2016 OODP 3주차 실습 보조자료

~ Eclipse customize 방법 ~

Eclipse는 Java 코드 작성을 돕기 위한 다양한 기능들을 가지고 있으며, 이들 대부분은 일반적인 프로그래머들이 곧잘 사용하는 방식으로 설정되어 있습니다. 하지만, 우리들 모두가 동일한 프로그래밍 습관을 가지고 있는 것은 아니지요. 따라서 Eclipse는 사용자의 입맛에 맞게 각 기능들을 어느 정도 조절할 수 있도록 많은 옵션들을 제공하고 있습니다. 이 문서에서는 이러한 Eclipse의 customizable 항목들 중 일부를 소개하며 각각에 대한 조교의 짤막한 의견을 함께 서술합니다.

# 코드 표현 방식 변경

이 단원에서는 Eclipse가 Java 코드를 표현하는 방식을 변경하는 방법을 알아 봅니다.

## 기본 font 변경

* Window 메뉴 🡪 Preferences를 클릭하면 창이 하나 뜹니다.
* 좌측 목록에서 General 🡪 Appearance 🡪 Colors and Fonts을 누른 다음,
* Basic 🡪 Text Font를 클릭하면 현재 설정을 확인할 수 있습니다.
* 여기서 우측 ‘Edit…’ 버튼을 눌러 설정을 변경할 수 있습니다.
  + 조교는 보통 돋움체 10pt를 사용하며 이는 Visual Studio의 기본 설정과 같습니다.  
    (요즘 버전 Visual Studio는 Ctrl + 마우스 휠을 통해 실시간으로 크기를 바꿀 수 있지요)
* 사용할 font를 정한 다음에 ‘스크립트(R)’ 항목을 꼭 ‘한글’로 바꾸어 주세요!

## 코드 강조 규칙 변경

* Window 메뉴 🡪 Preferences 창을 다시 띄웁니다.
* 좌측 목록에서 Java 🡪 Editor 🡪 Syntax Coloring을 누릅니다.
* 아직 배우지 않은 것들 투성이지만, 각 요소들에 대한 색상 및 특징을 조절할 수 있습니다.
  + 조교는 보통 모든 항목의 ‘Bold’ 옵션을 다 빼 놓고 씁니다. 조교는 **int**가 굳이 굵은 글씨로 보여줘야 할 만큼 중요한 요소라고 생각하지 않습니다.
  + 다만, 경우에 따라 return 키워드는 굵게 보여주는 것이 더 나을 지도 모릅니다. 그래서 Eclipse는 Keyword ‘return’ 옵션과 Keyword excluding ‘return 옵션을 따로 제공합니다.

# 코드 정렬 방식 변경

자동 완성 기능을 통해 코드를 추가하거나 Ctrl + Shift + F를 눌러 코드를 정렬할 때 Eclipse는 미리 설정된 정렬 방식에 따라 해당 작업을 수행합니다. 여러분이 Eclipse의 기본 정렬 기준과 동일한 코드 작성 습관을 가지고 있다면 좋겠지만, 그렇지 않은 경우 일일이 엔터 키나 스페이스 바를 쳐 줘야 자신이 원하는 코드를 볼 수 있게 될 것입니다. 이 단원에서는 Eclipse가 여러분과 동일한 코드 작성 습관을 갖도록 정렬 방식을 변경하는 방법을 알아 봅니다.

## 변경 방법

* Window 메뉴 🡪 Preferences를 클릭하면 창이 하나 뜹니다.
* 좌측 목록에서 Java 🡪 Code Style 🡪 Formatter를 누릅니다.
* ‘New…’ 버튼을 눌러 새로운 규칙 집합의 이름을 적고 엔터 키를 강타합니다.
  + 강타하면 꽤 큰 설정 창이 뜨는 것을 볼 수 있습니다. 나중에 이 창을 다시 열려면 이전 창에서 ‘Edit…’ 버튼을 누르면 됩니다.
* 우선 오른쪽의 ‘Show invisible characters‘를 체크해 둡니다. 앞으로 살펴 볼 설정들 중 일부는 ‘빈 공간’을 어떻게 얼마나 남겨 둘 것인지를 조절하는 데 사용됩니다.
* 이제… 아래에 나열된 각 항목 설명을 적당히 읽어 보면서 자신에게 가장 적합한 규칙을 만들어 보도록 합시다.

