Journal of the Korea Academia-Industrial cooperation Society
Vol. 23, No. 10 pp. 690-696, 2022

https://doi.org/10.5762/KAIS.2022.23.10.690
ISSN 1975-4701 / eISSN 2288-4688

무기체계 소프트웨어의 Java 코딩 규칙 검증 기준 개선 연구

윤재형, 류지선*
국방기술품질원

Improving the Coding Rules of Java for Weapon Systems Software

Jae-Hyeong Yun, Jiseon Yu*
Defense Agency for Technology and Quality

요 약 1991년 SUN Microsystems에서는 메모리 사용량이 적고 다양한 플랫폼을 갖는 가전 제품에 적용하기 위해 Java언어를 개발하였다. 이후 스마트폰 플랫폼인 안드로이드에서 Java API를 활용하며 누구나 쉽게 접할 수 있는 대중적인 언어로 발돋움 하였다. 이에 무기체계 소프트웨어 분야에서는 방위력개선사업으로 획득되는 소프트웨어의 체계적인 개발 및 관리를 위한 프로세스와 산출물 작성표준 등을 규정하는 「무기체계 소프트웨어 개발 및 관리 매뉴얼」에 Java 언어로 개발된 무기체계 소프트웨어의 코딩 규칙 검증과 취약점 점검 등을 수행할 수 있는 신뢰성 시험 기준을 추가하여 무기체계 소프트웨어에 최신 소프트웨어의 흐름을 반영하였다. 일반적으로 코딩 규칙은 코드를 확인하는 다른 사람들이 내용에 집중할 수 있도록 일관성 있는 코드를 제공하여 빠른 이해를 돕고 코드의 복사, 변경, 유지 관리할 수 있도록 하는 것을 목적으로 한다. 본 논문에서는 무기체계 소프트웨어 개발 및 매뉴얼의 Java언어 코딩 규칙 검증 기준인 오라클의 코딩 규칙과 최신의 구글의 코딩 규칙을 비교하고 행정안전부 소프트웨어 개발보안 가이드 적용에 대한 적절성을 분석하여 Java언어로 개발된 무기체계 소프트웨어 코딩 규칙의 발전 방향을 제안한다. 그 결과 적용 언어별 점검 난이도가 상이하고 일부 코딩 규칙이 취약점 분석과 유사한 사항이 있음을 식별하고, 개발 언어별 코딩 규칙 시험 난이도가 동일한 수준으로 개선이 필요함을 보인다.

Abstract In the field of weapon systems software, the coding rules for weapon systems software developed in the Java language are included in the "Weapon Systems Software Development and Management Manual", which stipulates the process and output standards for the systematic development and management of software acquired through the defense capability improvement project. The latest trends of software development are reflected in weapon system software by adding a reliability test standard. In general, the purpose of coding rules is to provide consistent code so that people looking at the code can focus on the content, making it easier to understand, copy, change, and maintain the code. In this paper, we compare Oracle's Java language coding rule verification standards for Weapon Systems Software Development and Management Manual with the lasted coding rules from Google. We also analyzed whether it is appropriate to apply MOIS's software development security guide to the manual. As a result, we suggest an improvement direction for the coding rules of weapon system software developed in Java language.

Keywords : Weapon System, Software, Java, Coding Rules, Coding Conventions

*Corresponding Author : Jiseon Yu(DTaQ)
email: gsun2@dtaq.re.kr
Received August 19, 2022 Revised September 28, 2022
Accepted October 7, 2022 Published October 31, 2022

