Kali Linux Java commands

write kali linux command for compile,execute, creating api document and many more comamnd that is helpful during learning of java

Here's a **comprehensive list of Kali Linux (and general Linux)** commands that are extremely useful when learning **Java** — including commands for **compiling, executing, debugging, managing files, and generating API documentation**.

🧰 1. Java Installation and Environment Setup

✅ Check if Java is installed

```
bash

java -version
javac -version
```

✅ Install Java (OpenJDK)

```
bash

sudo apt update
sudo apt install default-jdk -y
```

✅ Set JAVA_HOME environment variable

```
bash

echo "export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which java))))" >> ~/.bashrc
echo "export PATH=$PATH:$JAVA_HOME/bin" >> ~/.bashrc
source ~/.bashrc
```

📦 2. Compile and Run Java Programs

✅ Compile a Java file

```
bash

javac HelloWorld.java
```

✅ Run the compiled program

```
bash

java HelloWorld
```

✅ Compile and run in one line

```
bash

javac HelloWorld.java && java HelloWorld
```

✅ Compile all `.java` files in a directory

```
bash

javac *.java
```

✅ Run with classpath

```
bash

java -cp . com.example.MyApp
```

📄 3. API Documentation (Javadoc)

✅ Generate Javadoc for a single file

```
bash

javadoc HelloWorld.java
```

✅ Generate Javadoc for an entire package

```
bash

javadoc -d docs com.example.myapp
```

> **`-d docs`** creates documentation inside a **`docs/`** folder.

✅ Open documentation in browser

```
bash

xdg-open docs/index.html
```

🪛 4. Working with Packages and JAR Files

✅ Compile with packages

```bash
javac -d . com/example/MyClass.java
```

✅ Create a JAR file

```bash
jar cvf MyProgram.jar com/example/*.class
```

✅ Run a JAR file

```bash
java -jar MyProgram.jar
```

✅ View contents of a JAR

```bash
jar tf MyProgram.jar
```

🧩 5. Debugging and Analysis

✅ Run with debugging enabled

```bash
java -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005 HelloWorld
```

✅ Compile with debugging info

```bash
javac -g HelloWorld.java
```

✅ Check syntax only (no class generated)

```bash
javac -Xlint -Werror -d /tmp HelloWorld.java
```

⚙️ 6. Managing Projects (Optional)

✅ Create a simple directory structure

```
bash

mkdir -p src/com/example
mkdir bin
```

✅ Compile all source files into `bin/`

```
bash

javac -d bin src/com/example/*.java
```

✅ Run from `bin/` directory

```
bash

java -cp bin com.example.Main
```

📚 7. Useful Utilities for Learning

✅ Check class dependencies

```
bash

jdeps HelloWorld.class
```

✅ View bytecode (to understand compilation)

```
bash

javap -c HelloWorld
```

✅ Decompile `.class` file (using `javap`)

```
bash

javap -verbose HelloWorld
```

✅ Run interactive JShell (great for learning!)

```
bash

jshell
```

Inside **`jshell`**, you can test Java code line by line:

```
java

int a = 5;
System.out.println(a * 2);
```

🧹 8. Clean and Manage Files

✅ Remove all compiled class files

```
bash

rm *.class
```

✅ Delete docs

```
bash

rm -rf docs/
```

🧠 Bonus: Combining Commands with && and ;

Example:

```
bash

javac MyProgram.java && java MyProgram
```

Or to compile, generate docs, and run:

```
bash

javac MyProgram.java && javadoc MyProgram.java -d docs && java MyProgram
```

Would you like me to create a **cheat sheet (PDF)** with these commands — formatted for quick reference in the terminal (with syntax highlighting and examples)?

make all in tabular format

Here's a **complete tabular version** of all important **Kali Linux / Linux Java commands** — perfect for learners and quick reference 👇

☕ Java Setup & Environment

Task	Command	Description
Check Java version	`java -version`	Shows the installed Java runtime version.
Check compiler version	`javac -version`	Shows the installed Java compiler version.
Install Java (OpenJDK)	`sudo apt update && sudo apt install default-jdk -y`	Installs Java Development Kit (JDK).
Set `JAVA_HOME`	`echo "export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which java))))" >> ~/.bashrc`	Adds JAVA_HOME variable.
Update PATH	`echo "export PATH=$PATH:$JAVA_HOME/bin" >> ~/.bashrc && source ~/.bashrc`	Adds Java bin folder to PATH.

📦 Compile & Run Java Programs

Task	Command	Description
Compile a Java file	`javac HelloWorld.java`	Compiles Java source file into bytecode (`.class`).
Run the program	`java HelloWorld`	Runs compiled Java class.
Compile & Run together	`javac HelloWorld.java && java HelloWorld`	Compiles and runs in one line.
Compile all Java files	`javac *.java`	Compiles every `.java` file in directory.
Run with classpath	`java -cp . com.example.MyApp`	Runs program with specified classpath.

📄 Generate API Documentation (Javadoc)

Task	Command	Description
Generate Javadoc for one file	`javadoc HelloWorld.java`	Creates HTML documentation.
Generate for package	`javadoc -d docs com.example.myapp`	Saves documentation inside `docs/` folder.
Open documentation	`xdg-open docs/index.html`	Opens generated docs in browser.

🧩 Working with Packages & JAR Files

Task	Command	Description
Compile with package	`javac -d . com/example/MyClass.java`	Creates directory structure based on package.
Create JAR file	`jar cvf MyProgram.jar com/example/*.class`	Packages class files into a `.jar`.
Run JAR file	`java -jar MyProgram.jar`	Executes the Java archive.
View contents of JAR	`jar tf MyProgram.jar`	Lists all files inside JAR.

🔪 Debugging & Analysis

Task	Command	Descriptio
Compile with debug info	`javac -g HelloWorld.java`	Adds line number an variable inf
Run in debug mode	`java -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005 HelloWorld`	Enables remote debugging.
Check dependencies	`jdeps HelloWorld.class`	Displays cla dependenc
View bytecode	`javap -c HelloWorld`	Disassembl class bytecode.
Detailed class info	`javap -verbose HelloWorld`	Shows verbose cla structure.

⚙️ Project Structure Commands

Task	Command	Description
Create source & bin directories	`mkdir -p src/com/example && mkdir bin`	Sets up Java project folder structure.
Compile all files to **bin/**	`javac -d bin src/com/example/*.java`	Outputs **.class** files to **bin**.
Run from **bin**	`java -cp bin com.example.Main`	Executes Java main class from bin directory.

💡 Interactive Java (Learning Mode)

Task	Command	Description
Start interactive JShell	`jshell`	Opens Java REPL environment.
Exit JShell	`/exit`	Leaves JShell.
Load script in JShell	`/open myscript.jsh`	Loads code into JShell.

🧹 Clean-Up Commands

Task	Command	Description
Delete class files	`rm *.class`	Removes all compiled bytecode files.
Delete generated docs	`rm -rf docs/`	Deletes Javadoc folder.

🧠 Bonus Combined Commands

Task	Command	Description
Compile, document, and run	`javac MyProgram.java && javadoc MyProgram.java -d docs && java MyProgram`	Full cycle: compile → document → execute.

Task	Command	Description
Compile & run with safety check	`javac MyProgram.java && [ $? -eq 0 ] && java MyProgram`	Runs only if compilation succeeds.

Would you like me to **generate this table as a printable PDF cheat sheet** (with color formatting for commands and quick tips)?