УДК 004

Настройка плагина CheckStyle в IntelliJ IDEA

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
Целью данной статьи является настройка плагина CheckStyle в среде разработки IntelliJ IDEA, а также проверка с помощью этого плагина кода, написанного на языке Java.
Ключевые слова: CheckStyle, IntelliJ IDEA, Java, плагин

Configuring CheckStyle Plugin in IntelliJ IDEA

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to configure the CheckStyle plugin in the IntelliJ IDEA development environment, as well as check the code written in Java using this plugin.
Keywords: CheckStyle, IntelliJ IDEA, Java, plugin

1 Введение

1.1 Актуальность

Стандарт оформления кода — набор правил и соглашений, используемых при написании исходного кода на некотором языке программирования. Наличие общего стиля программирования очень важно для понимания и поддержания исходного кода, написанного более чем одним программистом. Также соблюдение общего стиля упрощает взаимодействие нескольких человек при разработке программного обеспечения. Плагин CheckStyle – это инструмент статического анализа кода, используемый при разработке программного обеспечения для проверки соответствия исходного кода Java, указанным правилам кодирования.

1.2 Обзор исследований

В своей работе К.А. Амосов, А.С. Волков изучили назначения и содержания стандартов написания кода, изучения различных подходов к их формированию, рассмотрения образцов действующих стандартов, а также задачу формирования [1]. П.Н. Советов представил обзор современных стандартов стиля кода, а также описание и особенности разработанной библиотеки [2]. С.С. Кизим рассмотрел вопрос внедрения стандартов оформления программного кода, разрабатываемого в рамках дисциплин,

связанных с изучением языков программирования [3]. В своей работе О.Г. Орлинская, А.А. Куликов применили рекомендации по правильному написанию кода при разработке программного обеспечения на языке Java [4]. А.Ю. Заковоротный, М.А. Зорян рассмотрели проблемы оценки качества исходного кода Java-приложений и библиотек [5].

1.3 Цель исследования

Цель исследования - настроить плагин CheckStyle в среде разработки IntelliJ IDEA, и проверить с помощью этого плагина код, написанный на языке Java.

2 Материалы и методы

Для проверки стандарта оформления кода используется плагин CheckStyle и стандарт кода от Oracle, подключаемый в среде разработки IntelliJ IDEA.

3 Результаты и обсуждения

Перед началом работы требуется установить плагин. Для этого в среде IntelliJ IDEA переходим в «Settings». Там выбираем раздел «Plugins» и в появившемся окне, в поисковую строчку вводим «CheckStyle-IDEA» (рис.1).