1. 서론

Java 언어는 1991년 SUN microsystems에서 가전 제품에 들어갈 소프트웨어를 개발하기 위한 '그린 프로젝트(Green Project)'가 시작되면서 제임스 고슬링(James Gosling)에 의해 만들어 졌다. 당시 개발 목적은 C나 C++ 등 기존 언어로 작성된 프로그램은 Unix 등 플랫폼 간 호환성 문제를 해결할 수 있도록 모든 플랫폼에 호환성을 갖는 웹에 최적화된 프로그래밍 언어의 필요성에 의해 메모리 사용량이 적고 다양한 플랫폼을 갖는 가전 제품에 적용하기 위함이었다. 이에 1995년 OAK라는 이름으로 개발되어 현재 Java에 이르렀다. Java의 큰 다섯 가지 특징은 다음과 같다. ① 단순하고 객체지향적이며 친숙해야 한다. ② 견고하며 보안상 안전해야 한다. ③ 아키텍처에 중립적이고 이식성(Portability)을 가져야 한다. ④ 높은 성능을 제공해야 한다. ⑤ 인터프리터 언어이고 쓰레드(thread)를 제공하며 동적(dynamic)인 언어이다. 이러한 특징으로 C, C++이 개발환경의 운영체제에 맞는 기계어로 컴파일 하는 것과 달리, Java는 가상 머신(JVM) 바이트 코드로 컴파일하여 C, C++보다 실행속도가 느리지만 보안성과 이식성을 갖는다. 또한, 메모리 관리 등 C, C++에서 개발자가 직접 제어해야 했던 각종 버그를 일으키기 쉬운 기능들을 없애 개발자에 친숙하고 빠른 개발이 가능하여 현재까지 인기 있는 언어로 자리매김 하였다. 특히 스마트폰 플랫폼인 안드로이드(Android)에서 리눅스 커널 기반의 Java API (Application Program Interface)를 활용하면서 누구나 쉽게 접할 수 있는 대중적인 언어로 발돋움 하였다.

이에 따른 무기체계 소프트웨어 개발과 관련한 움직임으로는 2016년, 방위력개선사업으로 획득되는 소프트웨어의 체계적인 개발 및 관리를 위한 프로세스와 산출물 작성표준 등을 규정하는 '무기체계 소프트웨어 개발 및 관리 매뉴얼(방위사업청)'에 Java로 개발된 무기체계 소프트웨어의 신뢰성 시험 기준을 추가해 개정하는 등 무기체계 소프트웨어에 최신 소프트웨어의 흐름을 반영할 수 있도록 하였다. 무기체계 소프트웨어에서 신뢰성 시험이란 소프트웨어가 동작할 수 있는 다양한 경우의 수를 확인함으로써 소프트웨어가 일으킬 수 있는 결함을 식별하는 시험을 말한다. 소프트웨어 신뢰성 시험에서 정적 시험은 소프트웨어를 실행하지 않은 상태에서 잠재적인 결함을 검출하는 시험인데 코딩 규칙 검증 시험, 취약점 점검 시험 및 소스코드 메트릭 점검을 수행하는 활동을 말한다. 이 중 본 논문에서 다루는 코딩 규칙은 소프트웨어 구현에 적용하는 소스 코드 작성 규칙을 말한다. 코딩 규칙 검증 활동으로는 체계에 적용된 언어에 따라 매뉴얼에서 정의한 관련 표준을 준수하였는지 확인한다. 코딩 규칙은 코드를 확인하는 다른 사람들이 내용에 집중할 수 있도록 일관성 있는 코드를 제공하여 빠른 이해를 돕고 코드의 복사, 변경, 유지 관리할 수 있도록 하는 것에 목적이 있다[1]. 즉, 소프트웨어 개발에서 코딩 규칙을 준수하고 검증하는 가장 큰 이유는 유지보수성(Maintainability)을 확보하기 위함이다. 유지보수성은 소프트웨어 시스템의 기능을 변경하거나 기능 추가 및 성능 향상을 위한 결함을 수정할 때의 편의성을 말하는데 관련 연구조직이나 개발사에서는 유지보수성 확보를 위해 다양한 언어에 대한 코딩 규칙을 만들어 배포하고 있다(Table 1). 이 중 우리 무기체계 소프트웨어는 위 매뉴얼에 따라 C, C++, C#, Java 언어에 대해 Table 1에서 굵게 표시한 코딩 규칙을 기준으로 검증하고 있다. 여기서 C#, Java가 준수해야하는 코딩규칙은 코드의 가독성과 일관성을 확보하여 유지보수성을 높이는 것이 목적인 반면에 C, C++이 준수해야하는 MISRA 규칙은 코드의 안전성 향상을 위한 규칙들로써 시큐어 코딩 가이드의 성격이 더욱 강하여 적용 언어 간 점검 기준의 강도가 다른 문제점이 있다. 또한 Java는 1997년 Java언어의 탄생과 함께 배포된 코딩 규칙을 기준으로 하고 있는 문제점은 앞으로 반드시 해결해 나가야 할 부분이다. 현재까지 개발된 무기체계 소프트웨어는 대부분 하드웨어를 조작하기 위한 내장형 소프트웨어가 대부분이었다. 그러나 전쟁

