

c++ 프로그래밍 및 실습

대학생 대상 금융 정보 제공

시스템 개발 프로젝트

진척 보고서 #2

제출일자: 2023.12.03(일)

제출자명: 김채흔

제출자학번: 223445

1. 프로젝트 목표

1) 배경 및 필요성

대학생들이 자취 등으로 돈이 필요해 대출을 하거나, 적금/예금 등의 경제활동을 시작할 때 정보를 알지 못해 신청하지 못하는 경우가 많음. 또한 지원해주는 정책을 알지 못해 혜택이 있지만, 이를 알지도 못해 신청하지 못하는 경우가 많음. 이를 방지하기 위해 각자 상황에 맞는 정책 정보나 금융 상품 등의 금융 정보를 제공해주는 프로그램이 필요함.

2) 프로젝트 목표

사용자가 입력한 문제에 따른 금융 및 정책 정보를 자동적으로 제공.

3) 차별점

기존 사이트는 대출, 적금 등 다양한 정보와 정책 관련 정보가 여기저기에 있어서 경제활동을 이제 처음 시작하는 대학생들이 정보에 접근하기에 쉽지 않음. 또한, 은행 별로 사이트가 전부 달라 하나하나 비교해보아야 함. 이를 보완하여 사용자의 상황에 맞는 금융 정보를 제공하여 대학생들이 손쉽게 정보를 얻을 수 있도록 함.

2. 기능 계획

1) 기능 1 대학생 대상 정책 추천 기능

- 설명

사용자가 입력한 상황에 따라 정책을 추천해 주는 기능

(1) 세부 기능 1 정부 지원 정책 데이터 수집

- 설명

현재 정부에서 지원하고 있는 청년 관련 정책들 정보를 수집하여 지역별, 분야별로 세분화함.

(2) 세부 기능 2 추천 시스템

- 설명

지역/분야의 정보를 입력받고 조건에 해당하는 정책 정보를 출력

(3) 세부 기능 3 자세한 정보 제공

- 설명

사용자가 정보를 자세히 알고자 하면 세부사항 출력(사이트 정보 포함)

2) 기능 2 청년 대상 금융 정보 제공 기능

- 설명

대학생들이 받을 수 있는 대출 정보와 청년 대상 금융 상품을 함께 추천해준다.

(1) 세부 기능 1. 은행 별 금융 상품 수집

- 설명

은행 별 청년 대상 금융 상품과 대출 정보 데이터를 수집한다.

(2) 세부기능 2. 사용자 상황 맞춤 금융 정보 제공

- 설명

소득 수준/나이대/ 구직자 or 학생 등 사용자의 상황과 대출/적금 등 사용자의 목적을 입력받고 이에 해당하는 정보를 제공한다

(3) 세부기능 3. 구체적인 정보 제공

- 설명

사용자가 특정 정보에 대해 더 알고 싶어 할 경우, 관련 사이트 정보를 통해 구체적인 정보를 제공한다.

(4) 세부기능 4. 사용자 맞춤 추천

- 설명

사용자에게 입력받은 정보들을 바탕으로 가장 많이 검색한 순으로 상단에 띄워 추천해준다.

3. 진척사항

1) 기능 구현

(1) 기능 1 - 세부기능 3 자세한 정보 제공 수정

- 입출력

<financial.cpp>

- PolicyShow 함수

RegionPolicy = 사용자 지역 입력 변수

FieldPolicy = 사용자 분야 입력 변수

<csv.h>

- PrintInfo 함수

Region = 지역 필터링 변수

Field= 분야 필터링 변수

input = 정책 번호 변수

info = 필터링한 정책 저장 변수

- 설명

기존의 사용자가 입력한 조건에 맞는 모든 정책의 자세한 정보를 출력했던 것에서,

사용자에게 자세한 정보를 제공할 정책 번호를 입력받고 해당 정책의 세부 사항만을 출력한다.

