

C++ 프로그래밍 및 실습

# 예금 상품 제작 프로그램 개발

진척 보고서 #1

제출일자: 2023.11.26.일

제출자명: 김 지 웅

제출자학번: 203185

# 1. 프로젝트 목표

## 1) 배경 및 필요성

금융시장이 발달함에 따라 금융상품 또한 더욱 복잡해지고 개수도 많아지고 있다. 우리는 금융상품 중 어떠한 상품이 본인에게 맞는지 파악하기 어려울 뿐만 아니라 금융상품을 이해하기 위한 지식도 부족한 것이 현 상황이다.

따라서 우리는 이러한 문제를 극복하기 위해 금융지식을 알려주며 고객이 직접 금융상품을 제작하고 그에 따른 수익을 확인시켜주는 프로그램을 만들고자 한다.

## 2) 프로젝트 목표

고객이 원하는 목표 수익에 맞춰 이자율과 납입 금액, 만기일을 직접 설정하고 해당 상품을 등록하는 프로그램을 만든다.

## 3) 차별점

기존 금융상품의 경우, 은행에서 정한 이자율과 만기에 따라 고객은 일정 금액을 납입하여 이자를 받는 구조였다. 하지만 이와 같은 구조의 상품들이 대량으로 쏟아지고 있어 고객은 어떠한 상품이 본인에게 맞는지 모르는 상황이 발생하고 있다. 우리가 만들고자 하는 프로그램은 고객의 니즈에 맞춰 이자율과 만기일을 설정하고 목표 수익을 달성하는 상품을 직접 제작한다는 것이 기존과의 차별점이다.

## 2. 기능 계획

### 1) 기능 1 : 원하는 목표 수익에 맞춘 상품 제작

- 설명 : 사용자가 직접 상품의 정보를 설정하여 금융 상품을 제작하는 기능이다.

#### (1) 세부 기능 1 : 사용자가 원하는 금융 상품을 제작

- 설명 : 사용자는 납입액은 얼마인지, 만기는 언제까지인지, 이자율을 몇퍼센트인지를 직접 설정하고 만든 상품의 수익을 확인하는 기능을 제공한다.

#### (2) 세부 기능 2 : 사용자의 정보 저장 후 확인

- 설명 : 사용자가 만든 상품을 저장하여 필요에 따라 상품의 정보를 확인하는 기능을 제공한다.

### 2) 기능 2 : 금융 상품 정보 제공

- 설명 : 고객이 원하는 상품과 유사한 상품을 검색할 수 있도록 정보를 제공하는 기능이다.

#### (1) 세부 기능 1 : 금융 상품 정보 제공

- 설명 : 시중에서 제공하는 상품을 데이터화하여 저장한 후 기능 1 에서 사용자가 만든 상품과 유사한 상품을 찾아 사용자에게 추천하는 기능을 제공한다.

### 3. 진척사항

#### 1) 기능 구현

##### (1) 예금 상품 제작 기능

- 입출력 : 사용자로부터 이름/생년월일/원금/이자율/만기/납입액을 입력 받아 이를 예금 계산식에 대입하여 최종지불액을 출력함.

- 설명 : DepositProduct클래스를 만들어 예금 상품을 만드는 멤버 함수를 정의함으로써 createDepositProduct 함수를 정의할 때 이를 사용하여 입력 받은 정보를 계산하여 최종 지불액을 계산함.

- 적용된 배운 내용

<클래스 및 객체 지향 프로그래밍>

먼저 예금 상품을 제작하기 위해 사용자 이름과 생년월일, 원금, 이자율, 만기, 납입액을 클래스 내에서 멤버 변수로 선언함. 이후 생성자를 만들고 예금을 계산하는 멤버 함수와 결과를 출력해주는 멤버 함수를 정의함.

<반복문과 조건문 활용>

예금 상품을 계산해주는 함수에서 매월 납입액 추가하는 구문과 매월 이자를 계산하는 구문을 반복문을 활용하여 계산해줌.

메인함수에서 초기화면을 출력 후 do-while문을 통해 업무 수행 후 다시 초기 화면으로 돌아가는 것을 구현하였으며 switch 조건문을 활용하여 번호에 맞게 임무를 실행함.