Table 1. The list of Coding Rules

Language	Coding Rules
C	**· MISRA-C** · NASA C style guide · Quantum C Coding Standard · Micrium C Coding Standard · Google C Style Guides
C++	**· MISRA-C++** **· JSF++(Joint Strike Fighter Air Vehicle C++)** · Quantum C++ Coding Standard · Google C++ Style Guides
C#	**· C# Coding conventions** **· .NET Framework Design Guideline**
Java	**· Code conventions for the Java Programming Language** · Google Java Style Guides
Python	· PEP8(Style Guide for Python Code) · Google Python Style Guides
JavaScript	· JavaScript Standard Style · Airbnb JavaScript Style Guides · Google JavaScript Style Guides

수행 개념에 대한 패러다임이 네트워크중심전(NCW, Network Centric Warfare)으로 변화하고 사이버전(Cyber Warfare)의 중요성이 대두되는 요즘, 고전 무기체계에 적용하기 위해 하드웨어를 제어하기 위한 소프트웨어뿐만 아니라 첨단 무기체계에 도입될 다양한 신기술이 폭넓게 적용될 수 있도록 이에 대한 대비가 필요하다.

관련 연구에서 개발 언어의 코딩 규칙은 Coding Conventions, Style Guides, Coding Standards 등의 영문으로 표기하고 있으나, 본 논문에서 국문 표기는 무기체계 소프트웨어 개발 및 관리 매뉴얼에서 정의한 '코딩 규칙'으로 한다. 코딩 규칙이란 코드를 일관되게 작성하여 가독성을 높이고 코드 복사, 변경 및 유지 관리를 용이하게 하는 것을 말한다. 본 논문에서 참조하는 오라클의 Java 코딩 규칙에서도 소프트웨어 개발에서 들어가는 비용 중 80%가 유지보수에 쓰이고 소프트웨어를 개발한 개발자가 유지보수를 담당하는 경우는 거의 보기 어려우며, 코딩 규칙을 준수하면 가독성이 높아지고 다른 소스코드들과 잘 어울릴 수 있음을 이유로 코딩 규칙의 중요성을 거론하고 있다[2].

따라서 본 논문에서는 무기체계 소프트웨어 개발 및 관리 매뉴얼의 Java 언어 코딩 규칙 검증 기준인 오라클(Oracle)의 코딩 규칙과 최신의 구글(Google)의 코딩 규칙을 비교하고, 무기체계 소프트웨어 코딩 규칙 적용을 위한 적절성을 분석하며 전체적으로 무기체계 소프트웨어 코딩 규칙의 발전 방향을 제안한다.

