

〔객체지향프로그래밍〕

가치 텔레콤



가치

황지훈, 이재혁, 송유리

목 차

001

계 획 및 구 상

002

Flowchart

003

코 드 분 석

004

분 석 및 마 무 리

프로젝트 목적 설정

- 오류와 도용 없이 프로그램 구현하기
- 객체지향 기능을 이용, 특징을 살려 설계하기
 - 1) 다종류의 제품을 정리하고, 보기 쉽게 정보 출력하기
 - 2) 사용자에게서 입력 받은 조건에 맞게 제품 추천해 주기
 - 3) 파일 입출력을 통하여 주문서를 작성 받아 텔레콤의 입장에서 확인하기


=> 사용자의 선택을 돕고,

쉬운 비교와 구매 선택을 돕는 것을 목적으로 구성하기

과정 구성

1) 서비스 종류 2) 기기 종류 3) 추천 조건 4) 주문 정보 입력 5) 할부 계산기

INPUT

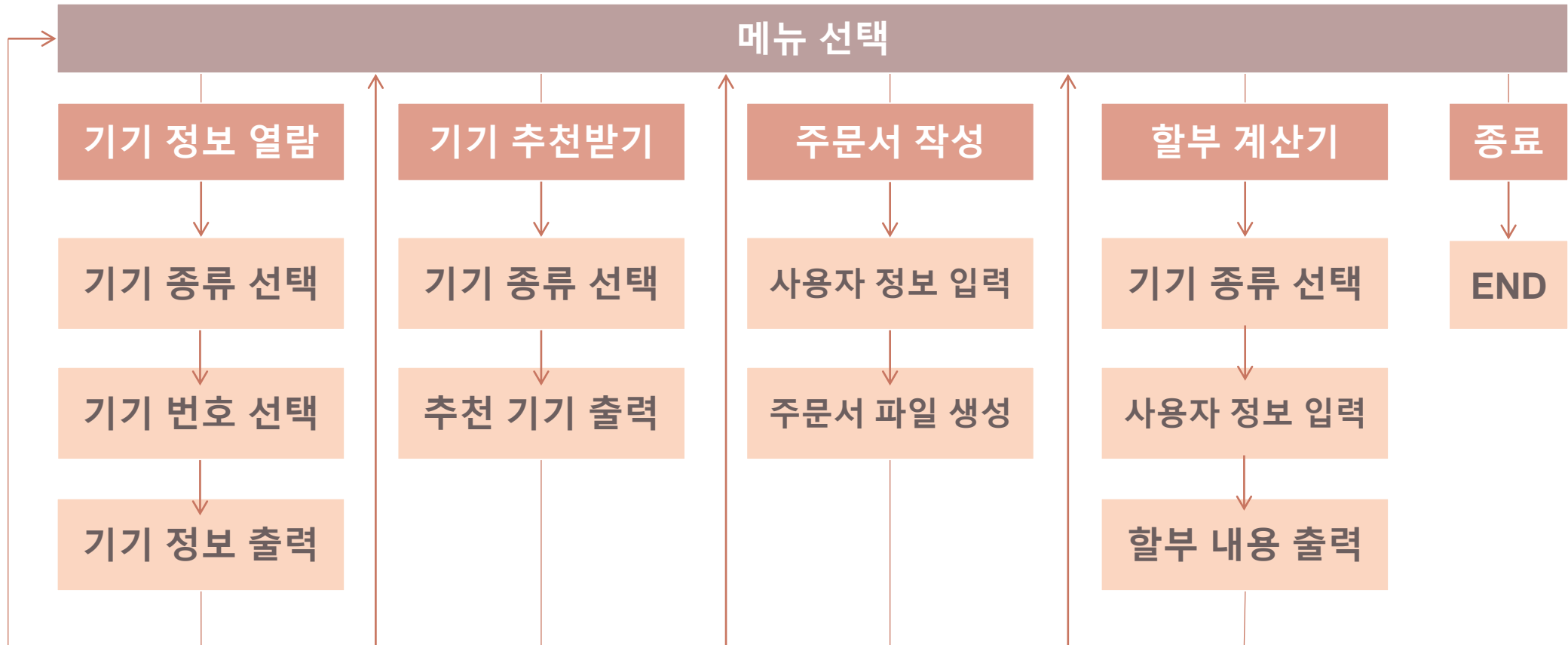
- 
1. 서비스 종류 : 기기 정보 열람, 추천, 주문
 2. 기기 종류 : 출력된 기기 목록 중에서 항목 입력
 3. 추천 조건 : 선택한 기기 종류 내에서 구매 희망 가격대 입력
 4. 주문 정보 입력 : 제품 주문 선택 후 할부 등의 정보 입력
 5. 할부 계산기 : 구매할 가격의 납부 개월 수를 입력