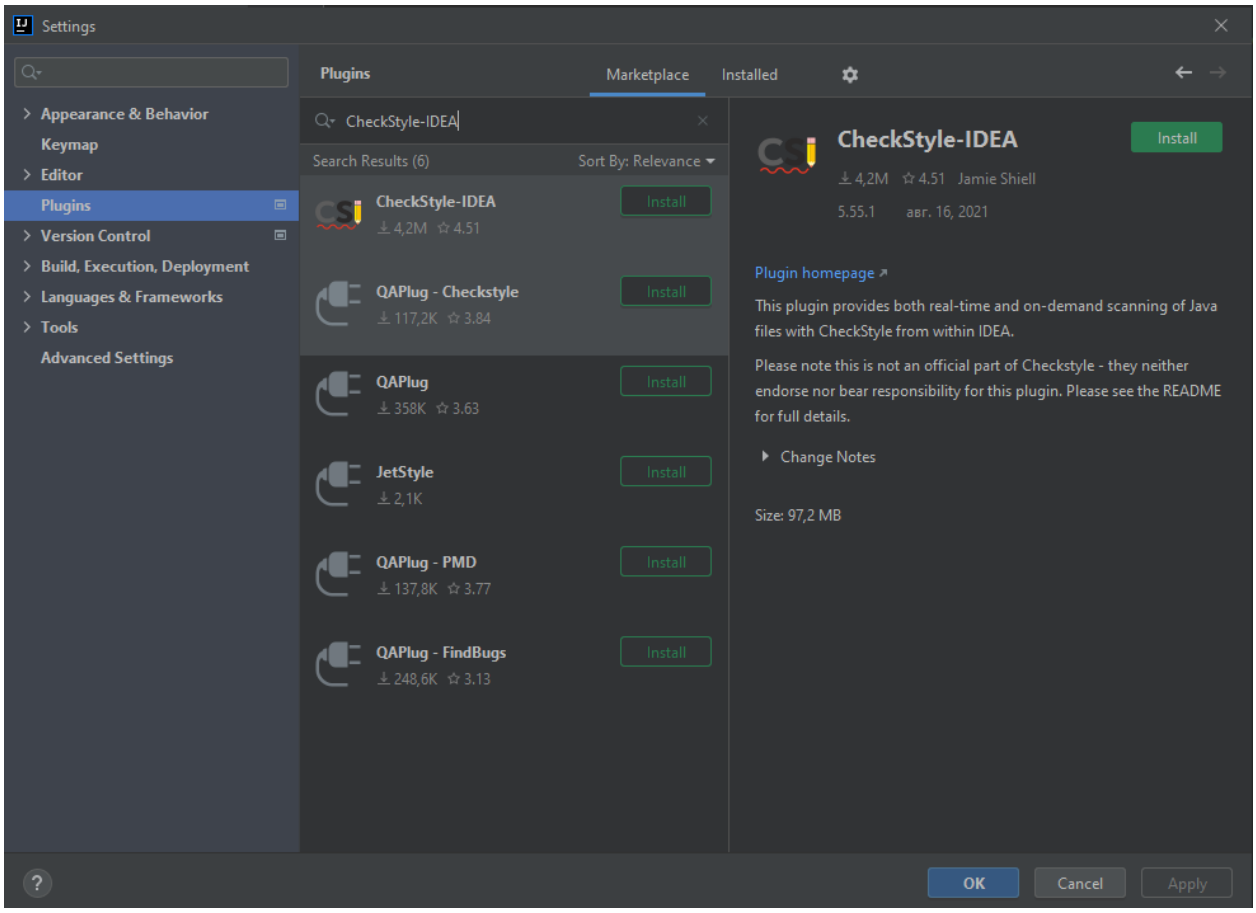

Рисунок 1 – Установка плагина

После установки принимаем изменения и перезапускаем IntelliJ IDEA.

Далее для подключения конфигурации необходимо перейти в настройки IDEA и выбрать раздел CheckStyle. Можно использовать уже встроенные файлы со стандартами кода от Google или Oracle. В данной работе был использован стандарт кода от Oracle. Также возможно использовать свой файл (рис. 2). Для этого необходимо добавить новый пункт в таблицу «Configuration File» и указать источник конфигурации. Возможно добавление по URL (рис. 3).