2. 본론

2.1 유지보수성 향상을 위한 무기체계 소프트웨어 코딩규칙 필요성

민간에서 기능이나 성능에 대한 공통된 기준과 목표 또는 프로세스를 정하여 KS 등의 표준을 제정하는 것과 같이 국방에서는 군수품 조달에 필요한 제품 및 용역에 대한 성능, 재료, 형상, 치수 등 기술적인 요구사항과 요구필요조건의 일치성 여부를 판단하기 위한 절차와 방법을 서술하여 관리하는 국방규격이 있다[3]. 이처럼 소프트웨어 개발에서는 일관된 기준을 따르도록 정하고 관리하기 위해 코딩 규칙을 적용하여 개발한다. 무기체계 소프트웨어는 「무기체계 소프트웨어 개발 및 관리 매뉴얼」의 부록 6 '무기체계 소프트웨어 코딩규칙(DAPA SCR-G)'에서 정의하는 기준을 적용된 언어에 따라 점검한다[4]. 그러나 이 매뉴얼의 코딩규칙은 코드의 안전성 향상을 위해 만들어진 MISRA 규칙과 유사하여, 앞서 정의한 일반적인 코딩 규칙이 아닌 시큐어 코딩 준수 사항과 혼합된 상태이다. 그래서 인지 무기체계 신뢰성 시험의 코딩 규칙 개선 방향을 연구한 대부분의 연구에서는 코딩 규칙의 의미를 시큐어 코딩 혹은 취약점 점검 관점으로 연구를 수행하였다. 현재의 신뢰성 시험 기준과 OWASP, CWE, 행정안전부 보안약점 기준의 IoT 주요 보안약점을 조사 분석하여 IoT 주요 보안약점을 제거하기 위한 무기체계 소프트웨어 코딩규칙 개정의 필요성을 보였다[5]. 최근 코딩규칙 기준 확대 방안으로 C++언어에 AUTOSAR C++14와 C++ Core Guidelines 확대 적용 필요성을 보이는 연구가 수행되었지만[6], Java 언어로 개발된 무기체계 소프트웨어의 코딩규칙과 관련된 연구는 전무한 상황이다.

무기체계의 경우 양산업체 선정 시 개발업체와 달라질 수 있기 때문에 유지보수성 확보를 위해 공동의 규칙을 준수하는 것이 매우 중요하며 특히 소프트웨어의 경우에는 하드웨어와 달리 시간의 흐름에 따라 닳거나 사라지지 않기 때문에 개발 과정에서 유지보수성을 확보하여 운용유지 단계에서 기능의 변경이 용이하도록 해야 한다. 즉, 코드의 형상을 변경하기 위해 코드를 다른 사람에게 제공하거나 다른 사람이 작성한 코드를 해석해야 하는 상황에서 코딩 규칙을 준수한 코드는 가독성과 유지보수성을 제공하므로 대규모 개발 환경인 무기체계 소프트웨어에서 유지보수성 향상을 위해 코딩 규칙을 준수하는 것은 필수이다.

2.2 코딩규칙 분석

본 장에서는 오라클과 구글의 Java 코딩 규칙 차이점을 살펴보고, 구글 코딩 규칙의 무기체계 소프트웨어 코딩 규칙 및 행정안전부 소프트웨어 개발보안 가이드 적용에 대한 적절성 분석 및 분석 결과 기반 무기체계 소프트웨어 개발 및 관리 매뉴얼의 코딩규칙 기준 개선 방향을 제안한다.

2.2.1 Code Conventions for the JAVA Programming Language(Oracle)

이 코딩 규칙은 Java가 오라클에 인수되기 전인 1997년에 Java 언어를 만든 Sun Microsystems에서 발표한 코딩 규칙이다. Table 2의 왼쪽에 작성된 것과 같이 파일 이름, 파일 구조, 들여쓰기, 주석, 선언, 문(statements), 공백 등 11개 카테고리로 구분되어 약 81개의 규칙이 있다[2]. 1997년에 발표된 이후 개정 이력은 없다. Java를

개발하는데 가장 많이 사용된 IBM에서 개발한 통합 개발 환경(이하 IDE, Integrated Development Environment) Eclipse는 Java언어에 대한 이 코딩 규칙이 기본적으로 적용되어 있다. Java언어를 개발한 Sun Microsystems에서 개발한 만큼 가장 기본적인 규칙들로 구성되어 있으며 이 코딩 규칙을 구글을 비롯한 다수의 기업에서 개발 환경에 적합한 코딩 규칙으로 발전시켜 널리 사용하고 있다.