INPUT 선택 및 입력에 따른 출력

OUTPUT

프로그램 흐름도



실행

```

★가치 텔레콤 애플 구매 프로젝트★
손쉽게 애플 기기 정보를 알아보고 구매해 보아요

-----원하시는 메뉴를 선택해주세요-----
1. 기기정보 열람하기
2. 기기 추천받기
3. 주문서 작성
4. 할부계산기
5. 종료하기
원하시는 번호를 선택해주세요 :
  
```

```

int main() {

    int select = 0;
    cout << "      ★가치 텔레콤 애플 구매 프로젝트★      " << endl;
    cout << "손쉽게 애플 기기 정보를 알아보고 구매해 보아요" << endl;

    do {
        menu();
        cout << "원하시는 번호를 선택해주세요 : ";
        cin >> select;
        cout << "-----" << endl;

    } while (select);

}
  
```

```

void menu() { //메인메뉴
    cout << endl << "-----원하시는 메뉴를 선택해주세요-----" << endl;
    cout << "1. 기기정보 열람하기" << endl;
    cout << "2. 기기 추천받기" << endl;
    cout << "3. 주문서 작성" << endl;
    cout << "4. 할부계산기" << endl;
    cout << "5. 종료하기" << endl;
}
  
```

기기 정보

```
void iPhoneShow() {
    cout << "----- 4. iPhone -----" << endl;
    cout << "1. iPhone 12 mini" << endl;
    cout << "2. iPhone 12 " << endl;
    cout << "3. iPhone 12 Pro " << endl;
    cout << "4. iPhone 12 Pro Max" << endl;
}
```

```
class AppleSelect { //애플 기기 종류 클래스
private:
    int num; //private으로 선언하여 멤버변수 보호
public:
    void show() { AppleKind(); } //오버라이딩 함수를 호출하는 동적 바인딩
    virtual void AppleKind() {
        cout << endl << "-----원하시는 기기를 선택해주세요-----" << endl;
        cout << "1. Mac" << endl;
        cout << "2. MacBook" << endl;
        cout << "3. iPad" << endl;
        cout << "4. iPhone" << endl;
        cout << "5. Watch" << endl;
        cout << "6. AirPods" << endl;
        cout << endl << "원하시는 번호를 선택해주세요 : ";
    }
    void AppleList();
};

void AppleSelect::AppleList() { // 파생클래스 메뉴리스트
    cin >> num;
    cout << endl;
    switch (num) {
        int m;
        case 1: {
            MacShow();
            cout << endl << "원하시는 기종의 번호를 입력해주세요 : ";
            cin >> m;
            if (m == 1) { MacShow_1(); break; }
            else if (m == 2) { MacShow_2(); break; }
        }
        case 2: {
```

```
void iPhoneShow_1() {
    ifstream fin("C:\\Users\\skjo6\\OneDrive\\바탕 화면\\Apple\\iPhoneShow_1.txt");
    if (!fin) {
        cout << "C:\\Users\\skjo6\\OneDrive\\바탕 화면\\Apple\\iPhoneShow_1.txt 열기 실패" << endl;
    }
    char buf[400];
    while (true) {
        fin.getline(buf, 400);
        if (fin.eof())
            break;
        cout << buf << endl;
    }
    fin.close();
}

void iPhoneShow_2() {
    ifstream fin("C:\\Users\\skjo6\\OneDrive\\바탕 화면\\Apple\\iPhoneShow_2.txt");
    if (!fin) {
        cout << "C:\\Users\\skjo6\\OneDrive\\바탕 화면\\Apple\\iPhoneShow_2.txt 열기 실패" << endl;
    }
    char buf[400];
    while (true) {
        fin.getline(buf, 400);
        if (fin.eof())
            break;
        cout << buf << endl;
    }
    fin.close();
}
```