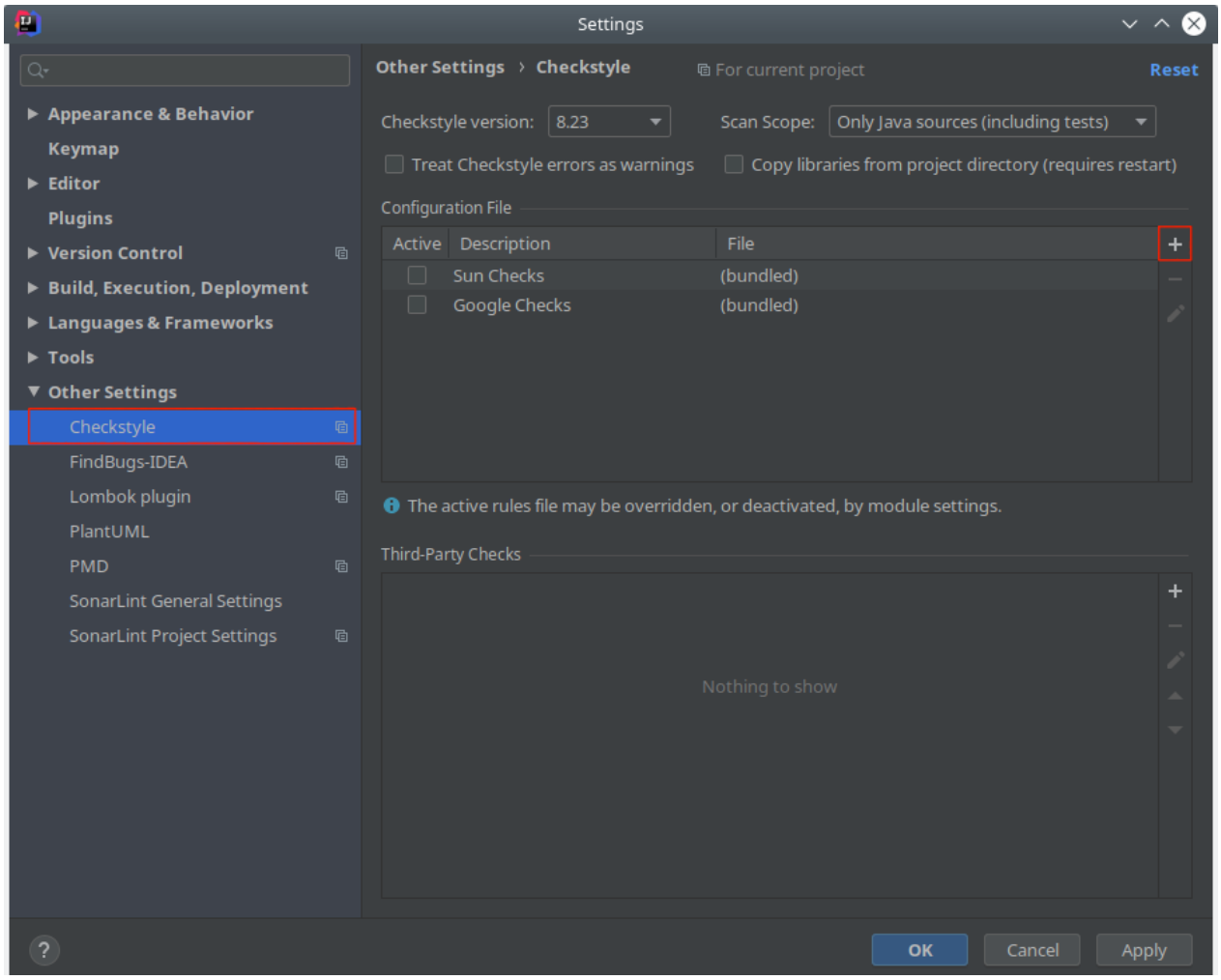

Рисунок 2 – Выбор файла

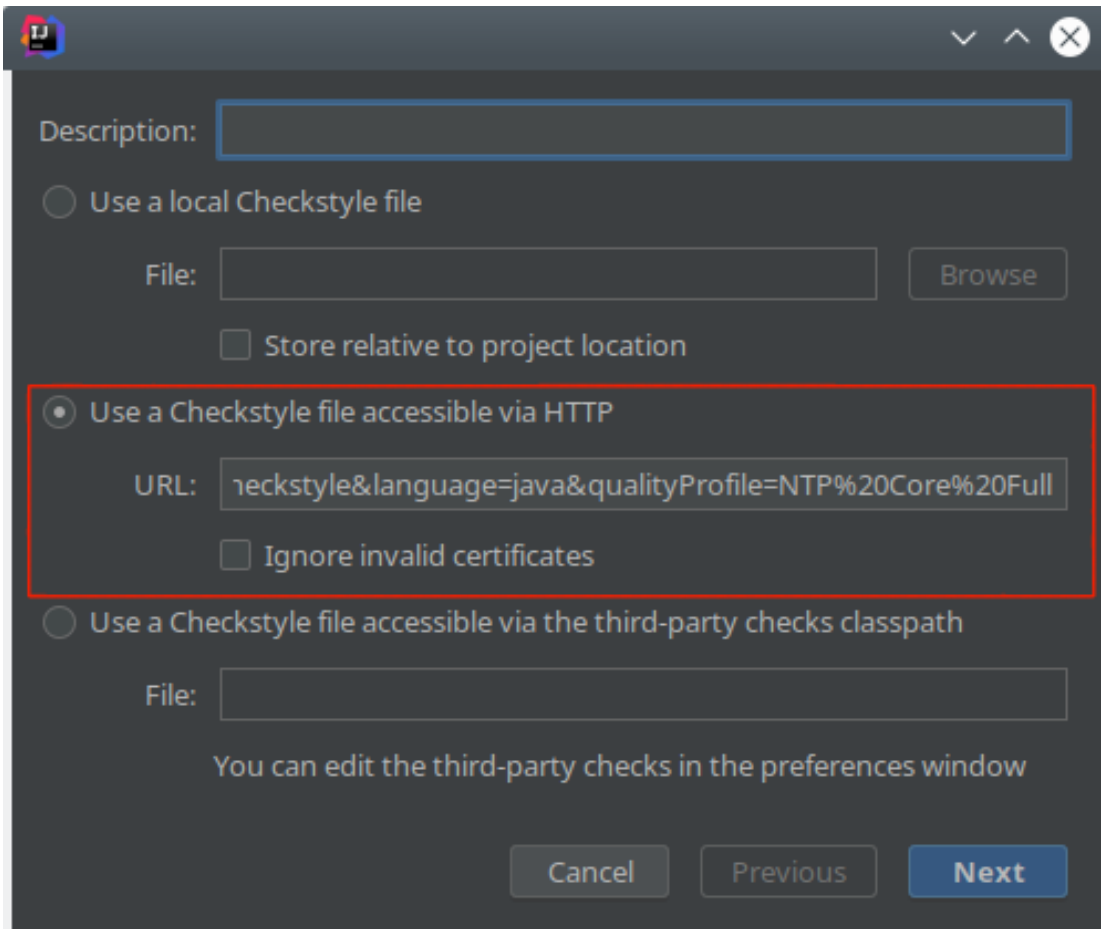

Рисунок 3 – Указание источника конфигурации

После этого плагин уже готов к работе. В нижней части панели появилась опция