<함수를 활용하여 적용>

프로그램에서 활용한 함수는 예금 계산 함수와 결과 출력 함수 그리고 사용자에게 정보를 입력 받아 예금 상품을 만드는 함수를 정의함.

예금 계산 함수의 경우 반환값이 존재하므로 Double 자료형으로 설정하였음

예금 상품 제작 함수의 경우에는 클래스에서 만든 함수들을 활용하여 결과를 출력하고 저장하는 방식의 함수를 제작함.

## - 코드 스크린샷

```
8  class DepositProduct {
9  public:
10     string name;           // 사용자 이름
11     string birthdate;      // 사용자 생년월일
12     double principal;      // 원금
13     double interestRate;   // 이자율
14     int maturityPeriod;    // 만기 기간 (월 단위)
15     double monthlyDeposit; // 월 납입액

17 public:
18     // 생성자
19     DepositProduct(const string& n, const string& bd, double p, double rate, int period, double deposit)
20         : name(n), birthdate(bd), principal(p), interestRate(rate / 100.0), maturityPeriod(period), monthlyDeposit(deposit) {}
21
22     // 예금 계산 메서드
23     double calculateTotalAmount() const {
24         double totalAmount = principal; // 최종 지불액 초기화
25
26         for (int month = 1; month <= maturityPeriod; ++month) {
27             totalAmount += monthlyDeposit; // 월 납입액 추가
28             totalAmount *= (1 + interestRate / 12); // 월 이자 계산
29         }
30
31         return totalAmount;
32     }
33
34     // 결과 출력 메서드
35     void displayResult() const {
36         cout << "\n정기예금 계좌 개설이 완료되었습니다.\n";
37         cout << "이름: " << name << "\t생년월일: " << birthdate << endl;
38         cout << "최종 지불액: " << fixed << setprecision(2) << calculateTotalAmount() << " 원\n";
39     }
40
41 void createDepositProduct() {
42     // 사용자로부터 입력 받기
43     string name, birthdate;
44     double principal, interestRate, monthlyDeposit;
45     int maturityPeriod;
46
47     cout << "이름을 입력하세요: ";
48     cin >> name;
49
50     cout << "생년월일을 입력하세요 (YYYYMMDD 형식): ";
51     cin >> birthdate;
52
53     cout << "원금을 입력하세요: ";
54     cin >> principal;
55
56     cout << "이자율을 입력하세요 (%): ";
57     cin >> interestRate;
58
59     cout << "만기 기간을 입력하세요 (월 단위): ";
60     cin >> maturityPeriod;
61
62     cout << "월 납입액을 입력하세요: ";
63     cin >> monthlyDeposit;
64
65     // DepositProduct 클래스를 사용하여 예금 계좌 개설 및 결과 출력
66     DepositProduct depositAccount(name, birthdate, principal, interestRate, maturityPeriod, monthlyDeposit);
67     depositAccount.displayResult();
68     depositAccount.saveToFile(); // 파일에 정보 저장
69 }

145 int main() {
146     int mainMenuChoice;
147
148     do {
149         cout << "-----예금 상품 제작-----" << endl;
150         cout << "1. 예금 상품 만들기 \t 2. 본인 상품 확인하기 \t 3. 추천 상품 확인하기 \t 4. 종료" << endl;
151         cout << "-----" << endl;
152         cout << "하시고자 하는 업무를 선택하여 주세요 : ";
153         cin >> mainMenuChoice;
154
155         switch (mainMenuChoice) {
156             case 1:
157                 createDepositProduct();
158                 break;
159             case 2:
160                 checkOwnProduct();
161                 break;
162             case 3:
163                 financialInformation();
164                 break;
165             case 4:
166                 cout << "프로그램을 종료합니다." << endl;
167                 break;
168             default:
169                 cout << "올바르지 않은 선택입니다. 다시 선택하세요." << endl;
170                 break;
171         }
172     } while (mainMenuChoice != 4);
173
174     return 0;
175 }
```