- 적용된 배운 내용

벡터 : 필터링된 정책을 벡터로 저장한다.

함수 : PrintInfo, PolicyShow 함수를 활용하여 자세한 정보를 출력할때 필요한 코드를 함수화하여 사용하였다.

반복문, 조건문 : 반복문과 조건문으로 사용자가 0을 입력하기 전까지 입력받은 번호의 정책의 열들을 출력하도록 하였고, switch문을 사용해 사용자가 더 편하게 볼 수 있게끔 해당 열이 나타내는 것을 출력하였다.

- 코드 스크린샷

<financial.cpp>

```
//TO DO: 세부 기능 3 : 자세한 정보 출력
// 입력받은 지역과 분야에 해당하는 정책 출력 함수
void PolicyShow() {
    CSVReader csvReader("C:/Users/chee0/Downloads/download/YouthData.csv"); // 본인 컴퓨터에 따라 경로 바꿔줘야 함

    // 입력받은 분야와 지역을 사용하여 정책을 필터링하고 출력
    csvReader.PrintValue("youthPolicy.polyBizSecd", RegionPolicy, "youthPolicy.polyRlmCd", FieldPolicy, "youthPolicy.polyBizSjnm");
    csvReader.PrintInfo(RegionPolicy, FieldPolicy);
}
```

<csv.h>

```
// 필터링 된 정책 자세한 정보를 출력하는 함수
void PrintInfo(const string& Region, const string& Field) {
    vector<vector<string>> info = getinfo(Region, Field);
    int input=0;
    cout << "자세한 정보를 보고 싶은 정책의 번호를 입력하세요(없으면 0)" << endl;
    cin >> input;
    // 저장된 행들 출력
    while (input != 0) {
        // 출력할 행이 있는지 확인
        if (input >= 1 && input <= Policy.size()) {
            // 입력받은 행의 일들 출력
            for (const auto& cell : info[input - 1]) {
                // 만약에 필요한 정보만 출력
                if (cell != "-") {
                    if ((&cell - &info[input - 1][0] > 2) or (&cell - &info[input - 1][0] < 0) or (&cell - &info[input - 1][0] > 32))
                        switch (&cell - &info[input - 1][0]) {
                            case 2:
                                cout << "지역: [" << cell << "]" << endl;
                                break;
                            case 3:
                                cout << "기관 및 지자체 구분: [" << cell << "]" << endl;
                                break;
                            case 4:
                                break;
                        }
                }
            }
        }
    }
}
```

(2) 기능 2 - 세부기능 1 은행별 금융상품 수집

- 입출력

<financial.cpp>

class YouthFinancial = 금융정보 관련 기능 클래스

FinancialType = 사용자에게 금융상품 종류 입력 받는 변수

BankType = 사용자에게 은행명 입력받는 변수

FinancialItem = 금융상품 종류 벡터 변수

Banks = 은행명 벡터 변수

- isValid 함수

userInput = 유효성 체크할 사용자 입력 변수

ValidOptions = 유효성 체크할 벡터 변수

<csv.h>

CSVReader = 엑셀 데이터 불러오는 클래스(앞선 정책클래스에서 활용한 것과 동일)

- 설명

사용자에게 입력받은 은행명과 금융 상품 종류를 세분화하고, 사용자가 해당 카테고리 안에서 입력했는 지 유효성 검증을 한다.

- 적용된 배운 내용

함수 : 유효성 검증을 함수화하여 지속적으로 사용할 수 있도록 했다.

벡터 : 은행 명과 금융 상품 종류를 벡터로 설정해 후에 기능 구현에 유용하도록 한다.

