

수업명

프로젝트 제목

최종 보고

제출일자: 2023-11-26

제출자명: 성세현

제출자학번: 213832

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

2018년 스플래시데이터의 통계에 따르면 유출된 비밀번호 중 가장 많이 유출된 비밀번호는 "123456" 이다. 두번째로 많이 유출된 비밀번호는 "password"이다. 많은 비밀번호가 암기의 늪에서 빠져나오지 못하고 단순화된 형태로 바뀌어져 간다. 단순해진 비밀번호는 유출의 위험도와 직결되며 실제로 위와같은 사례들이 그것을 증명한다. 과도하게 단순한 비밀번호가 아니더라도 대부분 사용자 본인에게 상징적이거나 의미 있는 숫자와 문자의 간단한 배열을 통해 비밀번호를 조합하지만 이 또한 유출의 위험이 크다. 개인번호 유출은 피해규모를 산정할 수 없을 뿐더러 유출피해가 발생하였을 때 피해의 책임은 개인정보를 관리하는 피해 기업에게로 전가된다. 이는 기업 실적에 상당한 악영향을 끼치게 된다. 따라서 이 문제를 해결하기 위해 자신만이 알고있는 간단한 정보를 받아 기억하기 쉽고 쉽게 유추할 수 없는 비밀번호를 조합하는 사회적 캠페인과 프로그램의 필요성을 느끼게 되었다.

2) 프로젝트 목표

고객의 개인정보(성별, 나이, 직장) 등을 입력 받아 미리 짜여진 일련의 과정을 거쳐 기억하기 쉬우며 쉽게 유추할 수 없는 비밀번호를 제작하는 것을 목표로 한다.

3) 차별점

기존 프로그램은 비밀번호를 조합하는 과정 속 무작위성이 짙다. 하지만 우리는 해당 비밀번호제작을 의뢰한 고객 본인이라면 쉽게 기억할 수 있도록 구성한다. 고객에게 어떤 정보를 입력할 것인지 선택지를 주고, 해당 입력을 받아 비밀번호를 제작함으로써 비밀번호로써의 완성도를 높인 점에서 차별점이 있다.

1) 기능 1 (정보 입력)

- 비밀번호 제작을 위해 고객에게 몇가지 정보를 입력받는 기능

(1) 세부 기능 1 (정보 선택의 개별화)

- 어떤 정보를 입력할 것인지 선택지를 제시하는 기능

(2) 세부 기능 2 (정보의 입력)

- 고객으로부터 정보를 입력 받는 기능

2) 기능 2 (비밀번호 제작)

- 고객이 입력한 정보를 바탕으로 미리 짜여진 조합방법을 통해 비밀번호를 제작하는 기능

(1) 세부 기능 1 (암호의 보안화)

- 비밀번호 조합 방법을 여러개를 두어 보안을 강화하는 기능

3) 기능 3(비밀번호 설명)

