УДК 004

Настройка системы JUnit в IntelliJ IDEA Community

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
Целью данной статьи является, настройка системы JUnit в среде разработки IntelliJ IDEA Community, а также создание JUnit-теста для тестирования метода, написанного на языке Java.
Ключевые слова: JUnit, тест, IntelliJ IDEA Community, Java

JUnit system setup in IntelliJ IDEA Community

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The goal of this article is to set up a JUnit system in the IntelliJ IDEA Community development environment and create a JUnit test for a test method written in Java.
Keywords: JUnit, test, IntelliJ IDEA Community, Java

1 Введение

1.1 Актуальность

JUnit — библиотека для модульного тестирования программного обеспечения на языке Java. Он необходим для тестирования отдельных участков кода. JUnit позволяет в любой момент быстро убедиться в работоспособности кода. Если программа не является совсем простой и включает множество классов и методов, то для её проверки может потребоваться значительное время, поэтому данный процесс лучше автоматизировать. Использование JUnit позволяет проверить код программы без значительных усилий и не занимает много времени.

1.2 Обзор исследований

В своей работе Р.А. Нагаев, И.С. Полевщиков изучили подход к автоматизации процесса тестирования программного обеспечения с использованием метода разбиения по эквивалентности и анализа граничных значений, а также средства модульного тестирования JUnit [1]. А.С. Тулупцева, Е.С. Кофанова, Е.В. Мельник рассмотрели использование фреймворка JUnit 4 на примере тестирования приложения [2]. И.Г. Кондуров проанализировал модульное тестирования программного обеспечения в языке программирования Java [3]. А.М. Гладун провел обзор и анализ

результатов оценки и апробации инструментов обнаружения дефектов проектирования на примере JUnit и MobileMedia [4]. Г.И. Кожомбердиева, А.М. Сухоногов, Д.А. Протопопов Рассмотрели принципы разработки модульных тестов в интегрированной среде Oracle JDeveloper [5]. В своей работе А.А. Крылов, А.И. Акбулатов рассмотрели методы тестирования Java кода и описали библиотеку JUnit 4, которая во многом упрощает и автоматизирует процесс написания тестов [6].

1.3 Цель исследования

Цель исследования - создать JUnit-тест для тестирования метода, написанного на языке Java, и проверить его.

2 Материалы и методы

Для тестирования программы используется встроенные библиотеки JUnit. С помощью неё выполняется проверка написанного метода на языке Java.

3 Результаты и обсуждения

IntelliJ IDEA работает с несколькими фреймворками тестирования "из коробки", например JUnit, TestNG, Cucumber или Arquillian JUnit.

В среде IDE можно создать тестовый класс непосредственно из исходного кода вместе с необходимыми методами тестирования. Также можно переключаться между классами тестов и исходным кодом с помощью ярлыка, запускать несколько тестов, просматривать статистику для каждого теста и экспортировать результаты тестов в файл.

Для начала необходимо в IntelliJ IDEA создать проект Maven и выбрать SDK (рис.1).