CheckStyle, при нажатии на которую выводиться окно (рис.4).

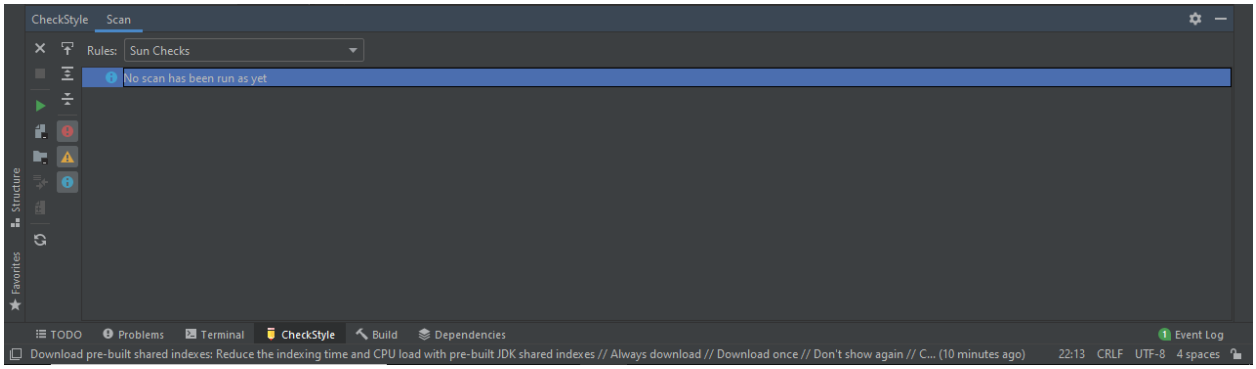

Рисунок 4 – Окно плагина CheckStyle

Для проверки работоспособности плагина написан код, находящий корни дискриминанта (рис.5).