기기 정보

iPhoneShow_1.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

아이폰 12 미니 - - - - ₩950,000~

색상 : 블루, 그린, 레드, 화이트, 블랙

용량 : 64GB, 128GB, 256GB

디스플레이 : 5.4형 (Super Retina XDR 디스플레이)

카메라 : 울트라 와이드, 와이드

칩 : A14 Bionic 칩

전원 및 배터리 : 최대 15시간 동영상 재생

방수 기능 : 최대 수심 6m, 최대 30분 생활 방수

기기 정보 열람하기를 선택하셨습니다.

-----원하시는 기기를 선택해주세요-----

1. Mac
2. MacBook
3. iPad
4. iPhone
5. Watch
6. AirPods

원하시는 번호를 선택해주세요 : 4

----- 4. iPhone -----

1. iPhone 12 mini
2. iPhone 12
3. iPhone 12 Pro
4. iPhone 12 Pro Max

원하시는 기종의 번호를 입력해주세요 : 1

아이폰 12 미니 - - - - ₩950,000~

색상 : 블루, 그린, 레드, 화이트, 블랙

용량 : 64GB, 128GB, 256GB

디스플레이 : 5.4형 (Super Retina XDR 디스플레이)

카메라 : 울트라 와이드, 와이드

칩 : A14 Bionic 칩

전원 및 배터리 : 최대 15시간 동영상 재생

방수 기능 : 최대 수심 6m, 최대 30분 생활 방수

기기 추천받기

```
class Mac : public recommend { //상속 클래스
public:
    void Mac1() { cout << endl << "★ 가격대에 맞는 추천 기기 : iMac ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void Mac2() { cout << endl << "★ 가격대에 맞는 추천 기기 : iMac Pro ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
};

class MacBook : public recommend { //상속 클래스
public:
    void MacBook1() { cout << endl << "★ 가격대에 맞는 추천 기기 : MacBook Air ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void MacBook2() { cout << endl << "★ 가격대에 맞는 추천 기기 : MacBook Pro 16 ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
};

class iPad : public recommend { //상속 클래스
public:
    void iPad1() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPad mini ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPad2() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPad Air ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPad3() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPad ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPad4() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPad Pro ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
};

class iPhone : public recommend { //상속 클래스
public:
    void iPhone1() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPhone 12 mini ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPhone2() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPhone 12 ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPhone3() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPhone 12 Pro ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
    void iPhone4() { cout << endl << "★ 가격대에 맞는 추천 기기 : iPhone 12 Pro Max ★" << endl << "구매를 원하시면 ↓↓↓ 3번을 입력해주세요." << endl; }
};
```

```
class recommend {
public:
    void reco() {
        cout << endl << "어떠한 기종을 추천 받으시겠습니까? " << endl;
    }
};
```

```

if (n == 3) {
    array[0] = 499000, array[1] = 779000, array[2] = 449000, array[3] = 1299000;
    for (i = 0; i < 10; i++) {
        if (i == 0) {
            tmp = abs(p - array[i]);
            result = 0;
        }
        else {
            tmp2 = abs(p - array[i]);
            if (tmp2 < tmp) {
                tmp = tmp2;
                result = i;
            }
        }
    }
    result = result + 1;

    switch (result) {
        case 1: c.reco(), c.iPad1(); break;
        case 2: c.reco(), c.iPad2(); break;
        case 3: c.reco(), c.iPad3(); break;
        case 4: c.reco(), c.iPad4(); break;
    }
}

if (n == 4) {
    array[0] = 950000, array[1] = 1090000, array[2] = 1350000, array[3] = 1490000;
    for (i = 0; i < 10; i++) {
        if (i == 0) {
            tmp = abs(p - array[i]);
            result = 0;
        }
        else {
            tmp2 = abs(p - array[i]);
            if (tmp2 < tmp) {
                tmp = tmp2;
                result = i;
            }
        }
    }
    result = result + 1;

    switch (result) {
        case 1: d.reco(), d.iPhone1(); break;
        case 2: d.reco(), d.iPhone2(); break;
        case 3: d.reco(), d.iPhone3(); break;
        case 4: d.reco(), d.iPhone4(); break;
    }
}

