# Глава 6

## Использование сторонних библиотек

### Поиск библиотек

Чтобы посмотреть описание некоторых сторонних библиотек, хорошей отправной точкой будет <https://github.com/akullpp/awesome-java> , которая содержит множество Java библиотек и фреймворков.

Проекты Spring, Hibernate, Apache, Eclipse и BouncyCastle предоставляют огромное количество библиотек.  
Прежде чем выбрать open-source библиотеку, вы захотите взглянуть на следующие критерии:

* Документация. Хорошая документация не только помогает вам научиться как использовать библиотеку, но и выступает как хороший индикатор завершенности библиотеки.
* Сообщество. Активное сообщество показывает что библиотека используется. Оно также обеспечивает некоторое представление о том, как те, кто поддерживает библиотеку, относятся к людям, кто задает вопросы.
* Momentum. Проверьте, как часто обновляется библиотека. Вы захотите выбрать библиотеку, которая находится в активной разработке.
* Работает ли это для вас? - Всегда пробуйте каждую библиотеку, чтобы убедиться, что она действительно работает для вашего проекта и что вы можете понять, как использовать библиотеку.
* Лицензия. Удостоверьтесь, что вы можете законно использовать библиотеку.

### Добавление зависимостей проекта

Сторонние библиотеки, которые вы включаете в свой проект, называются зависимостями. Думайте об этом как о том, что ваш проект теперь зависит от этой библиотеки. Вам понадобится следующее:

* GroupId
* ArtifactId
* Version information
* Репозиторий, из которого инструмент сборки сможет скачать библиотеку.

Сторонние open-source библиотеки, которые используются чаще всего, могут быть загружены из большого репозитория Maven Central (http://central.maven.org/maven2/).

Вы можете искать информацию о группе, артефакте и версии на удобном сайте <https://mvnrepository.com/.>

Хорошей и полезной библиотекой с открытым исходным кодом является Apache Commons Lang, которая содержит удобные классы для работы со строками и числами.

### Использование библиотеки Apache Commons Lang

Когда вы используете новую библиотеку, обычно хорошо бы начать с просмотра документации. Для библиотеки утилит Java Javadoc - хорошее первое место для начала.

Для библиотеки Apache Commons Lang вы можете найти Javadoc по адресу <https://commons.apache.org/proper/commons-lang/javadocs/api-release/index.html.>В этом пакете вы найдете действительно удобный набор служебных классов, включающий отличный StringUtils класс. Он обеспечивает множество методов для работы со строками. И, что еще лучше, методы null safe, то есть если вы передадите null строку, ваш код не выбросит исключение.

### Использование модулей

Пакеты Java позволяют вам собирать вместе родственные классы и другие типы. Затем вы можете связать несколько пакетов в jar файл, создавая библиотеку, которую вы можете использовать.

Модули идут еще дальше и позволяют эффективно инкапсулировать библиотеки. Это означает, что вы можете объявить, к каким из общедоступных классов модуля (и других типов) можно получить доступ вне модуля.

Java 9 и далее поддерживают модули, называемые Java Platform Module System, или JPMS.

Кроме того, модули могут объявлять явные зависимости от других модулей. Это помогает навести порядок в путях к классам Java. Вместо поиска классов в пути к классам модуль будет искать напрямую именованный зависимый модуль. Это действительно помогает, когда вы вводите много зависимостей. В большом приложении Java некоторые библиотеки могут зависеть от разных версий одних и тех же библиотек, вызывая всевозможные проблемы. Вместо этого каждый модуль позволяет изолировать его зависимости от остальной части приложения.

Модули ищут в так называемом модульном пути. Путь к модулю перечисляет только модули, а не классы.

Внутри модуля можно экспортировать пакеты в модуле. Если пакет в модуле не экспортируется, то никакой другой модуль не может использовать этот пакет.

Модуль, который хочет использовать код из другого модуля, должен указать, что ему требуется другой модуль. Внутри модуля ваш код может использовать только пакеты, которые экспортируются в зависимом модуле.

Когда вы начнете использовать модули, вы захотите преобразовать каждую созданную вами библиотеку Java в один или несколько модулей. У вас может быть только один модуль на файл JAR.

### Создание модулей

Модуль объединяет набор пакетов Java и дополнительных ресурсов (файлов). Для каждого модуля требуется файл module-info.java, в котором указывается, что модуль экспортирует, а также какие другие модули требуются.