- 코드 스크린샷

<financial.cpp>

```
// fig 2 : 은행 별 금융상품 정보 출력
//10.00 세분 기능 1 각 은행사 정보 수집 후 세분화
//각 은행사 홈페이지 정보 수집(금리 기준 2023.12.03)
class YouFinancial {
private:
    string FinancialType; //금융상품 종류 입력 받는 변수
    string BankType; //은행 앞 1 글자 변수
    string input;
    //금융상품 종류 벡터
    vector<string> FinancialItem = { "청년도입예저", "청년희망저축", "기타 금융 상품" };
    //은행 벡터
    vector<string> Banks = { "NH농협은행", "신한은행", "우리은행", "SC제일은행", "하나은행", "IBK기업은행", "KB국민은행", "DB대구은행", "FNK" };

public:
    //유효성 검증 함수
    bool isValid(const string& userInput, const vector<string>& validOptions) {
        return userInput == "0" || find(validOptions.begin(), validOptions.end(), userInput) != validOptions.end();
    }
};
```

<csv.h>

```
class CSVReader {
public:
    CSVReader(const string& filename, char delimiter = ',') : filename_(filename), delimiter_(delimiter) {}
    // 필터링 된 정책 벡터
    vector<string> Policy = {};
    //엑셀 파일 읽어오기
    vector<vector<string>> readData() {
        ifstream in(filename_);
        if (!in.is_open()) {
            cerr << "Failed to open file: " << filename_ << endl;
            return {};
        }

        vector<vector<string>> data;
        string line;
        string currentRow;
        while (getline(in, line)) {
            currentRow += line;

            // 큰 텍스트 블록을 묶은 경우 처리
        }
    }
};
```

(3) 기능 2 - 세부기능 2 사용자 맞춤 금융정보 제공

- 입출력

<financial.cpp>

- Bank 함수

BankType = 사용자 입력 은행명

- Item 함수

FinancialType = 사용자 입력 금융상품 정보

- FinancialShow 함수

YouthAccount = 사용자가 입력한 은행에 맞는 청년도약계좌 정보 출력 함수

<csv.h>

- GetInfo함수

bank = 출력하려는 은행명

filename_ = CSV 파일의 경로를 저장하는 변수

file = 파일을 열 때 사용되는 변수

line = 한 행의 데이터를 저장하는 문자열

iss = 한 행의 csv 데이터를 , 로 구분하기 위한 변수

token = 구분한 각 열 값 저장 변수

tokens = 열 값을 모으기 위한 벡터

result = 조건 만족하는 행을 저장 벡터

- YouthAccount 함수

info = 조건 만족하는 행들 저장

cell = 행에서 첫번째 열 값을 하나씩 읽기 위한 변수

- 설명

사용자가 금융 상품 종류(청년도약계좌)와 은행 명을 입력하면 그에 맞는 자세한 정보를 출력해준다.

- 적용된 배운 내용

반복문 : 사용자에게서 입력받을 때 유효성 검증함수 확인

: 사용자가 보기 편하도록 각 열이 뜻하는 정보 출력(switch 문)

- 코드 스크린샷

financial.cpp

```
//To Do: 시무 기능 2 사용자 입력 검증 금융 정보 출력
// 은행명 입력 함수
void Bank() {
    cout << "은행을 선택하세요(종류 시 C)" << endl;
    cout << "NH농협은행 /신리은행 /우리은행 /SC제일은행 /하나은행 /IBK기업은행 /KB국민은행 /DGB대구은행 /BNK부산은행 /광주은행 /제주은행 /BNK"
    cin >> BankType;

    while (!IsValid(BankType, Banks)) {
        cout << "잘못된 입력입니다. 다시 입력하세요." << endl;
        cin >> BankType;
    }
}

// 금융 상품 입력 함수
void Item() {
    cout << "금융 상품을 선택하세요(종류 시 A)" << endl;
    cout << "청년도약계좌 /청년도약계좌 /기타 금융 상품" << endl;
    cin >> FinancialType;

    while (!IsValid(FinancialType, FinancialItem)) {
        cout << "잘못된 입력입니다. 다시 입력하세요." << endl;
        cin >> FinancialType;
    }
}

// 입력받은 은행에 맞춘 청년도약계좌 정보 출력 함수
void FinancialShow() {
    if (FinancialType == "청년도약계좌") {
        CSVReader csvReader("C:/Users/chee0/Downloads/financial.csv"); // 본인 컴퓨터에 따라 경로 바꿔줘야 함
        csvReader.YouthAccount(BankType);
    }
}
```

