

# 컴퓨터 견적 프로그램 구현

김동진

# Q1 순서도



## 주요 코드설명(1) - 검색조건 외 데이터제거

```

1  public static void searchListCPUCompany(String searchWord) {
2
3      for (int i = (db.cpuList.size()-1); i > -1; i--) {
4          String character = db.cpuList.get(i).getCompany();
5          if (character.contains(searchWord) == false) { db.cpuList.remove(i); }
6
7      for (int i=0; i<db.cpuList.size(); i++) {
8          System.out.printf("(" + (i+1) + "번)" + " / ");
9          System.out.printf("제조사 : " + db.cpuList.get(i).getCompany() + " / ");
10         System.out.printf("제조사 : " + db.cpuList.get(i).getCompany() + " / ");
11         System.out.printf("제품명 : " + db.cpuList.get(i).getName() + " / ");
12         System.out.printf("코어수 : " + db.cpuList.get(i).getCore() + "개 / ");
13         System.out.printf("기본클럭 : " + db.cpuList.get(i).getMin() + "GHz / ");
14         System.out.printf("최대클럭 : " + db.cpuList.get(i).getMax() + "GHz / ");
15         System.out.printf("내장그래픽 유무 : " + db.cpuList.get(i).getInner() + " / ");
16         int price = db.cpuList.get(i).getPrice();
17         System.out.println("가격 : " + DF.format(price) + "원");
18     }
19     System.out.println(" ");
20     if(db.cpuList.isEmpty() == true) {
21         System.out.println("# 검색 정보가 없습니다. 다시 조회합니다.");
22         System.out.println(" ");
23         choiceCPU();
24     }
25 }

```

□ (3~5)행

- 순회 간 데이터 일치시 리스트 내용 삭제

□ (7~18)행

- 일치하는 데이터 리스트 출력

□ (20~23)행

- 조회되는 데이터 없을 경우 조회 메서드 호출로 초기화

## 주요 코드설명(2) - 견적서(quote) 테이블 내 값 입력

```

1      case 2:
2          System.out.println("# 선택하실 제품번호를 입력하세요.");
3          int selectNum = sc.nextInt();
4          System.out.println("# 위 제품을 견적서에 추가하시겠습니까?");
5          showCPU(selectNum);
6          System.out.println("① 예");
7          System.out.println("② 아니오");
8          choice = sc.nextInt();
9          switch (choice) {
10         case 1:
11             System.out.println("# 선택하신 제품을 견적서에 추가했습니다.");
12             String name = db.cpuList.get(selectNum-1).getName();
13             db.quote.setCPUName(name);
14             int number = db.cpuList.get(selectNum-1).getPrice();
15             db.quote.setCPUPrice(number);
16
17             check = false;
18             break;

```

□ (3~5)행

- 사용자가 입력한 정수값을 selectNum에 저장
- 사용자가 입력한 정수값을 인덱스로 활용  
showCPU에서 선택한 부품정보 출력

□ (10~18)행

- 사용자가 입력한 selectNum 값을  
부품List의 인덱스값으로 활용, 이름,  
가격 정보 추출

## 주요 코드설명(2) - 견적서(quote) 테이블 내 값 입력

```

1  public void insertQuote() {
2      String SQL = "INSERT INTO QUOTE(QUOTENAME, CPUName, CPUPRICE, MBNAME, MBPRICE, VGANAME,
3      VGAPRICE, RAMNAME,
4      + "RAMPRICE, SSDNAME, SSDPRICE)" + "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
5
6      try {
7          pstmt = con.prepareStatement(SQL);
8
9          String quoteName = quote.getQuoteName();
10         String cpuName = quote.getCPUName();
11         int cpuPrice = quote.getCPUPrice();
12         String mbName = quote.getMbName();
13         int mbPrice = quote.getMbPrice();
14         String vgaName = quote.getVGAName();
15         int vgaPrice = quote.getVGAPrice();
16         String ramName = quote.getRAMName();
17         int ramPrice = quote.getRAMPrice();
18         String ssdName = quote.getSSDName();
19         int ssdPrice = quote.getSSDPrice();
20
21         pstmt.setString(1, quoteName);
22         pstmt.setString(2, cpuName);
23         pstmt.setInt(3, cpuPrice);
24         pstmt.setString(4, mbName);
25         pstmt.setInt(5, mbPrice);
26         pstmt.setString(6, vgaName);
27         pstmt.setInt(7, vgaPrice);
28         pstmt.setString(8, ramName);
29         pstmt.setInt(9, ramPrice);
30         pstmt.setString(10, ssdName);
31         pstmt.setInt(11, ssdPrice);
32
33         System.out.println("# 견적이 성공적으로 저장되었습니다.");
34         pstmt.executeUpdate();
35     } catch (SQLException e) {
36         e.printStackTrace();
37     }
}

