**Запуск Java-программы через Docker**

Когда я запускал свою консольную Java-программу (будильник) с помощью Docker, процесс занял минимальное количество времени и оказался максимально удобным:

1. Сначала я **создал исполняемый JAR-файл** из своего Java-кода. Это можно сделать один раз локально, даже на своей машине с IDE (например, IntelliJ IDEA или Eclipse). У меня он назывался Alarm.jar.
2. Затем я **подготовил Dockerfile**, который выглядел так:
3. FROM openjdk:23-slim
4. WORKDIR /app
5. COPY Alarm.jar /app/Alarm.jar
6. CMD ["java", "-jar", "Alarm.jar"]
7. Я **перенёс Dockerfile и JAR-файл на виртуальную машину**, на которой был уже установлен Docker.
8. На виртуальной машине я выполнил всего две команды:

Сборка образа:

* + docker build -t myalarm .

Запуск контейнера:

* + docker run -it myalarm

1. Программа сразу запустилась и была полностью готова к работе.  
   Терминал остался внутри контейнера, и я мог спокойно вводить команды.

**Ничего не нужно было устанавливать** (ни Java, ни компиляторов), всё "зашито" в Docker-образ. Весь процесс был быстрый, автоматизированный, без лишних шагов.  
А главное — теперь я могу перенести этот образ куда угодно, и всё будет работать точно так же.

**Запуск Java-программы без Docker (на чистой виртуальной машине)**

Для сравнения я попробовал запустить ту же самую программу **без Docker** — и сразу ощутил, насколько больше шагов и усилий это требует:

1. **Java на машине не была установлена.**  
   Мне нужно было скачать JDK. Я выбрал JDK 23, как и в Docker-версии, и пошёл на официальный сайт.
2. Я скачал архив с JDK и вручную распаковал его в домашнюю директорию, потому что пакетного менеджера не было.
3. Далее я вручную добавил путь к bin JDK в переменную среды PATH, чтобы могли работать команды java и javac.
4. Только после этого я смог проверить:
5. java -version и javac -version — чтобы убедиться, что Java действительно установлена.
6. Затем я перенёс исходный .java-файл на виртуальную машину.  
   Например, Main.java — мой основной класс программы.
7. Выполнил компиляцию:
8. javac Main.java — иногда могут возникнуть ошибки, если программа использует внешние зависимости. Мне повезло — в этом случае всё обошлось без них.
9. Наконец, я запустил программу:
10. java Main

Всё нужно устанавливать вручную, следить за версиями, прописывать пути, и при этом легко что-то забыть или сделать неправильно.  
И если система будет другая (другой дистрибутив или архитектура), некоторые команды придётся переделывать.

**Мой вывод**

|  |  |  |
| --- | --- | --- |
|  | **Через Docker** | **Без Docker** |
| Установка | Не требуется | Требуется JDK |
| Запуск | 2 команды | Несколько шагов |
| Ошибки среды | Нет | Возможны |
| Переносимость | Отличная | Требует повторной настройки |
| Время | 2–3 минуты | 15–30 минут |