

Модульное тестирование. JUnit

№ урока: 7 **Курс:** Основы Java EE

Средства обучения: IntelliJ Idea

Обзор, цель и назначение урока

Знакомство с модульным тестированием приложений. Рассмотрение фреймворка JUnit.

Изучив материал данного занятия, учащийся сможет:

- Писать тесты для функционала больших проектов

Содержание урока

1. Введение в модульное тестирование
2. Начало работы с JUnit
3. Аннотации в JUnit
4. Утилитный класс Assert

Резюме

- **Единичное тестирование**, или **модульное тестирование** (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность единицы исходного кода, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.
Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.
Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.
Этот тип тестирования обычно выполняется программистами.
- **JUnit** — библиотека для модульного тестирования программного обеспечения на языке Java. Созданный Кентом Бекем и Эриком Гаммой, JUnit принадлежит семье фреймворков xUnit для разных языков программирования, берущей начало в SUnit Кента Бека для Smalltalk. JUnit породил экосистему расширений — JMock, EasyMock, DbUnit, HttpUnit и т. д.
JUnit был портирован на другие языки, включая PHP (PHPUnit), C# (NUnit), Python (PyUnit), Fortran (fUnit), Delphi (DUnit), Free Pascal (FPCUnit), Perl (Test::Unit), C++ (CPPUnit), Flex (FlexUnit), JavaScript(JSUnit), COBOL (COSUnit).
Опыт, полученный при работе с JUnit, важен в разработке концепций тестирования программного обеспечения.
 - junit.framework.Assert
 - assertEquals
 - assertFalse
 - assertNotNull
 - assertNull
 - assertNotSame
 - assertEquals
 - assertTruejunit.framework.TestCase extends junit.framework.Assert
 - run
 - setUp
 - tearDown

Закрепление материала

- Зачем нужно модульное тестирование?
- Как использовать тестирование Exceptions в JUnit?

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Напишите тесты для каждого из Command в проекте из урока 3

Рекомендуемые ресурсы

Mockito

<http://site.mockito.org/>