## (2) 본인 상품 확인 기능

- 입출력 : 입력 받은 사용자 정보를 파일에 저장한 뒤 이름과 생년월일을 입력하면 해당 인원의 상품 정보를 출력함.
- 설명 : Deposit클래스에서 사용자의 정보를 저장하고 불러오는 멤버함수를 정의하고 CheckOwnProduct 함수를 정의할 때 사용하여 본인의 정보를 입력하면 본인이 만든 상품의 정보를 출력하도록 함.
- 적용된 배운 내용

### <클래스 및 객체 지향 프로그래밍>

개인 정보를 파일에 저장하고 불러오기 위해 처음에 만든 DepositProduct클래스에서 선언된 멤버변수를 활용하여 계좌 정보를 저장하는 멤버 함수와 계좌 정보를 불러오는 멤버함수를 정의함. 이를 통해

### <반복문과 조건문 활용>

사용자의 정보를 불러오는 함수를 정의하는 과정에서 파일에 있는 정보를 전부 확인하기 위해 while문과 if-else문을 활용하여 파일 끝까지 확인하는 구문을 작성함.

또한 사용자의 정보를 입력 받고 출력하는 함수에서 사용자의 정보가 있는지를 확인하는 과정을 조건문을 활용하여 작성함.

### <벡터 동적 배열 활용>

사용자의 정보를 불러오는 함수에서 동적 배열 accounts를 선언하고 push\_back를 활용하여 원하는 계좌 정보를 벡터에 넣어 보관함.

### <함수를 활용하여 적용>

사용자에게 이름과 생년월일을 입력 받아 해당 계좌를 불러오기 위해 클래스에서 정의한 멤버 함수를 바탕으로 작성한 뒤 메임 함수에 적용함.

### <정적 멤버 함수 활용>

파일에서 계좌를 불러오는 멤버 함수를 정의하는 과정에서 정적 함수를 이용하여 클래스에 의존하지 않고 활용이 가능하도록 함.

## - 코드 스크린샷

```
41 // 계좌 정보를 파일에 저장
42 void saveToFile() const {
43     ofstream outFile("deposit_info.txt", ios::app);
44     if (outFile.is_open()) {
45         outFile << name << " " << birthdate << " " << principal << " " << interestRate << " " << maturityPeriod << " " << monthlyDeposit << endl;
46         outFile.close();
47     } else {
48         cout << "파일을 열 수 없습니다." << endl;
49     }
50 }
51
52 // 파일에서 계좌 정보를 불러오기
53 static vector<DepositProduct> loadFromFile() {
54     vector<DepositProduct> accounts;
55     ifstream inFile("deposit_info.txt");
56     if (inFile.is_open()) {
57         while (true) {
58             string n, bd;
59             double p, rate, deposit;
60             int period;
61             if (!(inFile >> n >> bd >> p >> rate >> period >> deposit)) {
62                 break; // 파일 끝에 도달하면 종료
63             }
64             DepositProduct account(n, bd, p, rate, period, deposit);
65             accounts.push_back(account);
66         }
67         inFile.close();
68     } else {
69         cout << "파일을 열 수 없습니다." << endl;
70     }
71     return accounts;
72 }
73 };

```

```
111 void checkOwnProduct() {
112     // 파일에서 계좌 정보 불러오기
113     vector<DepositProduct> accounts = DepositProduct::loadFromFile();
114
115     if (accounts.empty()) {
116         cout << "아직 개설된 계좌가 없습니다." << endl;
117     } else {
118         // 사용자로부터 이름과 생년월일 입력 받기
119         string name, birthdate;
120         cout << "이름을 입력하세요. ";
121         cin >> name;
122         cout << "생년월일을 입력하세요 (YYYYMMDD 형식): ";
123         cin >> birthdate;
124
125         // 해당 사용자의 계좌 찾아서 출력
126         bool found = false;
127         for (const auto& account : accounts) {
128             if (account.name == name && account.birthdate == birthdate) {
129                 account.displayResult();
130                 found = true;
131                 break;
132             }
133         }
134
135         if (!found) {
136             cout << "해당하는 사용자의 계좌가 없습니다." << endl;
137         }
138     }
139 }