```

```

(select == 2) {
    cout << endl << "기기 추천받기를 선택하셨습니다." << endl << endl;

    Mac a; MacBook b; iPad c; iPhone d; Watch e; AirPods f; //파생클래스
    int n, p, i, result = 0;
    int tmp = 0, tmp2 = 0;
    int array[10];
    cout << "1. Mac" << endl << "2. MacBook" << endl << "3. iPad" << endl << "4. iPhone"
        << endl << "5. AppleWatch" << endl << "6. Airpod" << endl << endl;
    cout << "어떠한 기종을 추천 받으시겠습니까? : ";
    cin >> n;
    cout << "생각하고 계산 가격대를 입력해주세요 : ";
    cin >> p;

    if (n == 1) {
        array[0] = 1440000, array[1] = 6300000;
        for (i = 0; i < 10; i++) {
            if (i == 0) {
                tmp = abs(p - array[i]);
                result = 0;
            }
            else {
                tmp2 = abs(p - array[i]);
                if (tmp2 < tmp) {
                    tmp = tmp2;
                    result = i;
                }
            }
        }
        result = result + 1;

        switch (result) {
            case 1: a.reco(), a.Mac1(); break;
            case 2: a.reco(), a.Mac2(); break;
        }
    }

    if (n == 2) {
        array[0] = 1290000, array[1] = 3190000;
        for (i = 0; i < 10; i++) {
            if (i == 0) {
                tmp = abs(p - array[i]);
                result = 0;
            }
            else {
                tmp2 = abs(p - array[i]);

```

기기 추천받기

원하시는 번호를 선택해주세요 : 2

기기 추천받기를 선택하셨습니다.

1. Mac
2. MacBook
3. iPad
4. iPhone
5. AppleWatch
6. Airpod

어떠한 기종을 추천 받으시겠습니까? : 5
생각하고 계신 가격대를 입력해주세요 : 300000

어떠한 기종을 추천 받으시겠습니까?

★ 가격대에 맞는 추천 기기 : AppleWatch 3 ★
구매를 원하시면 ↓↓↓ 3번을 입력해주세요.

원하시는 번호를 선택해주세요 : 2

기기 추천받기를 선택하셨습니다.

1. Mac
2. MacBook
3. iPad
4. iPhone
5. AppleWatch
6. Airpod

어떠한 기종을 추천 받으시겠습니까? : 3
생각하고 계신 가격대를 입력해주세요 : 600000

어떠한 기종을 추천 받으시겠습니까?

★ 가격대에 맞는 추천 기기 : iPad mini ★
구매를 원하시면 ↓↓↓ 3번을 입력해주세요.

주문서 작성

```

원하시는 메뉴를 선택해주세요 :-----
1. 기기 정보 열람하기
2. 기기 추천받기
3. 주문서 작성
4. 할부계산기
5. 종료하기
원하시는 번호를 선택해주세요 : 3

-----주문서 작성-----
이름>>황지훈
배송지>>한밭대학교N4동
구매할 기종>>애플워치SE

-----
입력하신 주문이 접수되었습니다. 감사합니다.

-----원하시는 메뉴를 선택해주세요 :-----
1. 기기 정보 열람하기
2. 기기 추천받기
3. 주문서 작성
4. 할부계산기
5. 종료하기
원하시는 번호를 선택해주세요 :

```

customer - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V)