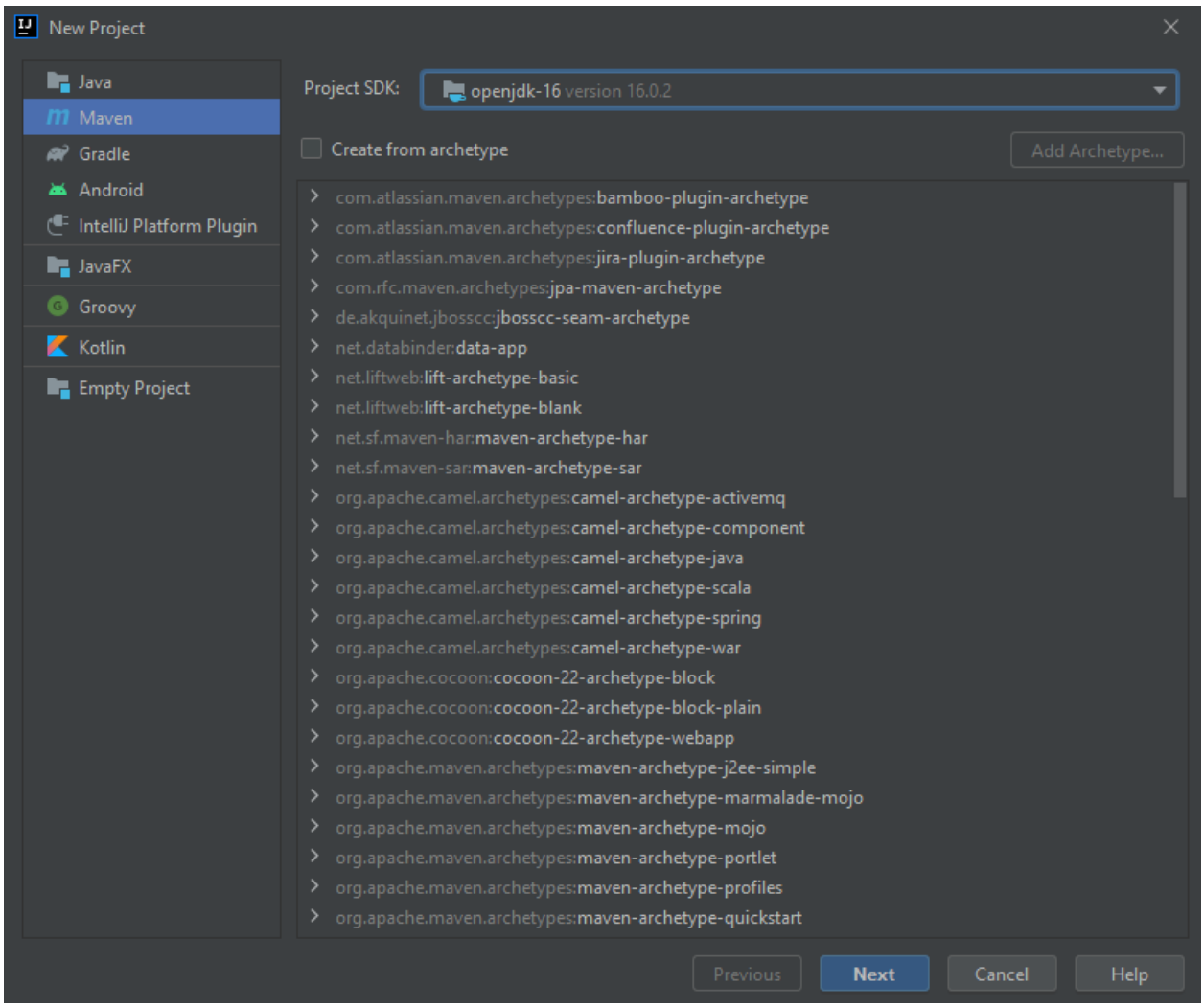

Рисунок 1 – Создание проекта

В созданном проекте появится структура, включающая в себя файл pom.xml, папку main и папку test (рис.2). В файле pom.xml описана информация о проекте в разделе project. Именно тут появится зависимость с библиотекой JUnit.

