|  |
| --- |
| Moduel Enterprise 0.1  개발가이드 명명규칙  Version 0.1  2020.06.06 |

|  |
| --- |
|  |

제·개정 이력

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 제·개정일 | Version | 제·개정 내용1) | 개정위치 | 작성자 | 검토자 | 승인자 |
| 2020.06.06 | 0.1 | 신규작성 |  | 정훈 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1) 제 ·개정 내용 : 최초작성, 수정, 추가, 삭제로 구분하여 작성함. 수정, 추가, 삭제의 경우 괄호 안에 해당 내용 기술함.

목차

1. [1.개요 14](#_Toc362876380)
2. [2.명명규칙표준 57](#_Toc362876387)

# 개요

## 프로젝트 준수사항

1. **제시된 표준은 철저히 준수한다.**
2. **새로운 정의가 추가로 필요한 경우는 본 자료를 작성한 담당자와 협의하여 해당 내용을 정의하고, 본 표준서에 수록한 후 프로젝트팀 전체에 공지한다.**
3. **모든 화면은 최대 응답속도 5초 이내를 목표로 한다.**
4. **표준을 준수하지 못할 경우에는 프로젝트 팀원과 협의 하여야 하며 인가를 받고 주석으로 자세한 내용을 표기한다.**
5. **본 표준안은 최소한의 필수사항을 지키도록 유도하고 해서는 안되는 것들을 지적하는 내용이며, 개개인의 다양한 능력과 개념들로 표준을 확장시켜 시스템에 적용시키도록 한다.**

# 명명규칙 표준

* 시스템을 개발하는데 있어 표준 Naming Rule을 적용하여 개발자 및 운영자가 분석 및 코딩하는데 있어 좀더 쉽게 접근할 수 있도록 표준 Naming Rule을 적용한다

## 시스템개발코드

1. 시스템 영역별 코드를 분류한다 (한가지 프로젝트에서 여러 시스템을 개발할경우)

Ex) springModule , springModuleXml

## Package Java class, Java archive 명명 규칙

1. 모든 Java class는 영문 대소문자를 혼용할 수 있지만, 반드시 명사를 사용하고 시작 글자를 대문자로 지정하며 camel case에 준하여 작성한다 \_ 사용 (X)
2. 기능명이 postFix에 포함될 수 있으면 포함시킨다.

Ex) StringUtil.java, PortalBeansFactory.java

* Data Access Object(DAO)
* 인터페이스(IF)
* Batch(Bat)
* 유틸리티(Util)
* 공통(Comm)
* 구현체(Impl)
* 서비스(Service)
* 컨트롤러(Controller)

## 변수명, 메소드명 명명 규칙

1. 변수의 이름은 대소문자를 혼용할 수 있지만 반드시 소문자로 시작한다.

* account
* eventHandler

1. 상수(final 변수)를 표현하는 이름은 반드시 모두 대문자로 지정하되 ‘\_’

를 사용하여 단어들을 구분한다.

* MAX\_ITERATIONS
* COLOR\_RED

1. 메소드의 이름은 대소문자를 혼용할 수 있지만 반드시 동사를 사용하며 소문자로 시작한다.

* getName()
* computeTotalWidth()

1. 축약형(Abbreviation) 과 두문자어형(頭文字語: Acronym)을 이름에 사용할 경우에는 전부 대문자로 지정하지 않는다.

* exportHtmlSource() // exportHTMLSource() - (X)
* openDvdPlayer() // openDVDPlayer() - (X)

1. 일반적인 변수의 이름은 타입의 이름과 동일하게 지정한다.

* void setTopic (Topic topic)
* void setTopic (Topic value) - (X)
* void setTopic (Topic aTopic) - (X)

1. 모든 이름은 영어로 작성한다.
2. 넓은 범위에 영향을 미치는 변수는 긴 이름을 부여하고, 좁은 범위의 변수는 짧은 이름을 부여한다.

* 임시 저장공간이나 인덱스로 사용되는 Scratch variable(loop문내에서의 i, j, k 등과 같은)들은 매우 짧은 이름을 부여하도록 한다.
* 프로그래머가 그러한 변수들을 읽음과 동시에, 이 변수는 몇 라인 뒤에 그 값이 유효하지 않을 것임을 짐작할 수 있게 해야 한다

1. 호출하려는 객체의 이름을 통해 의미를 짐작할 수 있다면, 메소드의 이름을 간략화 할 수 있다.

* line.getLength()
* line.getLineLength() - (X)

1. get/set 이라는 용어는 반드시 애트리뷰트에 직접 접근하는 메소드에 사용한다.

* employee.getName()

1. is 접두사를 불리언 변수와 메소드에 사용한다.

Ex) isSet, isVisible, isFinished, isFound, isOpen

* 불리언 메소드와 변수에 대한 명명 관례는 java 핵심 패키지들의 명명규칙을 준수해야 한다.
* is 접두사를 사용함으로써 일반적으로 status 와 flag 와 같은 좋지 않은 불리언 이름을 선택하는 문제를 해결할 수 있다.
* isStatus나 isFlag 는 간결하지만 프로그래머에게 의미를 풍부하게 전달할 수 없다는 점에서 바람직하지 못합니다.
* 일부사항에서는 is 접두사가 아닌 보다 더 적합한 접두사를 사용한다.

has, can, should접두사

Ex) **boolean** hasLicense(), **boolean** canEvaluate(), **boolean** shouldAbort = **false**

1. compute 라는 용어는 무엇인가를 계산(시간이 소요되는)하는 메소드에 사용할 수 있다.

* valueSet.computeAverage()
* matrix.computeInverse()

1. find 라는 용어는 무엇인가를 찾는 메소드에 사용할 수 있다.
2. initialize 라는 용어는 객체나 개념이 확립되어지는 곳에 사용 할 수 있다.
3. 컬렉션의 이름은 반드시 복수형으로 사용한다. Collection와 단순 배열
4. n의 접두사는 객체의 개수를 나타내는 변수에 사용한다. ex)nPoints, nLines
5. No 접미사는 엔티티 번호를 나타내는 변수에 사용한다.

Ex) tableNo

1. 대응하는 단어가 있는 이름은 반드시 함께 사용한다.

* get/**set**, **add**/remove, **create**/destroy, **start**/stop, **insert**/**delete**, increment/decrement, old/new, **begin**/**end**, **first**/**last**, up/down, **min**/**max**, **next**/previous, old/new, **open**/**close**, **show**/hide

1. 축약형 이름의 사용은 피한다.

* computeAverage() // compAvg() - (X)

1. 불리언 변수 이름은 절대로 부정적인 이름을 사용하지 않는다.

- **boolean** isError // isNotError - (X) \* **boolean** isFound // isNotFound - (X)

1. Package 규칙.

* com 밑에 버전별로 패키지를 구성한다 Ex) com.demo1
* 버전별 밑으로 업무 구분별로 패키지를 구성한다 Ex) com.demo1.admin , com.demo1.common
* 업무 구분별 밑에 세부 업무 구분으로 패키지를 구성한다. Ex) com.demo1.admin.usermanagement
* 업무 구분 밑 세부 업무관련 컨트롤러 서비스 DAO MODEL 이 있으면 해당 패키지 밑으로 패키지를 구성한다.