Библиотеки – это такие универсальные компоненты, которые могут использоваться в разных программах.

В сообществе Java-разработчиков популярным стало написание библиотек на все случаи жизни и выкладывание их в общий доступ. Таким образом, Java-программисты могли не писать снова один и тот же код (который зачастую был сыроват и содержал баги), а пользоваться готовыми и проверенными решениями.

Другими словами помимо классов уже написаных и имеющихся в джава (API) существуют *наборы классов* написанных разными разработчиками из разных сообществ которые можно подключать к основному функционалу джава.

**1. Apache Commons**

Apache Commons — это набор высококачественных библиотек Java, включающий множество утилит для повседневных задач. Например, StringUtils предоставляет богатый функционал для работы со строками: убирает нулевые строки, удаляет последние символы строки, способен на другие операции для преобразования и проверки кода. IOUtils облегчает ввод и вывод данных, позволяя обрабатывать потоки более эффективно.

2. Guava

Набор библиотек Guava разработан Google, а это сразу говорит о большом количестве пользователей. Инструментарий Guava расширяет стандартные возможности Java, например предлагает механизмы кеширования для оптимизации работы приложений. Благодаря утилитам для функционального программирования, таким как Optional и Predicate, разработчики могут писать код, соответствующий парадигмам Java 8 и выше.

3. Jackson

Простой и универсальный инструмент для работы с JSON-данными в Java. С его помощью можно легко преобразовать JSON-данные в объекты Java и наоборот. Jackson также поддерживает работу с различными форматами данных, что идеально подходит для задач сериализации и десериализации.

4. Datafaker

Библиотека в основном используется при тестировании. Чтобы протестировать, как система работает с данными, нужна некая база данных. Чем качественнее база, тем проще оценить работу системы. Datafaker — специальная библиотека для Java, которая генерирует фейковые данные, максимально похожие на настоящие. Это могут быть номера телефонов, данные кредитных карт, адреса, коды, и так далее.

5. JUnit

Золотой стандарт для модульного тестирования Java-приложений. С его помощью разработчики могут писать автоматизированные тесты для проверки функциональности своего кода.

6. SLF4J

SLF4J — это API для логирования в Java-приложениях с унифицированным интерфейсом. С помощью SLF4J можно легко переключаться между разными системами логирования, не изменяя код приложения. Это делает SLF4J полезным инструментом в крупных проектах, где могут использоваться различные библиотеки с разными системами логирования.

7. Lombok

Lombok — библиотека Java для оптимизации программного кода. Библиотека интегрируется прямо в среду разработки, где определяет, какие методы или классы повторяются, и предлагает упростить код с помощью аннотаций. Такая оптимизация делает код короче и проще для восприятия, а также позволяет перенести фокус внимания с процесса написания на бизнес-логику продукта.

8. Log4j

Log4j — это библиотека Java для работы с логами. Она помогает управлять логами, настраивать уровень их детализации, архивировать и заменять. Log4j поддерживает настройку шаблонов логов, то есть журнал событий, действий и ошибок может существовать в удобном виде. Эта библиотека облегчает управление журналами в крупных приложениях.

9. TestNG -

Напомню, что для того чтобы работать с этими или любой другой библиотекой надо её подключить в проект, иначе джава просто не поймёт что за классы вы хотите использовать.

Подключать можно добовляя скаченный плагин и указав место его расположения на компе или использовать Maven - так называемый сборщик проектов, основная фишка которого это конфигурационный файл .pom в котором в виде депенденси прописываем библиотеку с которой хотим работать и мавен сам её скачивает с удалённого репозитория к нам на локальную машину, а нам в это время ничего делать не надо, после чего можно спакойно пользоватся теми классами ради которых вы и подгрузили бублиотеку.