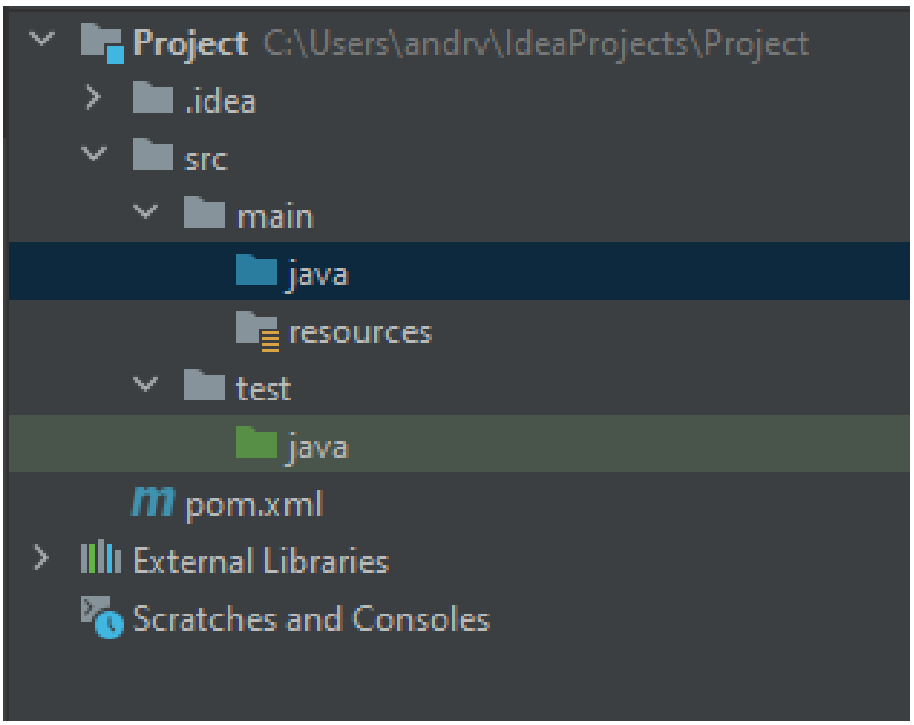

Рисунок 2 – Структура проекта

Теперь необходимо создать тестирующий класс. По умолчанию папка для исходного кода отмечена синим цветом, а папка для тестирования зеленым цветом. В папке main\java создаем Java Class и называем его. Для проверки напишем один метод, считывающий сумму чисел от 1 до числа N (рис.3).

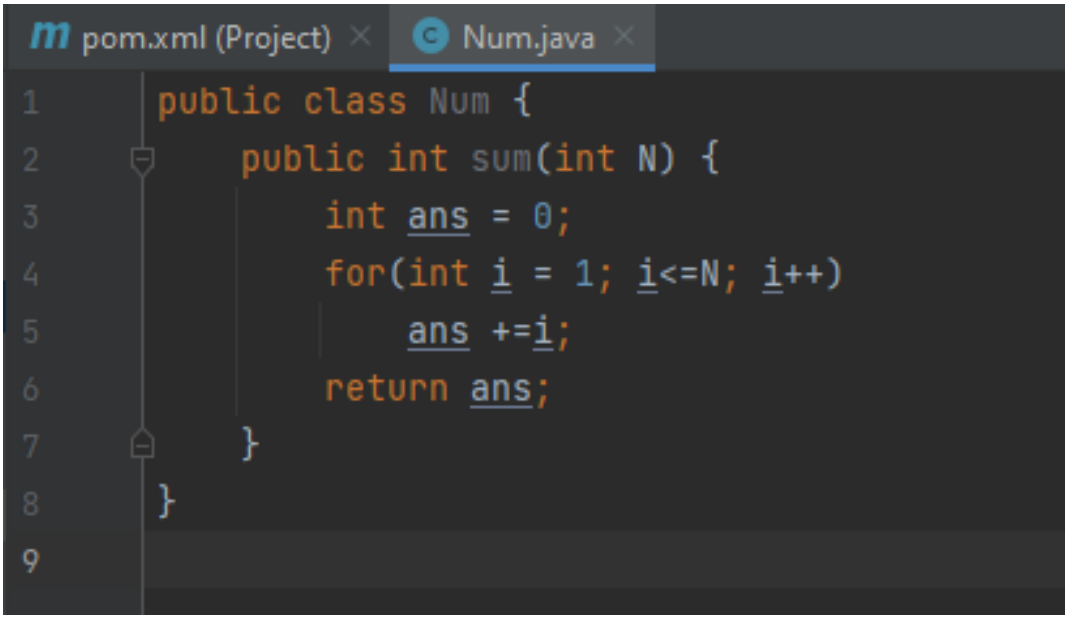

Рисунок 3 – Код метода

В этой же папке создаем класс Program и размещаем функцию main. Далее создаем объект нашего класса (рис.4). В нем будет считываться сумма от 1 до 30.