financial.cpp - main함수

```
int main() {
    YouthPolicy yp;
    YouthFinancial yf;
    while (true) {
        yp.Field();
        if (yp.getFieldPolicy() == "0") //0 입력하면 종료
            break;
        yp.Region();
        if (yp.getRegionPolicy() == "0")
            break;
        yp.PolicyShow();
        yf.Bank();
        if (yf.getBankType() == "0")
            break;
        yf.Item();
        if (yf.getFinancialType() == "0")
            break;
        yf.FinancialShow();
    }

    return 0;
}
```

csv.h

```
//사용자가 입력한 은행에 따른 청년도약계좌 정보 출력 함수
void YouthAccount( const string& value) {
    vector<vector<string>> info = GetInfo(value);
    // 저장된 행들 출력
    for (const auto& cell : info[0]) {
        //      // 각 열의 정보 출력
        switch (&cell - &info[0][0]){
            case 0:
                cout << "은행명: [" << cell << "]" << endl;
                break;
            case 1:
                cout << "기본 금리(3년 고정): [" << cell << "]" << endl;
                break;
            case 2:
                cout << "소득 우대금리: [" << cell << "]" << endl;
                break;
            case 3:
                cout << "적금담보대가산금리(만기일시상환대출): [" << cell << "]" << endl;
                break;
        }
    }
}
```

2) 테스트 결과

(1) 기능 1 - 세부기능 3 자세한 정보 제공 수정

- 설명

사용자가 입력한 정책 번호에 맞는 세부 정보 출력

- 테스트 결과 스크린샷

```
자세한 정보를 보고 싶은 정책의 번호를 입력하세요(없으면 0)
1
지역: [광주]
기관 및 지자체 구분: [지자체]
정책명: [성년 교통수당 드림]
정책 소개: [취업준비와 생활비 등으로 어려움을 겪고 있는 청년들에게 교통비를 지원하여 구직활동 참여를 지원]
지원 내용: [대중교통 전용카드 30만원 지급(일시불 교통카드 충전)]
지원 규모: [1000명]
사업유형기간: [2023.08.09.~2023.12.31.]
사업신청기간: [사정에 따라 변경 될 수 있습니다.]
연령: [제한없음]
신청 절차: [성년교통수당드림 홈페이지 (https://www.gjdream.kr) - 모든 서류는 스캔?촬영(JPG/PDF) 후 성명.zip파일로 업로드]
제출 서류: [공고문 참고(지원신청서 개인정보동의서 주민등록등본 최종하락증명서 건강보험료 납부확인서 건강보험자 고용보험 피보험자격 이력내역시 등)]
심사 번호: [광주광역시 홈페이지 및 청년정책플랫폼 지원지 신청 공고]
신청 사이트 주소: [http://www.gjdream.kr]
수관 부처 : [광주광역시 청년정책관]
주관부처 담당자 이름 : [고민수 주무관]
주관부처 담당자 연락처 : [062-613-2721]
유형기관명 : [(주)한페이지스]
유형기관 담당자 연락처 : [062-601-0041]
자세한 정보를 보고 싶은 정책의 번호를 입력하세요(없으면 0)
```

(2) 기능 2 - 세부기능 1 은행별 금융상품 수집

- 설명

사용자가 은행 명과 금융 상품 종류를 입력할 수 있는 카테고리 세분화하고 유효성 검증하여 출력

- 테스트 결과 스크린샷

```
은행을 선택하세요(종료 시 0)
NH농협은행 /신한은행 /우리은행 /SC제일은행 /하나은행 /IBK기업은행 /KB국민은행 /DGB대구은행 /BNK부산은행
은행 /BNK경남은행
우리은행
잘못된 입력입니다. 다시 입력하세요.
```