```

□ (8~18)행

- 각 부품별 get메서드로 가져온 값을 각 변수에 저장

□ (20~33)행

- 각 변수를 SQL 문에 삽입 간 활용

```

1  public static void inquiryQuote() {
2      if (update.quoteList.isEmpty()) {
3          System.out.println("# 저장된 견적서가 없습니다. 초기화면으로 돌아갑니다.");
4          System.out.println(" ");
5          main.menuWhile();
6      } else {
7          quoteChoice = sc.nextInt();
8
9          System.out.println("# 견적서 이름 : " + update.quoteList.get(quoteChoice - 1).getQuoteName());
10         System.out.printf("CPU : " + update.quoteList.get(quoteChoice - 1).getCPUName() + " | ");
11         int cpuPrice = update.quoteList.get(quoteChoice - 1).getCPUPrice();
12         System.out.println("가격 : " + DF.format(cpuPrice) + "원");
13         System.out.printf("MAINBOARD : " + update.quoteList.get(quoteChoice - 1).getMbName() + " | ");
14         int mbPrice = update.quoteList.get(quoteChoice - 1).getMbPrice();
15         System.out.println("가격 : " + DF.format(mbPrice) + "원");
16         System.out.printf("VGA : " + update.quoteList.get(quoteChoice - 1).getVGAName() + " | ");
17         int vgaPrice = update.quoteList.get(quoteChoice - 1).getVGAPrice();
18         System.out.println("가격 : " + DF.format(vgaPrice) + "원");
19         System.out.printf("RAM : " + update.quoteList.get(quoteChoice - 1).getRAMName() + " | ");
20         int ramPrice = update.quoteList.get(quoteChoice - 1).getRAMPrice();
21         System.out.println("가격 : " + DF.format(ramPrice) + "원");
22         System.out.printf("SSD : " + update.quoteList.get(quoteChoice - 1).getSSDName() + " | ");
23         int ssdPrice = update.quoteList.get(quoteChoice - 1).getSSDPrice();
24         System.out.println("가격 : " + DF.format(ssdPrice) + "원");
25         System.out.println("총 금액 : " + DF.format(cpuPrice + mbPrice + vgaPrice + ramPrice + ssdPrice) + "원");
26         System.out.println(" ");
27     }
28 }

```

□ DB\_Connection 클래스에서 DB서버를  
통해 받아온 각 부품의 이름과 가격 데이터를  
quoteList에 저장

□ (2~5)행

- quoteList에서 저장된 값이 없을 시  
초기화면 안내

□ (6~28)행

- static 전역변수로 quoteChoice 클래스  
생성 시 선언 → 추후 견적서 수정시에 이용
- quoteList로 부터 받아온 데이터 출력

```

1  case 2:
2      System.out.println("# 선택하실 제품번호를 입력하세요.");
3      int selectNum = sc.nextInt();
4      System.out.println("# 아래 제품으로 수정하시겠습니까?");
5      showCPU(selectNum);
6      System.out.println("① 예");
7      System.out.println("② 아니오");
8      choice = sc.nextInt();
9      switch (choice) {
10         case 1:
11             System.out.println("# 선택하신 제품으로 견적서가 수정되었습니다.");
12             String name = update.cpuList.get(selectNum - 1).getName();
13             int number = update.cpuList.get(selectNum - 1).getPrice();
14             String quoteName = update.quoteList.get(quoteChoice - 1).getQuoteName();
15             update.updateDbCPU(name, number, quoteName);
16
17             check = false;
18             break;
19         case 2:
20             System.out.println("# 제품을 다시 조회합니다.");
21             for (int i = (update.cpuList.size() - 1); i > -1; i--) {
22                 update.cpuList.remove(i);
23             }

```

□ (10~18)행 (case 1)

- 사용자가 입력한 selectNum을 부품List의 추출할 데이터 값 인덱스로 활용
- name, number 는 각 리스트로부터 추출된 데이터 저장 및 DB서버 접근 변수로 활용
- quoteName은 quoteList에서 견적서 상세조회에 사용된 quoteChoice를 인덱스로 활용

□ (6~28)행

- static 전역변수로 quoteChoice 클래스 생성 시 선언 → 추후 견적서 수정시에 이용
- quoteList로 부터 받아온 데이터 출력

## 02

### 주요 코드설명(4) - 견적서 수정

```
1  public void updateDbCPU(String name, int price, String quoteName) {
2      String SQL = "UPDATE QUOTE SET CPUNAME = ?, CUPPRICE = ? WHERE QUOTENAME = ?";
3
4      try {
5          pstmt = con.prepareStatement(SQL);
6          pstmt.setString(1, name);
7          pstmt.setInt(2, price);
8          pstmt.setString(3, quoteName);
9          pstmt.executeUpdate();
10     } catch (SQLException e) {
11         e.printStackTrace();
12     }
13 }
```

04

04 견적서 수정 4



# Q&A

Q4

Q4 2016