2.2.2 Google Java Style Guides(Google)

구글 코딩 규칙은 오라클의 코딩 규칙 기반으로 작성되어 대부분 동일한 구조를 가진다. Table 2의 오른쪽에 작성된 것과 같이 소스 파일 기본 사항, 소스 파일 구조, 코드 형식(formatting), 식별자 선언(naming), 실습으로 목차가 구성되어 있으며 7개의 카테고리로 구분되어 27개의 규칙이 있다. 이 코딩 규칙에서는 개발자 사이에서 보편적으로 준수하고 있는 엄격한 규칙에 초점을 맞추고 명확하게 시행할 수 없는 항목은 제시하지 않는다는 점을 명시하였다. 구글에서는 Java 언어 외에도 C, C++, Python, R, JavaScript 등 다양한 언어에 대한 코딩 규칙을 최신의 스타일이 반영될 수 있도록 꾸준히 업데이트하고, github를 통해 제공하고 있다[7]. 개발자는 사용하는 IDE에 공개된 코딩 규칙을 적용하여 편하게 개발할 수 있다.

최근 안드로이드 운영체제의 스마트폰 앱 개발 IDE인 안드로이드 스튜디오가 Eclipse에서 IteliJ 기반으로 바뀌고, 구글 코딩 규칙이 최신의 세련된 스타일의 코딩 규칙이라는 보편적 인식이 생기면서 각종 언어 개발 시 개인의 IDE에 구글 코딩 규칙을 적용하여 개발하는 추세이다.

2.2.3 행정안전부 소프트웨어 개발보안 가이드

이 가이드는 행정기관 및 공공기관 정보시스템 구축 및 운영 지침에 따라 정보시스템을 구축하고 운영함에서 있어서 보안약점이 나타나지 않도록 안전한 소프트웨어 개발을 위한 가이드이다. 각 사업단계별 그리고 주체별 개발보안 활동을 소개하고 개발 프로세스의 분석/설계/구현단계에서 소프트웨어 안전성 향상 활동을 소개하며 Java, C, C#언어로 작성된 안전한 코드와 안전하지 않은 코딩 예제를 소개한다[8].

Table 2. The list of Contents

Oracle	Google
1. Introduction 1.1 Why Have Code Conventions 1.2 Acknowledgments 2. File Names 2.1 File Suffixes 2.2 Common File Names 3. File Organization 3.1 Java Source Files 3.1.1 Beginning Comments 3.1.2 Package and Import Statements 3.1.3 Class and Interface Declarations 4. Indentation 4.1 Line Length 4.2 Wrapping Lines 5. Comments 5.1 Implementation Comment Formats 5.1.1 Block Comments 5.1.2 Single-Line Comments 5.1.3 Trailing Comments 5.1.4 End-Of-Line Comments 5.2 Documentation Comments 6. Declarations 6.1 Number Per Line 6.2 Placement 6.3 Initialization 6.4 Class and Interface Declarations 7. Statements 7.1 Simple Statements 7.2 Compound Statements 7.3 return Statements 7.4 if, if-else, if-else-if-else Statements 7.5 for Statements 7.6 while Statements 7.7 do-while Statements 7.8 switch Statements 7.9 try-catch Statements 8. White Space 8.1 Blank Lines 8.2 Black Spaces 9. Naming Conventions 10. Programming Practices 10.1 Providing Access to Instance and Class Variables 10.2 Referring to Class Variables and Methods 10.3 Constants 10.4 Variable Assignments 10.5 Miscellaneous Practices 10.5.1 Parentheses 10.5.2 Returning Values 10.5.3 Expressions before '?' in the Conditional Operator 10.5.4 Special Comments 11. Code Examples 11.1 Java Source File Example	1. Introduction 1.1 Terminology notes 1.2 Guide notes 2. Source file basics 2.1 File name 2.2 File encoding: UTF-8 2.3 Special characters 3. Source file structure 3.1 License or copyright information, if present 3.2 Package statement 3.3 Import statements 3.4 Class declaration 4. Formatting 4.1 Braces 4.2 Block indentation: +2 spaces 4.3 One statement per line 4.4 Column limit: 100 4.5 Line-wrapping 4.6 Whitespace 4.7 Grouping parentheses: recommended 4.8 Specific constructs 5. Naming 5.1 Rules common to all identifiers 5.2 Rules by identifier type 5.3 Camel case: defined 6. Programming Practices 6.1 @Override: always used 6.2 Caught exceptions: not ignored 6.3 Static members: qualified using class 6.4 Finalizers: not used 7. Javadoc 7.1 Fomatting 7.2 The summary fragment 7.3 Where Javadoc is used