구매자 이름 : 황지훈
 배송지 : 한밭대학교N4동
 구매할 기종 : 애플워치SE

```

if (select == 3) {
    char name[10], add[30], apple[10];

    cout << endl << "-----주문서 작성-----" << endl;
    cout << "이름>>";
    cin >> name;
    cout << "배송지>>";
    cin >> add;
    cout << "구매할 기종>>";
    cin >> apple;

    ofstream fout("C:\\Users\\skjo6\\OneDrive\\바탕 화면\\Apple\\customer.txt");
    if (!fout) {
        cout << "customer.txt 파일을 열 수 없습니다. ";
        return 0;
    }
    fout << "구매자 이름 : " << name << endl;
    fout << "배송지 : " << add << endl;
    fout << "구매할 기종 : " << apple << endl;

    cout << endl << "-----" << endl;
    cout << "입력하신 주문이 접수되었습니다. 감사합니다." << endl;
    cout << "-----" << endl;
    fout.close();
}

```

할부 계산기

```
class Average { //기본클래스
private: //private으로 선언하여 멤버변수 보호
    int n, s;
public:
    void input();
};

void Average::input() { //정수형 배열의 동적 할당 및 반환 함수
    cout << endl << "----- 할부계산기 -----" << endl;
    cout << " 몇 개의 기기를 구매하시겠습니까? ";
    cin >> n;
    if (n <= 0) cout << "유효하지않은 값입니다." << endl;
    int* p = new int[n]; // 정수 배열 동적 할당
    if (!p) cout << "메모리를 할당할 수 없습니다.";
    for (int i = 0; i < n; i++) {
        cout << " 구매하시는 기종의 가격을 입력해주세요 : ";
        cin >> p[i];
    }
    cout << " 몇 개월 할부로 구매하시겠습니까? : ";
    cin >> s;
    float sum = 0; //sum값 0으로 초기화
    for (int i = 0; i < n; i++)
        sum += p[i];
    cout << endl << "한달에 납부하셔야되는 금액은 " << sum / s << "원입니다." << endl;
    delete[] p; //배열 메모리 반환
}
```

```
if (select == 4) {
    Average input;
    input.input();
}
```

```
void Average::input() { //정수형 배열의 동적 할당 및 반환 함수
    cout << endl << "----- 할부계산기 -----" << endl;
    cout << " 몇 개의 기기를 구매하시겠습니까? ";
    cin >> n;
    if (n <= 0) cout << "유효하지않은 값입니다." << endl;
    int* p = new int[n]; // 정수 배열 동적 할당
    if (!p) cout << "메모리를 할당할 수 없습니다.";
    for (int i = 0; i < n; i++) {
        cout << " 구매하시는 기종의 가격을 입력해주세요 : ";
        cin >> p[i];
    }
    cout << " 몇 개월 할부로 구매하시겠습니까? : ";
    cin >> s;
    float sum = 0; //sum값 0으로 초기화
    for (int i = 0; i < n; i++)
        sum += p[i];
    cout << endl << "한달에 납부하셔야되는 금액은 " << sum / s << "원입니다." << endl;
    delete[] p; //배열 메모리 반환
}
```

```
-----원하시는 메뉴를 선택해주세요-----
1. 기기정보 열람하기
2. 기기 추천받기
3. 주문서 작성
4. 할부계산기
5. 종료하기
원하시는 번호를 선택해주세요 : 4
```

```
----- 할부계산기 -----
몇 개의 기기를 구매하시겠습니까? 1
구매하시는 기종의 가격을 입력해주세요 : 300000
몇 개월 할부로 구매하시겠습니까? : 12

한달에 납부하셔야되는 금액은 25000원입니다.
```

종료하기

```
if (select == 5) {  
    cout << endl << "종료하기를 선택하셨습니다." << endl<<endl;  
    cout << "지금까지 가치텔레콤을 이용해주셔서 감사합니다." << endl;  
    exit(0);  
}  
}while(select);  
}
```

원하시는 번호를 선택해주세요 : 5

종료하기를 선택하셨습니다.

지금까지 가치텔레콤을 이용해주셔서 감사합니다.
계속하려면 아무 키나 누르십시오 . . .

객체지향 프로그래밍

- 객체지향 언어

- Class를 사용하여 멤버 설정

- Private과 public을 사용하여 멤버 변수 접근 및 보호

- 동적 바인딩

- 파생 클래스 사용

- ect

- 동적 할당 및 반환

- 파일 호출 및 생성



Thank You
감사합니다