## 세부 설명

* Indentation: ‘들여쓰기’에 대한 일반적 규칙을 설정합니다.
* Braces: ‘여는 중괄호’를 어디에 둘 지 정합니다.
  + 조교는 대부분의 중괄호를 다음 줄에 쓰지만 ‘배열 초기식’은 같은 줄에 둡니다.
* White Space: 식을 구성하는 각 요소들 간의 ‘공백’에 대한 규칙을 정합니다.
  + 몇몇 옵션들은 건드려도 아직 차이가 안 보일 수 있습니다. ‘줄 바꿈’ 설정이 ‘공백’ 설정에 우선하기 때문입니다(다음 줄로 보낸 중괄호를 한 칸 앞으로 보내면 안 되겠지요).
  + 조교는 if/for/while문 등을 ‘while ( true )’처럼 공백을 앞/뒤에 넣어 씁니다. 수식을 쓸 때도 ( 3 + 5 )처럼 앞/뒤 공백을 넣습니다.
* Blank lines: 코드 정렬 기능이 ‘빈 줄’을 얼마나 확보할 / 남겨 둘 것인지를 지정합니다.
  + Eclipse가 빈 줄들을 그냥 내버려 두기를 원하는 경우, 모든 설정을 0으로 두고 맨 아래 것만 99로 지정해 두세요.
* New Lines: 코드 정렬 기능이 엔터 키를 어디에 적용할 것인지를 지정합니다.
* Control Statements: if/for/while문 등에 엔터 키를 어떻게 적용할 것인지를 지정합니다.
  + 조교는 ‘중괄호는 항상 그 줄에 혼자 있게 만들기’를 추구합니다. 이렇게 하면 코드 전체 줄 수가 꽤 길어지지만 코드의 중첩 관계를 파악하기 쉽고 복사/붙여넣기도 좀 더 편하게 할 수 있게 됩니다.
  + 자신이 if문 구성에 약하다면 Keep ‘else if’ on one line 옵션을 해제해 두는 것이 좋습니다. if-else문 연결이 항상 같은 들여쓰기 수준으로 표시되므로 ‘else를 빼먹은’ 상황을 파악하기에 좀 더 수월합니다.
* Line Wrapping: 코드가 너무 길 때 자동 줄바꿈을 어떻게 수행할 것인지를 지정합니다.
  + 자동 줄바꿈을 원하지 않는 경우 Maximum line width를 9999로 지정해 두세요.
  + 자동 줄’합침’을 원하지 않는 경우 Never join already wrapped lines를 체크해 두세요. 이 옵션을 체크하지 않으면 여러분이 직접 엔터 키를 쳐서 보기 좋게 해 놓아도 Eclipse가 그걸 다시 한 줄로 만들어버릴 수 있습니다.
  + 조교는 항상 수동으로 코드 길이를 조절하기 때문에 이 기능을 사용하지 않습니다.
* Comments: 코드 정렬 기능이 주석 내용을 어떻게 손 댈 것인지를 지정합니다.
  + 여러분이 주석 아티스트라면 이 기능을 싫어하게 될 것입니다. 그렇지 않은 경우 주석 배치 방법을 코드의 것과 유사하게 설정해 두면 좋습니다. 조교는 이 정렬 기능을 Javadoc에 대해서만 씁니다.
* Off/On Tags: 코드 정렬 기능을 끄거나 다시 켜는 명령어를 지정합니다.
  + 중증 완벽주의자가 아닌 이상 아마도 이 옵션을 손 댈 일은 없을 듯 하군요.

## 설정이 다 끝나면…

* 설정 창의 ‘OK’ 버튼을 눌러 기나긴 설정을 끝냈다면 일단 Preferences 창의 ‘Apply’ 또는 ‘OK’ 버튼을 눌러 설정을 꼭 **저장**하세요! 실수로 취소해 버리면 **우리 실습에서 느낄 수 있는 가장 강력한 허탈감**을 맛보게 될 것입니다.
* Eclipse는 ‘Import…’ 버튼과 ‘Export All…’ 버튼을 통해 설정을 .xml 파일에 저장하거나 다시 읽어 오는 기능을 제공합니다. Eclipse를 새로 설치할 때를 대비하여 지금 미리 설정 백업을 해 두는 것도 좋겠군요.
* 이제 코드 창에서 Ctrl + Shift + F를 누르면 화면에 있는 코드가 ‘여러분이 작성한 코드’로 바뀌는 기적을 볼 수 있게 됩니다.

# 자동 완성 기능 조정 (초보자용 for문 추가하기)

Eclipse의 for문 자동 완성 템플릿들은 모두 배열 등 자료구조를 처리하기 위한 것들로 구성되어 있습니다. 하지만 우리는 아직 초보 프로그래머기 때문에 데이터 조작보다는 ‘코드 반복 실행’을 위해 for문을 자주 사용하고 있습니다(나중에 중수 이상이 되면 복잡한 탈출 조건을 구현하기 위해 while문을 더 자주 쓰게 됩니다). 이 단원에서는 ‘초보자용 for문’ 자동 완성 기능을 Eclipse에 추가해 봅니다.

* Window 메뉴 🡪 Preferences를 클릭하면 창이 하나 뜹니다.
* 좌측 목록에서 Java 🡪 Editor 🡪 Templates를 누릅니다.
* 다양한 자동 완성 키워드들을 볼 수 있습니다. 스크롤 바를 쭈욱 내리다 보면 for문도 보이고 우리가 자주 썼던 sysout도 발견할 수 있습니다.
* 이제 ‘New…’ 버튼을 누릅니다.
* Name은 ‘for’로 적습니다. 이 이름은 자동 완성 키워드가 됩니다.
* Context는 ‘Java statements’로 설정합니다. 이제 메서드 안에서 for를 치고 Ctrl + Space 치면 이 자동 완성 템플릿이 보이게 됩니다.
* Description은 자신이 좋아하는 글귀 또는 ‘for statement for beginners’라고 적어 줍시다.
* 이제 Pattern에 아래의 코드를 입력합니다:

|  |
| --- |
| for ( int ${index} = 0; ${index} < ${comparer:var(int, long, short)}; ++${index} ) {  ${cursor} } |

* + 이 코드를 직접 키보드로 치는 것은 효율적이지 않습니다. 조교라면 Ctrl + C, Ctrl + V를 통해 그대로 옮기는 쪽을 선택할 것입니다.
  + ${index}와 같은 표현이 어떤 의미인지는 일단 지금은 크게 중요하지 않습니다. 그래도 만약 궁금하다면 입력 칸에서 Ctrl + Space를 눌러 사용 가능한 표현 목록 및 설명을 확인할 수 있습니다.
* 이제 OK OK를 누르면 새로운 자동 완성 템플릿을 사용할 수 있게 됩니다!