2.3 제안하는 무기체계 소프트웨어의 Java언어 코딩규칙 기준 개선 방향

2.3.1 Google의 무기체계 Java 코딩 규칙 적용 적절성 분석

기본적으로 구글의 코딩 규칙은 오라클의 코딩 규칙을 바탕으로 만들어져 있음을 알아야 한다. 그렇기 때문에 개발자는 구글 코딩 규칙이 오라클의 코딩 규칙 보다 더 상세하고 엄격하다고 느낄 수 있다. 구글 코딩 규칙이 더욱 엄격하게 제안하는 규칙을 정리하면 아래와 같다.

Table 3. The difference between Oracle and Google coding rules

	Oracle	Google
Comment	/* .. */	// /* .. */
Line Length	80	80~100
Indentation	Not tabs 8 spaces	4 spaces(tabs)
White Space	One or Two blank line	Not required multiple line

· 소스파일 인코딩 : UTF-8
· 특수 문자(공백, escape sequences, Non-ASCII)
· 라이센스 또는 저작권 정보
· Overloads
· Vertical, Horizontal 공백
· 예외 처리 구문에서 주석 작성

오라클의 코딩 규칙이 배포된 1995년과 달리 현재는 Java가 많은 플랫폼에서 사용되어 각종 오픈소스나 유료 API 사용에 따른 라이센스 또는 저작권 정보를 명시하는 규칙과 다양한 플랫폼에서 활용될 수 있도록 인코딩(encoding)관련 규칙이 추가된 것으로 보여 진다. 이 밖에 들여쓰기는 개발 편의성을 높여주는 다양한 IDE의 등장과 관습처럼 굳어진 개발자들의 습관이 코딩규칙으로 반영되어 tab 사용을 권고하지 않았던 오라클과 달리 구글에서는 들여쓰기에 tab을 사용하도록 하였다. 또한 하드웨어 성능 발전에 따라 메모리 사양이 향상되어 한 줄에 작성할 수 있는 코드의 길이 또한 구글 코딩규칙에서 더 많이 허용하고 있음을 확인할 수 있다(Table 3). 따라서 구글 코딩 규칙을 적용하면 개발자는 IDE를 통해서 개발 과정에서 자연스럽게 코딩 규칙을 적용하여 개발함으로써 최신 개발 트렌드에 맞는 코딩 규칙이 적용된 유지보수성을 확보한 무기체계 소프트웨어를 획득할 수 있을 것이다.

2.3.2 행정안전부 소프트웨어 개발보안 가이드의 무기체계 Java 코딩 규칙 적용 적절성 분석

현재 무기체계 소프트웨어 개발 및 관리 매뉴얼에서는 C, C++로 개발된 소프트웨어의 신뢰성 시험의 코딩규칙 기준으로 자동차 소프트웨어의 안전성 향상을 위해 ISO/IEC 26262를 기반으로 작성된 MISRA 코딩규칙을 적용했는지 확인한다. 다만 Java 언어로 개발된 무기체계 소프트웨어는 안전성이 아닌 유지보수성 향상을 위한 일반적인 코딩규칙인 오라클 코딩규칙을 기준으로 확인하기 때문에 다른 언어와 유사한 수준으로 시험을 수행하고 코딩규칙 점검을 통해 안전성을 확보할 수 있는 방안이 필요한 상황이다.