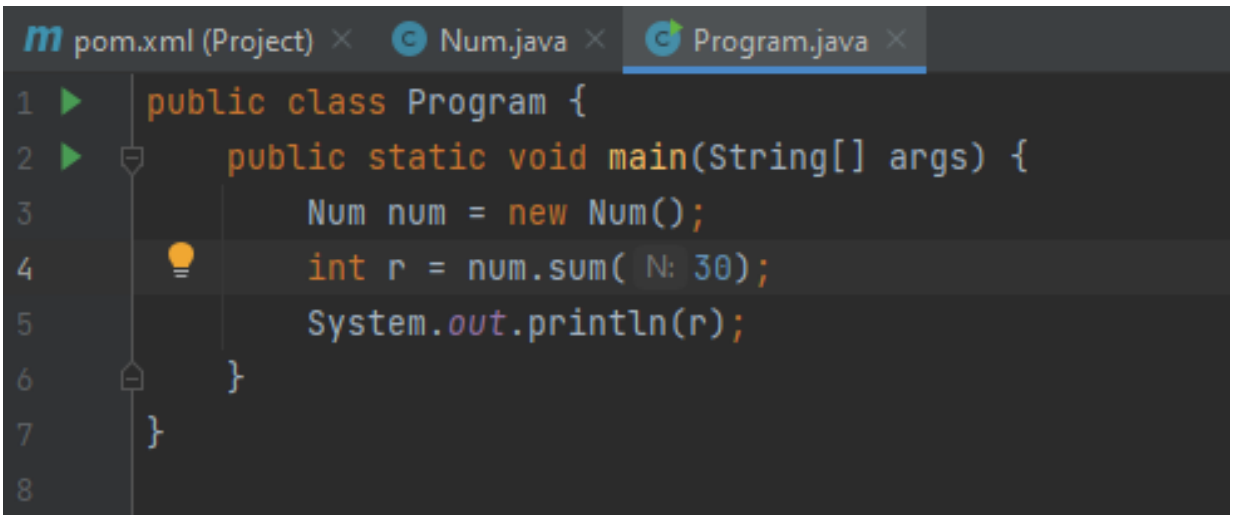

Рисунок 4 – Код класса Program

Далее создаем тест. Для этого обращаемся к классу Num и на строке класса нажимаем сочетание клавиш Alt + Enter, и создаем тест (рис.5).

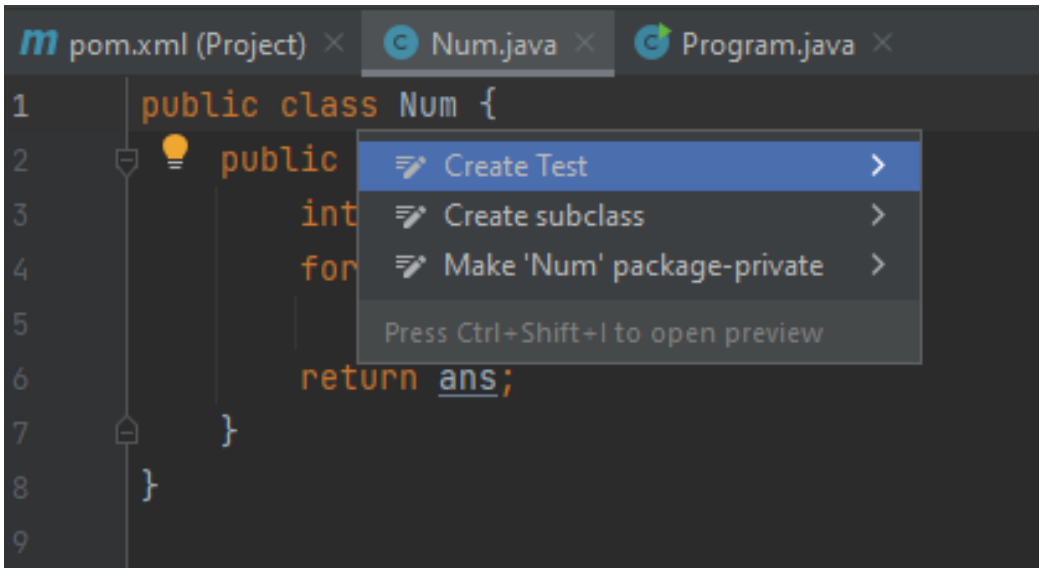

Рисунок 5 – Создание теста

В появившемся окне выбираем библиотеку JUnit 4 и нажимаем Fix. Далее отмечаем те методы класса, которые будут тестироваться (рис.6).

