떤 문제를 틀렸는지 등을 알려준다. 잘못 채점한 경우를 고려하여 입력한 답안을 수정하는 기능도 제공한다.

이 프로그램을 통해 시험 응시자들은 직접 시험지를 채점할 필요 없이 답안만 입력하면 쉽게 점수 및 합격 여부를 알 수 있을 것이다. 가능하다면 웹페이지에서 정답 데이터를 추출하여 데이터베이스화 시키는 기능이나, 점수 화면을 캡처하여 사진 파일로 저장하는 기능도 추가할 것이다.

프로젝트 계획

초기 계획

1주차	~11/3	프로그램 구상 및 구현가능여부 조사
2주차	~11/10	기초 틀 구현
3주차	~11/17	데이터 수집 틀 구현
4주차	~11/24	데이터베이스 구현
5주차	~12/1	채점 기능 구현
6주차	~12/8	기능 연결 및 실행 여부 확인
7주차	~12/13	프로그램 검토

실제 진행

1주차	~11/3	프로그램 구상 및 구현가능여부 조사
2주차	~11/10	프로그램 설계
3주차	~11/17	데이터베이스 구조 구현
4주차	~11/24	데이터 수집 틀 설계
5주차	~12/1	프로그램 틀 구현
6주차	~12/8	채점 기능 구현 및 프로그램 완성
7주차	~12/13	데이터베이스, 부가기능, 주석 추가 / 프로그램 검토

프로젝트 결과

a) 계획서 대비 어떤 부분을 추가, 수정하였는지 기술

데이터베이스를 구현할 다양한 방법을 고려하였으나, 저장해야 할 데이터의 양이 많지 않다고 판단, 공식 홈페이지의 답안을 전부 데이터베이스에 입력하기로 했다. 실행 과정에 공식 홈페이지의 답안을 입력하는 기능은 직관성도 떨어지고 프로그램의 의의에 맞지 않다고 판단하여 넣지 않기로 했다. 대신 데이터베이스를 구현하는 과정의 편의를 위해, 데이터베이스는 공식 홈페이지의 답안 틀을 복사+붙여넣기 하는 방식으로 구현하였고, 이를 정답 데이터로 활용하기 위해 StringTokenizer 클래스를 이용해 데이터를 추출하는 방식을 사용하였다. 데이터베이스는 Database.java에, 데이터 추출 메소드는 App.java에 각각 구현하였다.

본 프로그램은 크게 초기 화면, 입력 화면, 결과 화면의 세 화면으로 나눌 수 있다. 초기 화면에서는 채점 회차, 난이도, 채점 방식을 선택한다. 채점 회차는 최근 회차와 기출을 따로 구분하지는 않았고, 최근 회차에 표시를 해 두었다. 채점 방식은 1문제씩 채점과 5문제씩 채점하는 방식을 구현하였다. 1문제씩 채점하는 방식은 OMR 답안지에 마킹하는 것과 유사하게, 5지선다 답안을 5개의 라디오 버튼에 입력하는 방식이다. 5문제씩 채점하는 방식은 '13242'와 같은 방식으로 5문제씩 나눠 답안을 5자리 숫자로 입력하는 방식이다. 한번에 입력하는 방식은 5문제씩 입력하는 방식보다 나은 점이 없다고 판단해서 제외하였다.

채점 방식까지 선택한 후 '채점하기' 버튼을 누르면 입력 화면으로 넘어간다. 입력 화면은 1문제씩 채점과 5문제씩 채점하는 방식 각각을 구현하였다. 중급, 고급 난이도는 50문제, 초급 난이도는 40문제이므로 각각을 적용하여 구현하였다. 1문제씩 채점하는 방식은 라디오버튼과 벡터를 사용했고, 5문제씩 채점하는 방식은 텍스트필드와 문자열 배열을 사용했다.

