Отчет по практике

выполнил: Чепкасов Владимир

гр.М3305

## Результаты выполнения этапов:

[GIT](https://github.com/DnyashA/Practice)

Этап 2:

Результатом выполнения второго этапа являются бандлы [HelloWorld](https://github.com/DnyashA/Practice/raw/master/Practice/Stage2/HelloWorld-1.0.jar) и [HelloWorldExp](https://github.com/DnyashA/Practice/raw/master/Practice/Stage2/HelloWorldExp-1.0.jar). В бандле HelloWorld содержится реализация сервиса SayHello, который выводит на экран сообщение “Hello OSGi World!”. В бандле HelloWorldExp осуществляется получение ссылки на зарегистрированный в HelloWorld сервис и его использование.

Этап 3:

То же что и этап 2, но с использованием декларативного сервиса. Результатом являются два бандла [DecService](https://github.com/DnyashA/Practice/raw/master/Practice/Stage3/DecService-1.0.jar) и [HelloWorldExp](https://github.com/DnyashA/Practice/raw/master/Practice/Stage3/HelloWorldExp-1.0.jar).

Этап 4:

Результатом выполнения этого этапа является единственный бандл [MyCommand](https://github.com/DnyashA/Practice/raw/master/Practice/Stage4/MyCommand-1.0.jar). В нем содержится реализация метода hello который принимает переменную var типа String и выводит ее в сообщении “Hello, <var>”. При активации регистрируется сервис позволяющий использовать кастомную gogo команду practice:hello. Данная команда вызывает метод hello, описанный выше.

Этап 5:

На этом этапе создается бандл [Lentaparse](https://github.com/DnyashA/Practice/raw/master/Practice/Stage5/LentaParser-1.0.jar). В нем реализуется подсчет наиболее часто встречаемых в заголовках новостей слов и создание gogo команды, которая выводит эти слова и частоту их появления.

Класс Parser содержит реализацию xml парсера и алгоритма подсчета слов, а также их вывод на консоль.

В классе iStats объявляется интерфес, содержащий 2 сигнатуры метода stats: с параметром типа String и без параметров.

Класс Exec имплементирует iStats. В случае ввода команды без параметра или в качестве параметра введен незарегистрированный источник, на консоль будет выведено сообщение о не существовании в реестре источника и перечислены доступные источники. В случае ввода существующего источника будут вызваны методы stats без параметров из классов, соответствующих источников(AIF, Lenta в данном случае), которые так же имплементируют iStats. В случае вызова метода stats() из этих классов будет вызван метод parse(String address) из класса Parser, где address – URL соответствующего источника.

# Выводы.

Главным преимуществом OSGi является возможность изменять части приложения, подключать новые модули или отключать их, при этом не останавливая и не перезапуская его. Это достигается за счет разбиения приложения на модули, при этом также достигается выполнение требования слабой связности, что тоже хорошо. Недостатком, пожалуй, можно назвать некоторую сложность и трудоемкость. Так же проблемы могут возникнуть из-за того, что OSGi «является слоем над Java» и не имеет поддержки в самом языке. OSGi не стоит использовать, например, при разработке расширяемых приложений т.к. при переконфигурировании будут возникать проблемы с производительностью.