- 고객이 비밀번호를 기억하기 쉽도록 비밀번호에 대한 설명을 첨부하는 기능

3. 진척사항

1) 기능 구현

(1) 구현한 기능 이름 : 정보 입출력 선택

- 고객이 어떤 정보를 입력할지 제시하고, 고객의 질문을 입력받는다.
- 고객의 비밀번호 제작에 필요한 여러 정보를 받기위해 어떤 정보를 입력할것인지 물어보고, 그에 대한 답변을 받는 기능.
- 함수화, 클래스

```

1  #include <iostream>
2  #include <string>
3  #include <vector>
4
5  using namespace std;
6
7  class Password {
8  private:
9      string passWord;
10
11      string inPutWord;
12      int inPutNum;
13
14  public:
15      //고객이 입력한 정보를 바탕으로 미리 짜여진 조합 방법을 통해 비밀번호를 제작하는 기능
16      string CreatePassword(string inPutWord, int inPutNum) {
17
18      }
19      //사용자가 어떤 개인정보를 선택할지 제시하는 함수
20      int makeQuestionWord() {
21          int Wordnum;
22
23
24          cout << "=====" << endl;
25          cout << "*** 영문 질문입니다. (10글자 이내) ***" << endl;
26          cout << "1. 제일 친한 친구 이름은?" << endl;
27          cout << "2. 나의 보물 1호는?" << endl;
28          cout << "3. 내가 존경하는 인물은?" << endl;
29          cout << "4. 내가 초등학교때 들었던 별명은?" << endl;
30          cout << "5. 기타" << endl;
31          cout << "=====" << endl;
32          cout << "다음 중 답변 할 질문을 고르세요 : ";
33          cin >> Wordnum;
34          cout << Wordnum << "번을 고르셨습니다." << endl;
35
36          return Wordnum;
37      }
38      int makeQuestionNum() {
39          int Numnum;
40
41
42          cout << "=====" << endl;
43          cout << "*** 숫자 질문입니다. (8자리 날짜 ex) 20231111) *** " << endl;
44          cout << "1. 부모님의 결혼 기념일은?" << endl;
45          cout << "2. 가까운 형제의 생일은?" << endl;
46          cout << "3. 현재 다니고있는 직장예 처음 입사하게 된 날짜는?" << endl;
47          cout << "4. 추억하고 싶은 날짜가 있다면?" << endl;
48          cout << "5. 기타" << endl;
49          cout << "=====" << endl;
50          cout << "다음 중 답변 할 질문을 고르세요 : ";
51          cin >> Numnum;
52          cout << Numnum << "번을 고르셨습니다." << endl;
53          return Numnum;
54      }

```

```

//답변을 입력받는 함수
string InputPassword() {
    cout << "답변을 입력하세요 : ";
    cin >> inPutWord;
    return inPutWord;
}

int InputPassNum() {
    cout << "답변을 입력하세요 : ";
    cin >> inPutNum;
    return inPutNum;
}

//문자열을 문자 배열로 변환시켜주는 함수
const char* SwitchStringToCharArray(string inPut) {
    inPutWord = inPut;
    return inPutWord.c_str();
}

};

int main() {
    Password obj;
    obj.makeQuestionWord();
    string inPutWord = obj.InputPassword();
    obj.makeQuestionNum();
    int inPutNum = obj.InputPassNum();

    cout << inPutWord << " + " << inPutNum;
}

