

# 코드정의서

- 피자 자동화 가게 시스템

1 조

# 목 차

1. 개요.....	
1.1. 목적.....	
2. 공통 법칙.....	
2.1. 작업 디렉토리.....	
2.2 소스 파일 .....	
3. 명명법.....	
3.1. 공통 규칙 .....	
3.2 클래스명 .....	
3.3. 메소드명 .....	
3.4 필드명.....	
4. 주석문.....	
4.1. 공통 주석문 .....	
4.2 멤버 필드 주석문.....	
4.3. 멤버 메소드 주석문 .....	
4.4. 기타 주석문 .....	
5. 기타 스타일 .....	
5.1. 들여쓰기 .....	

5.2. 선언.....	
5.2.1. 패키지 / 임포트.....	
5.2.2. 클래스 / 인터페이스.....	
5.3. 공백.....	
5.4. 제어문 및 반복문.....	
5.4.1. if.....	
5.4.2. for.....	
5.4.3. while.....	
5.4.4. switch.....	
5.4.5. try / catch / finally.....	
6. 출력 형식.....	
6.1. 선택지 출력 형식.....	
6.2. 리스트 출력 형식.....	

## 1. 개요

### 1.1. 목적

소프트웨어를 개발하는 모든 과정에 들어가는 비용 중 80%가 유지보수에 쓰여진다. 소프트웨어의 직접 개발한 개발자가 그 소프트웨어의 유지보수를 담당하는 경우는 거의 볼 수 없는 경우다. 규칙을 정하고 코드를 작성하게 되면 가독성이 높아져 다른 개발자가 소스 코드를 처음 보더라도 더 빠르고 확실하게 이해할 수 있게 된다. 따라서 규칙을 정하고 지킴은 개발자 간 반드시 지켜야 할 약속임을 항상 인지해야 한다.

## 2. 공통 법칙

### 2.1. 작업 디렉토리

프로젝트, 패키지, 모듈, 클래스 형태로 형성한다.

### 2.2 소스 파일

2.2.1. 각 자바 소스 파일은 하나의 Public Class 혹은 Public Interface 를 포함한다.

2.2.2. 하나의 Method 혹은 Class 에서 사용될 객체의 선언 및 생성은 상단에 위치시킨다.

## 3. 명명법

### 3.1. 공통 규칙

두 개 이상의 단어로 이루어져 있는 고유명사의 경우 첫번째 단어를 제외, 시작 문자를 대문자로 한다.

### 3.2 클래스명

3.2.1. 단어의 첫 번째 글자마다 대문자를 사용한다.

3.2.2. 클래스의 이름은 반드시 명사여야 한다.

3.3.3. 약자로 나타내는 것은 반드시 피하고, 전체 이름을 사용한다.

### 3.3. 메소드명

메소드의 이름은 주로 동사이며 두 개 이상의 단어로 구성될 경우 첫번째 단어를 소문자인 동사로 하고, 이후 각 단어의 시작 문자는 대문자로 한다.

Ex. viewCart(), selectCart() 등

### 3.4 필드명

3.4.1. 변수, 모든 인스턴스 등의 첫번째 글자는 소문자로 하고, 이후 각 단어의 시작 문자는 대문자로 한다.

3.4.2. 클래스 변수의 이름은 타입의 이름과 동일하게 지정한다.

3.4.3. select, count 등은 입력 값, 제어문 혹은 반복문의 임시 변수명으로 사용할 수 있다.

## 4. 주석문

### 4.1. 공통 주석문

클래스의 주석문은 사용하지 않는다.

### 4.2 멤버 필드 주석문

멤버 필드 주석문은 라인 단위 `/**` 주석으로 한다.

### 4.3. 멤버 메소드 주석문

멤버 메소드의 주석문은 필요에 따라 작성한다.

### 4.4. 기타 주석문

코드 작성 중 설명이 필요한 부분의 경우 라인 단위 주석을 통해 기입한다.

## 5. 기타 스타일

### 5.1. 들여쓰기

5.1.1. 들여쓰기는 이클립스 프로그램 내 자동 정렬 기능(`ctrl+shift+F`)을 사용한다.

5.1.2. `{` 기호는 메소드 이름과 같은 줄에 위치하고, `}` 기호는 다른 줄에 위치하며 해당 줄에는 주석을 제외한 어떤 코드도 위치할 수 없다.

Ex.

```
public void setTime() {  
    System.out.println("");  
}
```

### 5.2. 선언

### 5.2.1. 패키지 / 임포트

필요에 따라 Package 문이 나타날 수 있다. Package 문 이후 한 줄을 띄우고 다음으로 import 문이 위치하도록 한다. Import 의 경우 한 줄에 하나의 import 만 명시한다.

### 5.2.2. 클래스 / 인터페이스

Import 문 이후 class 혹은 interface 선언문을 명시한다. class 혹은 interface 선언문의 구성은 다음과 같은 순서로 나타낸다.

5.2.2.1. class/interface 주석

5.2.2.2. class/interface 선언문

5.2.2.3. class/interface 에 속하는 멤버 필드 (private, protected, public 순으로 선언)

5.2.2.4. class 객체 생성자

5.2.2.5. class method 주석

5.2.2.6. class method 객체에서 사용될 메소드

## 5.3. 공백

조건문, 제언문의 키워드 (if, while, return, switch, for 등)와의 관계

5.3.1. 시작 시 '(', '{'의 괄호 앞에 한 칸의 빈칸을 포함하여 작성한다.

5.3.2. 종료 시 ')', '}'의 괄호 뒤에는 주석을 제외한 어떠한 코드도 작성하지 않는다.

## 5.4. 제어문 및 반복문

### 5.4.1. if

If-else statement 에서 else 는 같은 줄에 위치한다.

Ex.

```
if () {  
} else if () {  
} else {  
}
```

### 5.4.2. for

for 문에서 '{'는 같은 줄에, '}'는 새로운 줄에 위치한다.

Ex.

```
for () {
```

```
}
```

### 5.4.3. while

while 문에서 '{'는 같은 줄에, '}'는 새로운 줄에 위치한다.

Ex.

```
while (0) {  
}
```

### 5.4.4. switch

switch 문에서 실행 코드와 'break;'는 새로운 줄에 위치한다.

Ex.

```
switch () {  
case 1:  
    method();  
    break;  
case 2:  
    method2();  
    break;  
default:  
}
```

### 5.4.5. try / catch / finally

try-catch-finally 문에서 '{'는 같은 줄에, '}'는 새로운 줄에 위치한다.

Ex.

```
try {  
} catch () {  
} finally {  
    try {  
    } catch () {  
    }  
}
```

## 6. 출력 형식

### 6.1. 선택지 출력 형식

프로그램 실행 시 출력되는 선택지는 '1. 선택지 1 | 2. 선택지 2 | 3. 선택지 3'의 양식을 따른다.

### 6.2. 리스트 출력 형식

프로그램 실행 시 출력되는 리스트는 출력 값을 하나의 줄로 나열하고, tab 을 사용하여 구분할 수 있도록 한다.

