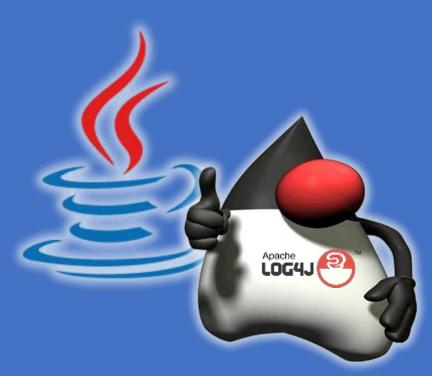


## Laboratory Work #28 (part 3) Using Apache Log4j in Java

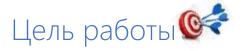


#### LEARN. GROW. SUCCEED.

® 2020-2021. Department: <Software of Information Systems and Technologies> Faculty of Information Technology and Robotics Belarusian National Technical University by Viktor Ivanchenko / ivanvikvik@bntu.by / Minsk

### ЛАБОРАТОРНАЯ РАБОТА #28 (part 3)

# Использование библиотеки де-факто для журналирования Apache Log4j



Изучить технологию журналирования в Java на примере использования широко распространённой библиотеки де-факто Apache Log4j.

## Требования 🔏

- В качестве логгера необходимо использовать ставшую уже стандартом дефакто библиотеку логгирования Apache Log4j (ресурс для загрузки соответствующей библиотеки: http://logging.apache.org/log4j/).
- 2) Скорректировать UML-диаграмму классов и интерфейсов, составляющих архитектуру приложения.
- 3) При проектировании и реализации программы рекомендуется использовать архитектурный шаблон MVC, а также фундаментальные SOLID и GRASP принципы.
- 4) Классы и другие сущности программы должны быть грамотно структурированы по соответствующим пакетам и иметь отражающую их функциональность названия.
- 5) Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом. Работа с консолью должна быть минимальной.
- 6) Для подтверждения работоспособности и адекватности программы, вся модель проекта должна быть покрыта соответствующими модульными тестами. Для модульного (unit) тестирования рекомендуется использовать тестовый фреймворк jUnit или TestNG.
- 7) Необходимо по максимуму пытаться разрабатывать универсальный код.

- 8) Программа должна обязательно быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и её название, версию программы, ФИО разработчика, номер группы и дату разработки.
- 9) Исходный текст классов и демонстрационной программы рекомендуется также снабжать комментариями.
- 10) При разработке программ придерживайтесь соглашений по написанию кода на JAVA (Java Code-Convention).

#### Основное задание

Необходимо произвести рефакторинг приложения предыдущей лабораторной работы таким образом, чтобы результирующие данные сохранялись в лог-файл, а также вести журналирование любых исключительных ситуации, которые могут возникнуть в процессе работы программы.

Дополнительно необходимо расширить иерархию принтеров ещё одним компонентом, вывод которого будет базироваться на базе библиотеки log4j.

Best of LUCK with it, and remember to HAVE FUN while you're learning :) Victor Ivanchenko



### Контрольные вопросы



- 1) Что такое журналирование данных системы и зачем оно нужно?
- 2) Какие есть средства журналирование данных в Java?
- 3) Почему в Java-сообществе разработчиков так популярна библиотека логирования данных Apache Log4j?
- 4) Где можно скачать данную библиотеку, как подключить её к своему проекту, сценарий использования данной библиотеки?
- 5) Из каких основных объектов состоит библиотека Apache Log4j? Опишите в общем жизненный цикл их взаимодействия.
- 6) Иерархия логгеров, как создать объект Logger и для чего он нужен?
- 7) Особенности использования объекта *Logger*, его приоритет и основные методы?
- 8) Как создать объект Appender и для чего он нужен?
- 9) Разновидности Appender-объектов, предоставляемых библиотекой Log4j?
- 10) Как создать объект Layout и для чего он нужен?
- 11) Разновидности *Layout*-объектов, предоставляемых библиотекой Log4j?
- 12) Разновидности спецификаторов шаблонной строки при использовании PatternLayout?
- 13) Каким образом можно сконфигурировать объекты библиотеки Apache Loq4j?
- 14) Куда лучше помещать конфигурационный файл?
- 15) Какую конфигурацию объектов библиотеки Log4j будет подтягивать программная система по умолчанию, если в проекте находится конфигурация в виде properties-файла и в виде xml-файла?