```

(2) 구현한 기능 이름 : 비밀번호 제작

- 고객이 입력한 정보를 바탕으로 비밀번호를 제작한다
- 비밀번호는 일련의 과정을 통해 제작되며 제작 방식은 보안수준에 따라 다르다
- 고객이 원하는 보안수준을 선택할 수 있게 하여 보안성을 높인다.
- 여러 예외처리를 통해 고객이 입력하는 정보를 제한함으로써 불필요하게 짧거나 긴 비밀번호가 아니라 간편화된 강력한 비밀번호를 제작한다.
- 캡슐화를 통해 private 변수인 password에는 접근하지 못하도록 하여 보안성을 강화시켰다.
- 예외처리, 캡슐화, 매개변수

```
//보안성 하
void CreatePasswordEasy(string inPutWord, int inPutnum) {
    :   // 문자열과 숫자를 합쳐 비밀번호 제작
    :   password = inPutWord + to_string(inPutnum);
    : }
//제작한 비밀번호를 반환하는 함수
string GetPasswordEasy() {
    :   return password;
    : }
}
```

```
// 보안성 중
void CreatePasswordMid(string inPutWord, int inPutNum, int wordQuestionNum, int numQuestionNum) {
    // 문자열의 가장 첫번째 문자는 대문자로 변환
    string PasswordWord = inPutWord;
    PasswordWord[0] = toupper(inPutWord[0]);

    // 문자를 뒤쪽에 입력받은 숫자 삽입
    string inPutNumStr = to_string(inPutNum);
    PasswordWord += inPutNumStr;

    // 입력받은 숫자 뒤쪽에 선택했던 질문에 해당하는 특수문자를 삽입
    char specialChar1 = "!" ;
    char specialChar2 = "@" ;

    switch (wordQuestionNum)
    {
    case 1:
        specialChar1 = "!";
        break;
    case 2:
        specialChar1 = "@";
        break;
    case 3:
        specialChar1 = "#";
        break;
    case 4:
        specialChar1 = "$";
        break;
    case 5:
        specialChar1 = "%";
        break;
    default:
        break;
    }

    switch (numQuestionNum)
    {
    case 1:
        specialChar2 = "!";
        break;
    case 2:
        specialChar2 = "@";
        break;
    case 3:
        specialChar2 = "#";
        break;
    case 4:
        specialChar2 = "$";
        break;
    case 5:
        specialChar2 = "%";
        break;
    default:
        break;
    }

    // 특수 문자를 비밀번호에 추가합니다.
    PasswordWord += specialChar1;
    PasswordWord += specialChar2;

    passWord = PasswordWord;
}
```

```
// 보안성 상
void CreatePasswordHigh(string inPutWord, int inPutNum, int wordQuestionNum, int numQuestionNum) {
    //각수번째 문자 대문자로 변환
    string PasswordWord = inPutWord;
    for (int i = 0; i < inPutWord.length() / 2; i++) {
        PasswordWord[2*i] = toupper(inPutWord[2*i]);
    }

    int reverseInt = reverseInt(inPutNum);
    string inPutNumStr = to_string(reverseInt);

    PasswordWord += inPutNumStr;

    char specialChar1 = "@" ;
    char specialChar2 = "@" ;

    switch (wordQuestionNum)
    {
    case 1:
        specialChar1 = "!";
        break;
    case 2:
        specialChar1 = "@";
        break;
    case 3:
        specialChar1 = "#";
        break;
    case 4:
        specialChar1 = "$";
        break;
    case 5:
        specialChar1 = "%";
        break;
    default:
        break;
    }

    switch (numQuestionNum)
    {
    case 1:
        specialChar2 = "!";
        break;
    case 2:
        specialChar2 = "@";
        break;
    case 3:
        specialChar2 = "#";
        break;
    case 4:
        specialChar2 = "$";
        break;
    case 5:
        specialChar2 = "%";
        break;
    default:
        break;
    }

    // 특수 문자를 비밀번호에 추가합니다.
    PasswordWord += specialChar1;
    PasswordWord += specialChar2;

    passWord = PasswordWord;
}
```

```

while (1) {
    int how;
    cout << "보안수준을 선택하세요. " << endl;
    cout << "1 : 보안수준 하" << endl;
    cout << "2 : 보안수준 중" << endl;
    cout << "3 : 보안수준 상" << endl;
    cout << "보안수준을 선택해주세요(숫자입력) : ";
    cin >> how;
    switch (how)
    {
    case 1:
        obj.CreatePasswordEasy(inPutWord, inPutNum);
        Newpassword = obj.GetPasswordEasy();
        break;
    case 2:
        obj.CreatePasswordMid(inPutWord, inPutNum, wordNum, numNum);
        Newpassword = obj.GetPasswordMid();
        break;
    case 3:
        obj.CreatePasswordHigh(inPutWord, inPutNum, wordNum, numNum);
        Newpassword = obj.GetPasswordHigh();
        break;
    default:
        cout << "유효하지 않은 옵션입니다." << endl;
        continue;
    }
    break;
}