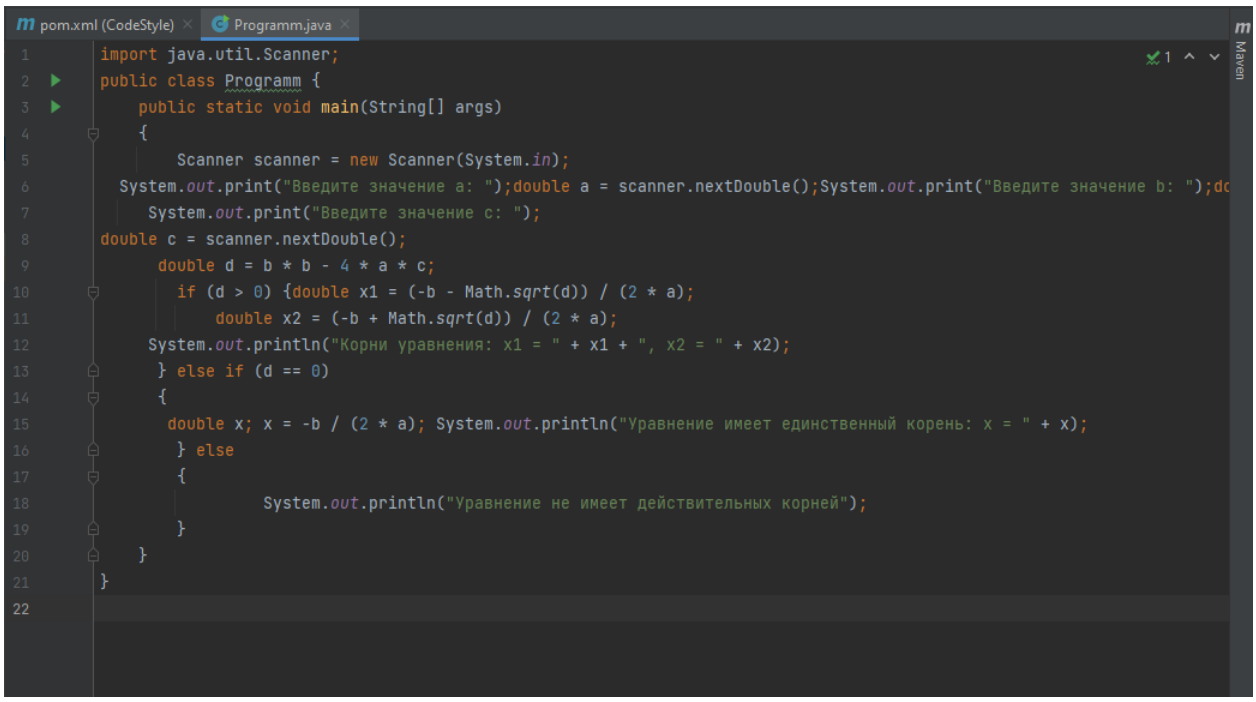

Рисунок 5 – Код для проверки плагина

Далее запускаем плагин (рис. 6).

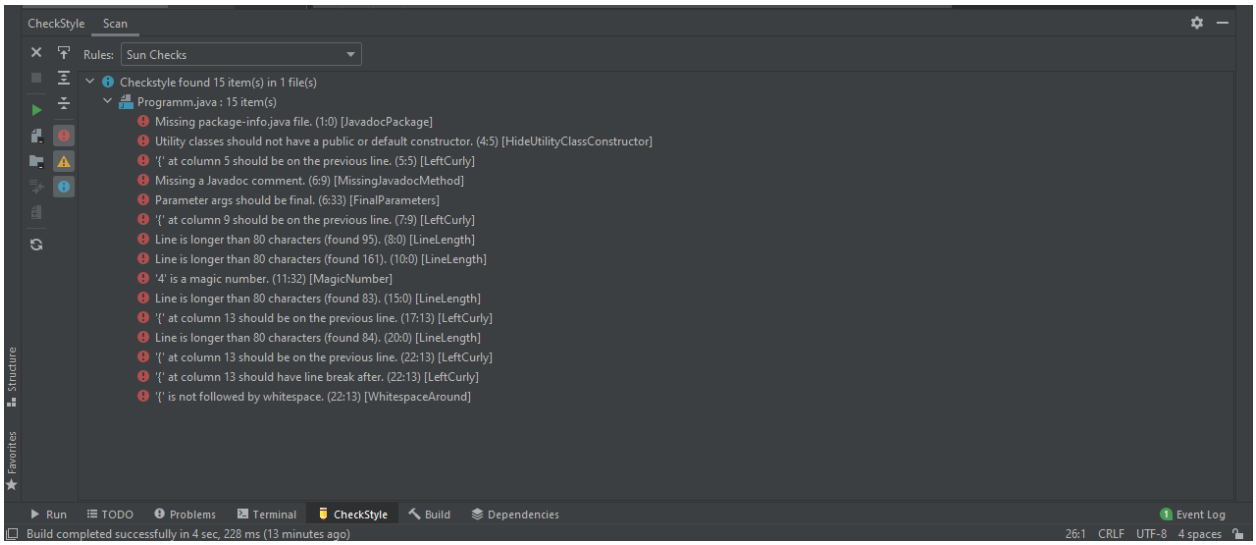

Рисунок 6 – Вывод работы плагина

При анализе кода было выявлено 15 ошибок стиля кода. Каждый вид ошибки расписан и при нажатии на ошибку в консоли, курсор ставится на место ошибки. Теперь необходимо исправить код (рис.7).