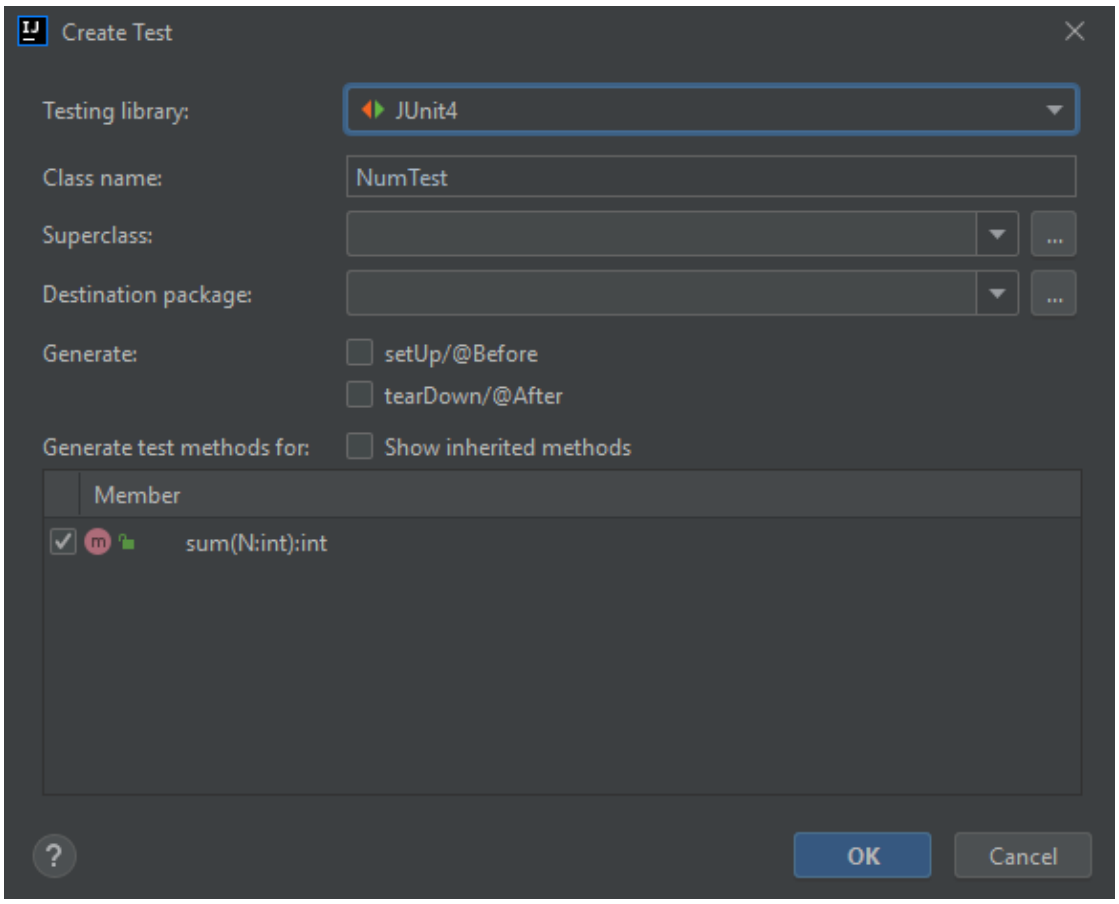

Рисунок 6 – Выбор библиотеки и метода для теста

После этого в папке test\java создается класс NumTest. При этом в файл pom.xml добавилась зависимость с указанной библиотекой JUnit 4. Теперь в файле NumTest набираем код теста. Создаем экземпляр класса Num и переменной actual присваиваем результат работы метода Sum для значения 30. Ожидаемое значение при этом равно 465 (рис.7).

Рисунок 7 – Код теста

Теперь собираем проект и запускаем его. (рис.8). Если где-то будет допущена ошибка, тест не будет пройден, и в консоли будет указанно, какие именно данные не сошлись.