```

(3) 구현한 기능 이름 : 비밀번호 설명

- 보안성 상, 중, 하에 맞추어 비밀번호 제작 과정을 설명합니다.
- 프로그램 종료와 비밀번호 설명 둘 중 하나를 선택할 수 있습니다.

```

void explainPassword(int how) {
    cout << "생성된 비밀번호는 다음과 같은 특징을 가집니다:" << endl;

    switch (how) {
    case 1:
        cout << "- 비밀번호는 입력한 단어와 숫자를 그대로 연결한 형태입니다." << endl;
        break;
    case 2:
        cout << "- 비밀번호는 영문 답변은 대문자로 변환되었고, 숫자 답변과 선택한 질문에 해당하는 특수문자가 추가되었습니다." << endl;
        break;
    case 3:
        cout << "- 비밀번호는 영문 답변의 짝수번째 글자가 대문자로 변환되었고, 숫자 답변은 거꾸로 배치되었으며, 선택한 질문에  

        break;
    default:
        break;
    }
}

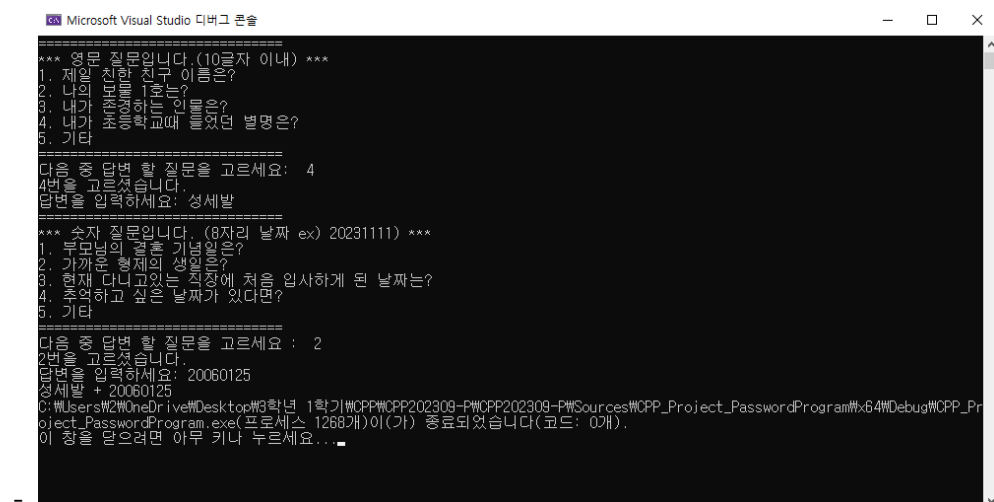
```

```
cout << "프로그램 종료 0번 비밀번호 설명 1번" << endl;
int what;
cin >> what;
if (what == 1) {
    obj.explainPassword(how);
}
else {
    return 0;
}
```

4. 테스트 결과

(1) 테스트한 기능 이름 : 정보 입출력 선택

- 고객이 어떤 정보를 입력할지 설정하고, 정보를 입력받아 저장한다.



(2) 테스트한 기능 이름 : 비밀번호 제작, 보안수준 선택

- 고객이 답변한 정보와 보안수준을 통해 비밀번호를 제작한다.


```

=====
*** 영문 질문입니다.(10글자 이내) ***
1. 제일 친한 친구 이름은?
2. 나의 보물 1호는?
3. 내가 존경하는 인물은?
4. 내가 초등학교때 들었던 별명은?
5. 기타

=====
다음 중 답변 할 질문을 고르세요: 4
4번을 선택하셨습니다.답변을 입력하세요: sehyun
입력 : sehyun
=====
*** 숫자 질문입니다. ***
1. 부모님의 결혼 기념일은?
2. 가까운 형제의 생일은?
3. 현재 다니고있는 직장예 처음 입사하게 된 날짜는?
4. 추억하고 싶은 날짜가 있다면?
5. 기타

=====
다음 중 답변 할 질문을 고르세요: 1
1번을 선택하셨습니다.답변을 입력하세요: 1228
입력 : 1228
보안수준을 선택하세요.
1 : 보안수준 하
2 : 보안수준 중
3 : 보안수준 상
보안수준을 선택해주세요(숫자입력) : 2
비밀번호 생성중입니다.잠시만 기다려주세요.
.
.
.

완성된 비밀번호 : Sehyun1228$!

```

(3) 테스트한 기능 이름 : 비밀번호 설명

```

비밀번호 생성중입니다.잠시만 기다려주세요.
.
.
.
완성된 비밀번호 : Sehyun1228$!
프로그램 종료 0번 비밀번호 설명 1번
1
생성된 비밀번호는 다음과 같은 특징을 가집니다:
- 비밀번호는 영문 답변은 대문자로 변환되었고, 숫자 답변과 선택한 질문에 해당하는 특수문자가 추가되었습니다.

```

5. 계획 대비 변경 사항

1) 변경 내역 제목

- 이전 -

- 이후 -

- 사유 -

6. 느낀 점

- 프로젝트를 처음 계획할 때에는 혼자서 프로젝트를 구상하고 진행한다는 것이 꽤 두렵기도 하고 걱정을 많이 했었다. 하지만 시작해보니 프로젝트 구성에만 꽤 고생하고 한 기능을 담당하는 함수를 차근차근 구현하다 보니 어느새 어느정도 완성되어 있는 프로그램을 발견하였다. 또한 생각보다 예외처리 할 부분이 많다는 것을 느꼈고 이것 또한 앞으로 프로젝트를 해 나가는데에 있어 필수적인 부분이 될것이라는것도 느꼈다. 결론적으로 많은 도움이 된 프로젝트였다.