

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side are several concentric circles and arcs, some with degree markings (40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) and arrows, suggesting a technical or scientific theme.

# SOCKET PROGRAMMIERUNG IN JAVA-SE 8

1W1-KAAN KAYGISIZ

# 0 (1) - GLIEDERUNG

1. Überblick
2. Implementierung
3. Protokollbasierende und nicht Protokollbasierende Interaktionen
4. Implementierung in anderen (C-Style) Sprachen

Alle Dateien (inkl. Folien) findet ihr unter:  
[github.com/akb1154/REF-WSEM-CG-FN](https://github.com/akb1154/REF-WSEM-CG-FN)

# 1 (2) – ÜBERBLICK

- Was machen Sockets?
  - Ermöglichen einem Programm eine direkte Maschinenkommunikation (Netzzugriff)
- Seit Wann sind sie in Java?
  - JDK 1.0 (1996)
- Wo kann man Sockets finden?
  - Spiele mit Online-Multiplayer, VoIP/Streaming, Web

# 1 (3) - WICHTIGE KLASSEN

Paket: `java.net`

***Socket***: Eröffnet eine Verbindung zu einer im Konstruktor angegebenen Adresse

***ServerSocket***: Lässt Verbindungen an einem bestimmten Port zu

***InetAddress* / *InetSocketAddress***: Klassen die sich mit Internet-Adressen beschäftigen



# 1 (4) - WICHTIGE KLASSEN

## Paket: `java.io`

***PrintStream***: Wird genutzt um Text zu senden

***InputStream***: Einkommender Datenstream

***OutputStream***: Ausgehender Datenstream

***BufferedReader***: Klasse die zum Lesen genutzt wird

***InputStreamReader***: Leser für Einkommende Datenstreams

***IOException***: Input/Output Fehler

# 1 (5) - WICHTIGE KLASSEN

Paket: `java.util/java.lang`

***Java.util.Scanner***: Kann `java.io.BufferedReader` ersetzen, Klasse die zum lesen genutzt werden kann

***java.lang.Thread***: Kann einen weiteren Anwendungsfaden erschaffen

***Java.util.HashMap<K,V>***: kann einem Schlüssel der Klasse K einen Wert der Klasse V zuordnen

## 2 (6) - IMPLEMENTIERUNG