Table 4. Contents of similar coding rules

Rules	Contents
MISRA C	- Conversions shall not be performed between a pointer to a function and any other type - Conversions shall not be performed between a pointer to an incomplete type and any other type - A conversion should not be performed between a pointer to object and an integer type - A conversion should not be performed from pointer to void into pointer to object
Software Development Security Guides	· Verification of Security function input value - Design of validation method for security function(authentication, authorization, etc.) input value, external input value of function(or method) and execution result, and design of handling method for invalid value

따라서 본 절에서는 소프트웨어 개발 과정의 안전성 향상 활동을 소개하는 행정안전부 소프트웨어 개발보안 가이드를 Java언어로 개발된 무기체계 소프트웨어에 적용하고 이를 기준으로 코딩규칙을 점검을 수행할 수 있는지 적절성을 분석하였다. MISRA 코딩 규칙에서는 포인터가 부적절한 값으로 변환되어 시스템에 오버플로우 같은 영향을 미치지 않도록 포인터를 부적절한 값으로 변환하지 말라는 규칙이 있고, 이와 유사하게 행정안전부 소프트웨어 개발보안 가이드에도 널(NULL) 포인터 역참조나 정수형 오버플로우가 발생하지 않도록 보안기능 결정에 부적절한 값을 입력하지 않도록 하는 규칙이 있다(Table 4). 따라서 행정안전부 소프트웨어 개발보안 가이드를 Java언어에 적용하면 C, C++과 유사한 수준으로 안전성을 향상시킬 수 있을 것으로 판단된다.

2.4 제안하는 무기체계 소프트웨어 코딩규칙 기준 개선 방향

본 논문에서는 무기체계 소프트웨어의 유지보수성 향상을 위한 코딩 규칙의 필요성과 최신 코딩 규칙 적용에 대한 적절성을 분석하고 개발 언어별 적용 기준의 차이가 존재함을 확인하였다. 현재 무기체계는 첨단기술의 발전을 빠르게 도입함에 따라 소프트웨어의 중요성과 개발 단계에서부터 품질을 확보하는 것이 무엇보다 중요해지고 있는 상황이다. 정해진 규격에 의해 하드웨어 부품을 만드는 것과 같이 소프트웨어 역시 정해진 규칙에 의해 공동의 기준으로 개발하여 개발 초기 단계부터 유지보수성 등의 품질을 확보해 나가는 것이다. 따라서 본 논문에서 확인한 무기체계 소프트웨어에서 코딩 규칙의 필요성과 최신 개발 동향에 따라 최신 코딩 규칙을 각 언어의 점검 기준으로 수정 또는 추가가 필요하며, 개발 언어에 차이 없이 일관된 기준으로 코딩 규칙을 검증하도록 언어별 코딩규칙 기준 개선이 필요하다.

무기체계 소프트웨어 개발 및 관리 매뉴얼에서는 부품 국산화 등 표준 적용이 제한되는 사업의 경우 본 매뉴얼 부록 6 제3장에 정의된 프로그램 작성 규칙을 따를 수 있는데, 여기에 정의된 코딩 규칙 역시 코드의 가독성을 높이고 유지보수성 향상을 위한 일반적인 코딩 규칙이 아닌 시큐어 코딩 규칙과 혼재된 형태의 규칙을 제안하고 이를 준수하도록 하고 있기 때문에 C, C++언어와 Java언어를 사용했을 때 준수해야 하는 코딩 규칙의 의도와 방향이 다르다고 볼 수 있다. 따라서 Java언어에는 C, C++과 같이 일반적인 코딩 규칙을 준수하면서 안전성을 확보해 나갈 수 있도록 C, C++ 코딩규칙과 유사한 수준의 행정안전부 소프트웨어 개발보안 가이드를 Java에 적용이 필요하다.