```

```
145 int main() {
146     int mainMenuChoice;
147
148     do {
149         cout << "-----예금 상품 제작-----" << endl;
150         cout << "1. 예금 상품 만들기 \t 2. 본인 상품 확인하기 \t 3. 추천 상품 확인하기 \t 4. 종료" << endl;
151         cout << "-----" << endl;
152         cout << "하시고자 하는 업무를 선택하여 주세요. ";
153         cin >> mainMenuChoice;
154
155         switch (mainMenuChoice) {
156             case 1:
157                 createDepositProduct();
158                 break;
159             case 2:
160                 checkOwnProduct();
161                 break;
162             case 3:
163                 financialInformation();
164                 break;
165             case 4:
166                 cout << "프로그램을 종료합니다." << endl;
167                 break;
168             default:
169                 cout << "올바르지 않은 선택입니다. 다시 선택하세요." << endl;
170                 break;
171         }
172     } while (mainMenuChoice != 4);
173
174     return 0;
175 }

```

## 2) 테스트 결과

### (1) 예금 상품 제작

- 설명 : 사용자가 입력 상품의 정보를 입력하고 이를 계산하여 출력되는지를 확인하는 테스트임.

- 테스트 결과 스크린샷

-----예금 상품 제작-----

1. 예금 상품 만들기   2. 본인 상품 확인하기   3. 추천 상품 확인하기   4. 종료

-----

하시고자 하는 업무를 선택하여 주세요 : 1

이름을 입력하세요: KimJiung

생년월일을 입력하세요 (YYYYMMDD 형식): 20020118

원금을 입력하세요: 1000000

이자율을 입력하세요 (%): 10

만기 기간을 입력하세요 (월 단위): 24

월 납입액을 입력하세요: 50000

정기예금 계좌 개설이 완료되었습니다.

이름: \343\205KimJiung   생년월일: 20020118

최종 지불액: 2553756.28 원

### (2) 본인 상품 확인

- 설명 : 사용자가 입력한 상품 정보를 확인하기 위해 이름과 생년월일을 입력 후 상품의 정보가 나오는지를 확인한 테스트

- 비고 : 입력한 정보가 파일에 저장되는 과정에서 오류인지, 불러오는 과정에서 발생하는 오류인지를 모르겠지만 이름과 생년월일을 입력 후 해당 사용자의 계좌가 없다는 오류가 발생함. 추후 확인하여 수정할 예정임.

- 테스트 결과 스크린샷

-----예금 상품 제작-----

1. 예금 상품 만들기   2. 본인 상품 확인하기   3. 추천 상품 확인하기   4. 종료

-----

하시고자 하는 업무를 선택하여 주세요 : 2

이름을 입력하세요: KimJiung

생년월일을 입력하세요 (YYYYMMDD 형식): 20020118

해당하는 사용자의 계좌가 없습니다.



### (3) 종료

- 설명 : 프로그램을 종료하기 위해 입력한 값이 잘 구현 되는지 확인하는 테스트
- 테스트 결과 스크린샷

```
-----예금 상품 제작-----  
1. 예금 상품 만들기  2. 본인 상품 확인하기  3. 추천 상품 확인하기  4. 종료  
-----  
하시고자 하는 업무를 선택하여 주세요 : 4  
프로그램을 종료합니다.  
Program ended with exit code: 0
```

## 4. 계획 대비 변경 사항

### 1) 기능 계획 변경

- 이전 : 기능(2) 부분에서 사용자에서 해당 일의 금융 정보를 제공함.
- 이후 : 기능(1) 부분에서 제작한 사용자의 상품을 바탕으로 하여 시중에서 제공하는 실제 금융 상품과 비교한 뒤 사용자의 상품과 유사한 상품을 추천해주는 기능으로 변경함.
- 사유 : 기능(1) 부분에서 사용자가 제작한 상품을 만든 뒤, 실제로 비슷한 상품이 있는지를 확인하는 것이 필요하다고 판단하여 이를 활용한 기능으로 변경함.

## 5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/10	11/17	.....
제안서 작성		완료			
기능1	세부기능1		진행 중		
	세부기능2			----->	
기능2	.....				
업무		11/26	12/3	12/10	12/23
1차 진척 보고서 작성		완료			
2차 진척 보고서 작성			----->		
최종 프로그램 및 보고서				----->	
기능2	세부기능1		진행		