(3) 기능 2 - 세부기능 2 사용자 맞춤 금융정보 제공

- 설명

사용자가 알고자하는 은행의 청년도약계좌 세부 정보 제공

- 테스트 결과 스크린샷

NH농협은행 / 신한은행 / 우리은행 / SC제일은행 / 하나은행 / IBK기업은행 / KB국민은행 / DGB대구은행 / BNK부산은행 / 광주은행 / 신한은행 / BNK경남은행
신한은행
금융 상품을 선택하세요(종료 시 0)
청년도약계좌 / 청년희망적금 / 기타 금융 상품
청년도약계좌
은행명: [신한은행]
기본 금리(3년 고정): [4.50]
소득 우대금리: [0.50]
적금남보태가산금리(만기일시상환대출): [1.00]
적금남보태가산금리(한도대출): [1.00]
적금남보태가산금리(대출가능률): [100]
은행별 우대금리: ["우대금리 최대 1.00%p @ 급여이체 0.30%p @ 신한카드(신용/체크) 잔액 0.30%p @ 첫 거래 우대 0.40%p"]
만기 후 금리: ["만기후 1개월 이내 : 만기일 당시의 가임기간에 해당하는 일반정기적금 연 이자율의 1/2 (단 최저금리 연 0.1%)
- 만기후 1개월 초과 6개월 이내 : 만기일 당시의 가임기간에 해당하 : 일반정기적금 연 이자율의 1/4 (단 최저금리 연 0.1%)
- 만기후 6개월 초과 : 연 0.1%]
우대조건: ["@ [급여이체 우대] 이 상품 신규 후 본인명의 신한은행 입출금통장으로 급여이체 실적(건당 50만원 이상)이 30개월 이상인 경우 @ [신한카드 결제 우대] 이 상품 신규 후 본인명의 신한은행 입출금통장으로 본인명의 신한카드(신용/체크) 결제실적이 30개월 이상 있 : 경우 @ [첫 거래 우대] 이 상품 신규 직후 1년간 신한은행 정기예금 정기적금 주택신탁 상품이 없었던 고객 이 이 상품을 신규한 경우"]
기타 유의사항: [* 급여 및 카드결제는 30개월 연속하지 않아도 됨.]
참고 링크: [https://bank.shinhan.com/index.jsp#020102010110]
정액 분야를 선택하세요(종료 시 0)
일자리/ 주거/ 교육/ 복지/ 참여

4. 계획 대비 변경 사항

1) 금융 정보 세분화 카테고리 설정

- 이전

금융 정보의 카테고리를 은행 별로 다르게 두어, 사용자가 은행을 입력하면 해당 은행의 카테고리를 나열 수 있게끔 하였음.

- 이후

청년도약계좌와 청년희망적금은 따로 뒀음.

- 사유

가장 대표적이고 많이 찾는 두 가지는 더 쉽게 비교할 수 있게끔 따로 빼는 게 낫다고 판단. 나머지 상품은 기타에서 은행별로 출력 예정

2) 계획 일정 변경

- 이전

본래 기능 2의 세부기능1/2를 모두 구현하려 했음.

- 이후

세부기능 2의 일부(청년도약계좌)만 구현함.

- 사유

데이터를 찾는 데에 생각보다 오래걸렸으며, 엑셀 데이터의 인코딩문제가 겹쳐 일정대로 하지 못함.

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/22	11/25	11/26
제안서 작성		완료			
기능1	세부기능1		완료		
	세부기능2			완료	
	세부기능3				완료

업무		11/30	12/1	12/3	12/16
기능1	세부기능3	완료			
기능2	세부기능1		완료		
	세부기능2			-----> (진행 중)	

업무		12/22	12/23	12/3	12/16
기능2	세부기능3	(예정)			
	세부기능4		(예정)		