위와 같이 제안하는 개선 방향이 실무에 적용되면 다양한 플랫폼에서 활용될 최신 코딩 규칙이 적용된 Java 언어로 개발된 무기체계를 확보하여 유지보수성을 확보하여 코딩 규칙 본래의 의도와 맞는 점검 기준으로 활용할 수 있으며, 다른 언어와 코딩 규칙 점검 기준의 수준을 좁혀 어떤 언어를 사용하더라도 안전성을 확보할 수 있기 때문에 기존 제도 내에서 소프트웨어 품질을 향상시킬 수 있을 것으로 판단된다.

3. 결론

본 논문에서는 무기체계 소프트웨어에서 코딩규칙의 필요성을 식별하고, 최신 코딩 규칙 반영과 언어별 코딩 규칙 수준을 맞추기 위해 Java언어로 개발된 무기체계 소프트웨어에 구글의 코딩 규칙을 적용하고 행정안전부 소프트웨어 개발보안 가이드를 적용하는 발전 방향을 제안하였다. 이를 적용함으로써 .Java언어로 개발된 무기체계 소프트웨어에 최신의 코딩 규칙을 적용하고 C, C++언어 수준의 안전성을 확보할 수 있다.

무기체계는 양산업체 선정 시 개발업체가 아닌 업체가 선정될 수 있기 때문에 일반적으로 개발과 유지보수까지 한 개발업체에서 수행하는 다른 소프트웨어 보다 유지보수성을 확보하는 것이 매우 중요하다. 그렇기 때문에 무기체계 소프트웨어 개발 프로세스에서 공동의 규칙을 정하고 이를 준수하기 위한 모두의 노력이 필요하다. 다만, 무기체계의 특성 상 타 업체의 사례를 공유하거나 공개적으로 활발한 정보 공유가 이루어 질 수 없는 한계가 있기 때문에 유관 업무를 수행하는 업체와 기관이 비교적 자유롭게 정보를 교류할 수 있는 다양한 기회를 만들어 무기체계 소프트웨어 품질 향상이라는 공동의 목표를 수행할 수 있는 방안에 대한 추가 연구가 필요하다. 또한, 하루가 다르게 발전하고 있는 IT 기술을 무기체계에 빠르게 적용하고 새로운 플랫폼과 진화된 언어 반영에 뒤처지지 않도록 끊임없는 무기체계 소프트웨어 연구를 통해 시대의 흐름에 맞춘 제도 개선 등의 노력이 필요하다.

References

[1] Microsoft, “C# Coding Conventions”, 2022.

[2] SUN Microsystems, “Java Coding Conventions”, 1997.

[3] Defense Acquisition Program Administration(DAPA) Rules, ”Rules for Task of Standardization”, 2021.

[4] Defense Acquisition Program Administration(DAPA) Manual “Weapon System Software Development and Management Manual”, 2020.

[5] Youngjun Lee, Joonseon Ahn, Jin-Young Choi “Research on Improving Security of Software Coding Rule Guide, SCR-G”, Conference for Software, 462-464, 2017.

[6] Minkwan Choi, Daun Bak, Seunghak Kook, Taeho Lee, “An Expansion Method of Coding Rules Standard in Weapon System Software Static Test”, Korea Computer Congress, 46-47, 2022.

[7] Google, “Java Style Guide”, 2021.

[8] Ministry of the Interior and Safety(MOIS), “Guide for Software Development Security”, 2021.

윤 재 형(Jae-Hyeong Yun) [정회원]

- 2017년 2월 : 건국대학교 전자공학부 (전자공학학사)
- 2017년 2월 ~ 현재 : 국방기술품질원 연구원

〈관심분야〉
국방, 무기체계 소프트웨어, 소프트웨어 품질

류 지 선(Jiseon Yu) [정회원]

- 2018년 8월 : 고려대학교 정보보호대학원 정보보호학과 (정보보호학석사)
- 2018년 12월 ~ 현재 : 국방기술품질원 연구원

〈관심분야〉
무기체계 보안, 정보보호