결과 화면에서는 계획서에 제시했던 대로 점수와 합격 여부, 오답 번호를 제시한다. 초급, 중급, 고급에 따라 합격하는 급수가 다르고, 70점 이상과 60점 이상 69점 이하의 급수도 다르기에 각각을 전부 구현하였다. 59점 이하는 불합격 안내문을 출력한다. 잘못 채점한 경우를 고려하여 수정 기능을 넣는 대신 다시하기 버튼을 넣어 처음부터 다시 채점할 수 있도록 했다. 잘못 채점한 경우 외에도 채점 회차 및 난이도를 잘못 선택할 수도 있기 때문에 수정 기능보다는 다시 채점하는 기능이 더 좋을 것 같았다. 다시하기 버튼을 누르면 초기 화면으로 돌아가며, 다시 채점하는 과정에서 이전에 입력한 정보는 전부 초기화된다. 웹페이지 데이터 추출, 점수 화면 캡처 등 추가적으로 넣으려 했던 부가 기능들은 구현하기 어려울 것 같아 넣지 않았다.

b) 깃허브 링크 주소

<https://github.com/hanjoo0211/Korean-history-test-scoring-app.git>

c) 동작 데모 영상 링크 주소

<https://youtu.be/IL-pJM2GuXA>

회고

a) 프로젝트를 설계 및 개발하는 과정에서 본인이 느낀점, 재미있었던 점, 어려웠던 점 등

프로젝트는 저번 학기 컴퓨터 프로그래밍 시간에도 진행했었지만, 그 때는 개발 틀이 주어지기도 했고, 주어진 과제를 구현하는 게 목적이라 프로젝트라기보다는 과제를 하는 느낌이었다. 그러나 이번 프로젝트는 주제도 직접 정하고, 개발 수준도 직접 정하는 프로젝트라 약간 부담이 되었다. 개인 프로젝트라 만들고 싶은 프로그램을 마음대로 만들 수 있어서 좋았지만 내가 뭔가를 잘 못하고 있어도 누가 도와주거나 지적해주지 않기 때문에 걱정도 되었다.

프로젝트 주제는 딱히 이렇다할 게 떠오르지 않았지만, 방학 때 1학기에 배웠던 c로 간단하게 한국사능력검정시험 채점 틀을 만들었던 게 계속 떠올라 자바로 제대로 구현해보자 해서 결정하게 되었다. 정답 데이터베이스를 구현하고, 답안을 입력 받아 데이터베이스와 비교하는 것이라서 구현하기 어렵지 않을 것 같았다. 당시까지 자바 응용 프로그램을 만드는 법이나 스윙 같은 것을 배우지 않아 어디까지 구현할 수 있는지 잘 몰랐지만, 그리 어렵지 않으면서 실용적으로 사용할 수 있는 프로그램이라고 생각했다.

코딩 과정은 처음이 가장 힘들었다. 어떤 구조로 개발을 해야 하는 지 하나도 몰라서 구글링을 계속하며 어떻게 시작할 지 고민했다. 내가 개발해야 할 프로그램은 화면을 넘기는 방식을 사용해야 하는데, 그동안 강의 시간에 배운건 JFrame을 사용해 한 화면에서만 작동하는 프로그램이라 어떻게 화면을 넘겨야 하는지 다양한 방법을 검색해 보았다. 그 결과 JPanel을 사용하는 게 가장 많아서 JPanel에 대해 알아보면서 어떻게 화면을 넘길 지 고민해 보았다. main 함수가 있는 java 파일 따로, 각각의 JPanel들을 따로 두는 방법 등 다양한 방법이 있었지만, 나는 모든 메소드를 한 java 파일에 모으는 게 코딩하기 편할 것 같아서 모든 메소드를 App.java에, 데이터베이스는 Database.java 에 구현하기로 했다.

일단 화면을 넘기는 방식을 구현하니 그 다음은 할 만했다. 강의 시간에 배운 내용들을 활용해서 필요한 기능들을 구현했고, 모르는 내용들은 검색하거나 이클립스 내부의 메소드 목록을 참고했다. 내가 강의 시간에 발표를 맡은 내용은 컬렉션과 제네릭이었는데, 그래서 답안 입력 저장엔 벡터를 이용해서 하기로 했었다. 가변 배열이기에 오류가 날 가능성도 적고 해서 배열을 사용하는 대신 벡터를 사용한 것인데, 프로그램을 열추 개발하고 나니 채점 과정에서 자꾸 오류가 발생했다. 여러 차례 코드를 수정해서 원인을 알고 보니 벡터의 add 메소드가 인덱스에 값을 삽입하면서 원래 있던 값을 덮어쓰는 게 아니라 뒤로 미는 게 그 원인이었다. 발표한 내용임에도 불구하고 잘 모르는 부분이 많았다. 이 오류는 벡터 대신 배열을 사용하는 것으로 해결했다. 벡터를 사용한 상태에서 해결할 수도 있었겠지만, 문제 개수는 50개 또는 40개로 고정되어 있으므로 굳이 가변 저장소를 사용할 필요가 없을 것 같았다.