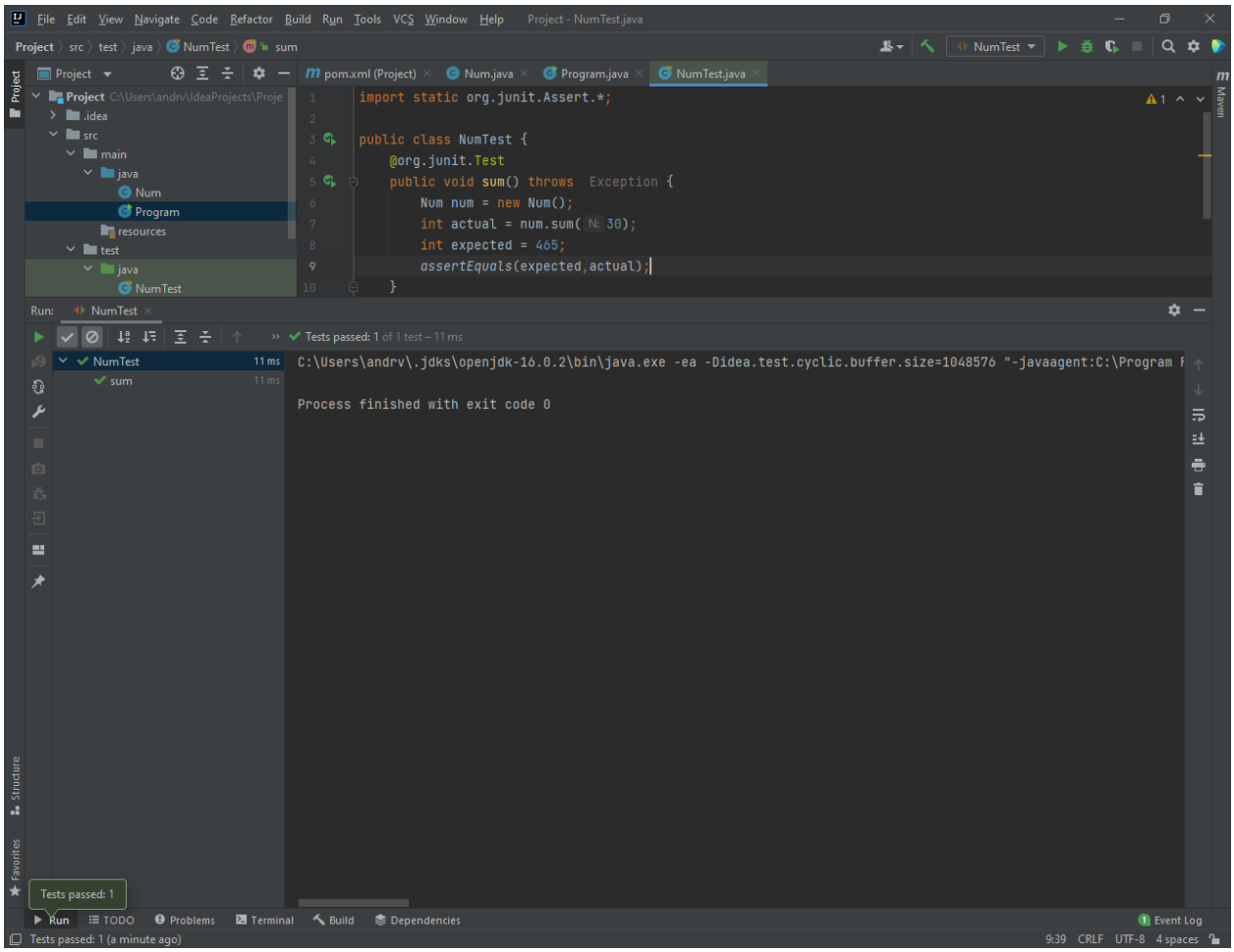

Рисунок 8 – Тест пройден

Выводы

В данной статье была подключена библиотека JUnit. Также был создан и пройден тест, проверяющий правильность работы кода.

Библиографический список

1. Нагаев Р.А., Полевщиков И.С. Автоматизация процесса тестирования программного обеспечения с применением JUnit. // Вестник науки и творчества. 2016. № 5 (5). С. 329-336.
2. Тулупцева А.С., Кофанова Е.С., Мельник Е.В. Использование фреймворка JUnit 4 для тестирования приложений на Java. // Интеллектуальные информационные системы: тенденции, проблемы, перспективы. 2017. С. 185-188.
3. Кондуров И.Г. Модульное тестирование программного обеспечения на Java с применением библиотек JUnit и Mockito. // Программно-техническое обеспечение автоматизированных систем. 2021. С. 11-17.

4. Гладун А.М. Обнаружение и визуализация дефектов проектирования. // Вопросы устойчивого развития общества. 2020. С. 743-750.
5. Кожомбердиева Г.И., Сухоногов А.М., Протопопов Д.А. Использование средств тестирования JUnit при разработке Java-приложений в среде Oracle JDeveloper. // Петербургский государственный университет путей сообщения Императора Александра I. 2014. С. 1-33.
6. Крылов А.А., Акбулатов А.И. Тестирование Java кода. // Развитие инструментов управления научной деятельностью. 2017. С. 75-80.