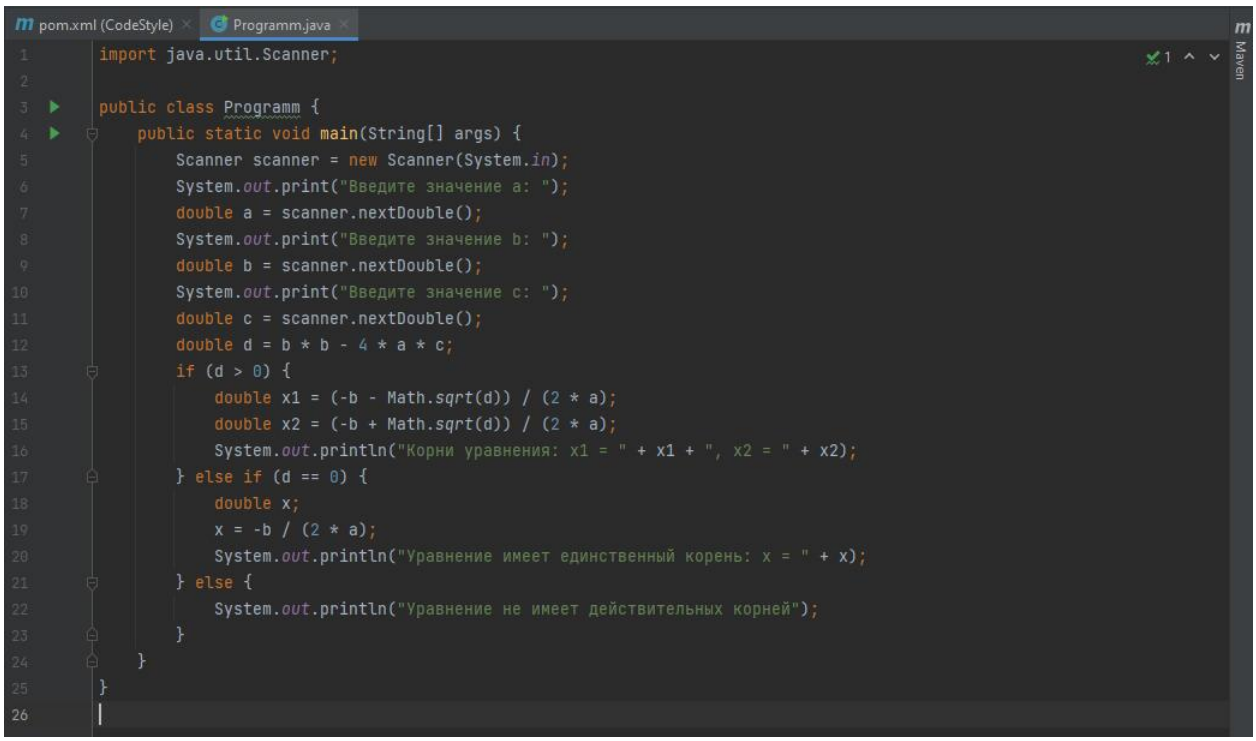

Рисунок 7 – Исправленный код

Исправленный код более компактный и удобный в прочтении, что позволяет намного быстрее понять функционал программы.

Выводы

В данной работе был настроен плагин CheckStyle. Также была выполнена проверка программы на ошибки стилистики кода, впоследствии чего ошибки были исправлены.

Библиографический список

1. Амосов К.А., Волков А.С. Стандарты написания программ: стандарты, регламентирующие использование средств языка и стандарты оформления кода // Научно-техническое и экономическое сотрудничество стран АТР в XXI веке. 2011. С. 160-165.
2. Советов П.Н. Библиотека универсального автоформатирования кода для проблемно-ориентированных языков. // ИТ-Стандарт. № 3 (16). 2018. С. 1-7.
3. Кизим С.С. Методика преподавания дисциплины "Технология WEB-программирования" с применением стандартов оформления кода. // Воспитание и обучение: теория, методика и практика. 2015. С. 318-319.
4. Орлинская О.Г., Куликов А.А. Соблюдение стандартов при разработке программного обеспечения, как один из факторов современного управления качеством. // Современное общество: к социальному единству, культуре и миру. 2016. С. 64-66.
5. Заковоротный А.Ю., Зорян М.А. Модель накопления уязвимостей в исходном коде Java-приложений, а также бинарная система оценки

качества кода на ее основе. // Вестник национального технического университета Харьковский политехнический институт. Серия: информатика и моделирование. № 24 (1300). 2018. С. 37-46.