1문제씩 채점하는 과정에서도 벡터를 사용하였다. 라디오버튼을 저장하려면 배열은 사용하지 못해서 벡터를 사용해야만 했다. 라디오버튼 그룹은 벡터로 선언할 수 있었는데, 문제에 딸린 5개의 라디오 버튼을 문제 단위 벡터로 저장하는 것이 문제였다. 검색을 해봐도 이중 배열은 나오는데

이중 벡터는 나오지 않아 해결할 수 없었다. 그러다 우리 발표에서 상자 안에 상자를 저장한다고 설명했던 다중 제네릭 기법이 떠올랐다. 그래서 발표 자료를 참고해서 라디오버튼 벡터를 다시 벡터로 구현할 수 있었다. 그 과정에서 여러 오류가 있었지만, 그 오류들을 해결하는 과정에서 제네릭에 대해 더 깊게 이해할 수 있었다.

프로그램을 다 개발하고 나니 여러 오류들이 많았지만, 그것들을 해결하는 과정에서 여러가지를 배울 수 있었다. 다양한 클래스들이 여러 유용한 메소드를 제공한다는 것을 알 수 있었고, 배열이나 벡터를 잘못 사용하였을 때 어떤 오류가 일어나고, 어떻게 해결하는 지 이해할 수 있었다. 무엇보다 지금까지 배웠던 자바의 여러 기능들을 직접 실용적으로 사용해 볼 수 있었다는 점이 좋았다. 교재를 읽기만 하는 것보다 예제를 풀 때 이해하기 쉽듯이, 프로젝트를 하나 완성하고 나니 자바를 더 깊게 이해할 수 있었고, 프로그램 개발에 대한 자신감도 생겼다. 간단한 프로그램을 하나 만드는 데도 많은 시간과 노력이 들었는데, 어떻게 하면 이 비용을 줄일 수 있을 지 궁금하기도 했다. 또한 보편적인 개발 방식에는 어떤 것이 있는지, 다른 사람들은 이러한 프로그램을 어떻게 구현하는 지 알고 싶다는 생각이 들었다.

b) 향후 프로젝트 진행 시 고려해야 할 사항

초반에 프로그램의 틀을 잡는 것이 가장 중요하다. 어떤 기능이 구현 가능하고 불가능하며, 쉽고 어려운지를 파악할 수 있는 것이 중요하다. 주어진 시간 내에 프로그램을 개발하기 위해, 구현하기 어렵거나 필요 없는 기능은 과감하게 제외하는 것이 필요하다. 수시로 프로그램을 디버깅하면서 오류가 발생하는 경우 그때그때 해결하는 것이 오류의 원인을 발견하기에도 수월하고, 수정할 코드의 양도 작아지게 된다.

보편적인 개발 방식을 사용하는 것도 중요하다. 개발의 효율성 측면에서도 그렇고, 오류를 수정할 때도 비슷한 오류를 겪는 사례를 찾아 쉽게 적용할 수 있다. 또한 그룹 프로젝트일 때는 더욱 중요하다. 같은 개발 방식을 그룹원들끼리 공유해야 하기 때문이다. 또한 코드를 읽을 다른 사람들을 위해 주석을 자세하게, 이해하기 쉽게 다는 방법도 알아야 한다.

참고 자료

황기태, *명품 자바 에센셜*, 생능출판사, 2018.

자바 swing 로그인 테스트 예제

<http://gogoke1.blog.me/220381451174>

자바 스윙(swing) – JPanel (2개의 패널 전환)

<http://blog.naver.com/battledocho/220012083751>