## Hauptkonzepte von Sealed-Klassen:

1. **Definition**: Eine sealed-Klasse oder ein sealed-Interface schränkt ein, welche anderen Klassen oder Interfaces sie erweitern oder implementieren können.

2. **Schlüsselwörter**:

- `sealed`: Dieses Schlüsselwort wird verwendet, um eine Klasse oder ein Interface als sealed zu deklarieren.
- `permits`: Mit dem Schlüsselwort `permits` werden die Klassen oder Interfaces angegeben, die die sealed-Klasse erweitern dürfen.
- `final` und `non-sealed`: Diese werden von den Unterklassen der sealed-Klasse verwendet.
    - `final`: Verhindert, dass die Unterklasse weiter abgeleitet wird.
    - `non-sealed`: Erlaubt, dass die Unterklasse ohne Einschränkung weiter abgeleitet werden kann.

**Syntax:**

java — Code kopieren

```java
public sealed class Form permits Kreis, Rechteck, Dreieck {
    // Klasseninhalt
}

public final class Kreis extends Form {
    // Diese Klasse kann nicht weiter abgeleitet werden
}

public non-sealed class Rechteck extends Form {
    // Diese Klasse kann weiter abgeleitet werden
}

public sealed class Dreieck extends Form permits GleichseitigesDreieck {
    // Diese Klasse schränkt ebenfalls die Vererbung ein
}

public final class GleichseitigesDreieck extends Dreieck {
    // Diese Klasse kann nicht weiter abgeleitet werden
}
```

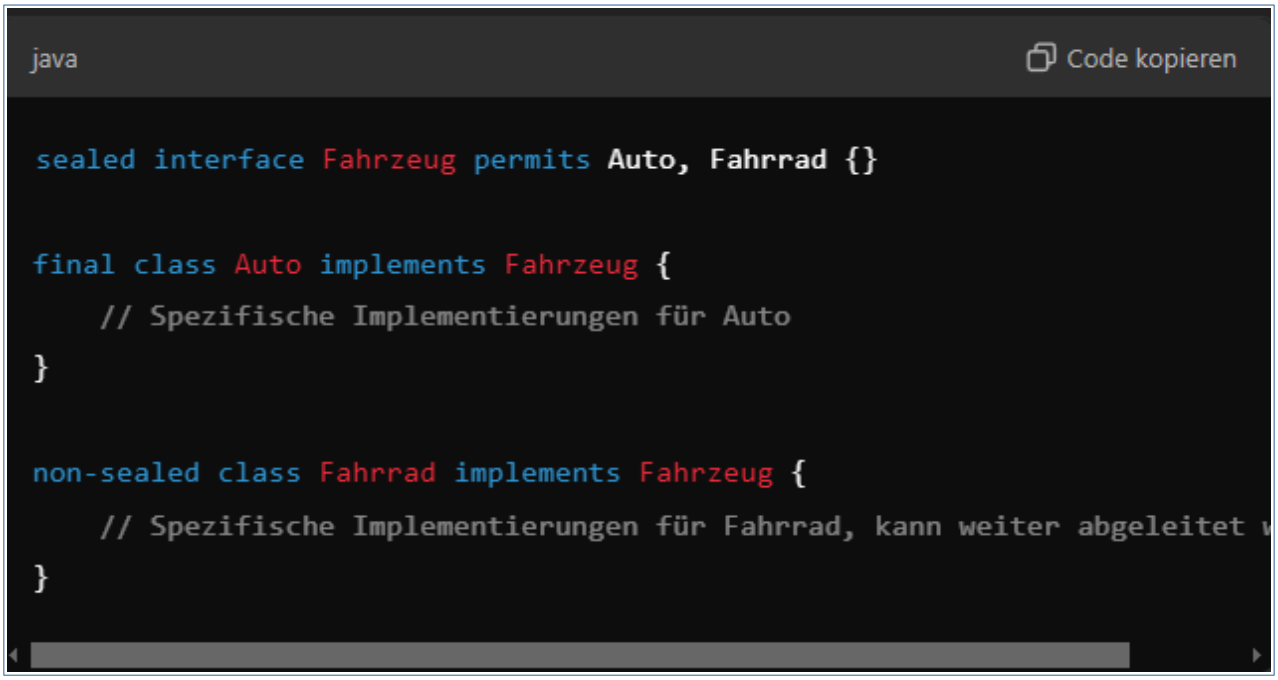
## Regeln für Sealed-Klassen:

- **Nur angegebene Unterklassen**: Die sealed-Klasse muss alle Unterklassen explizit mit dem `permits`-Schlüsselwort angeben.
- **Verhalten der Unterklassen**: Alle Unterklassen müssen entweder `final`, `sealed` oder `non-sealed` sein:
    - `final`: Keine weitere Vererbung erlaubt.
    - `sealed`: Die Unterklasse kann ebenfalls ihre Vererbung einschränken.
    - `non-sealed`: Erlaubt uneingeschränkte Vererbung von dieser Unterklasse.

**Wichtig:**

- Die Subklassen müssen im gleichen Package liegen und existieren
- Ferner müssen sie einen der folgenden Modifier haben
    - sealed
    - non-sealed
    - final

**Schnittellen:**




**In diesem Beispiel:**

- `Fahrzeug` ist ein sealed-Interface, das nur von `Auto` und `Fahrrad` implementiert werden darf.
- `Auto` ist `final`, d. h., keine andere Klasse kann von `Auto` erben.
- `Fahrrad` ist `non-sealed`, d. h., es kann uneingeschränkt weiter abgeleitet werden.

## Einschränkungen:

- **Die sealed-Klasse oder das Interface und alle Unterklassen müssen im selben Modul oder Paket** deklariert sein.
- **Vererbungseinschränkungen**: Eine Unterklasse muss sich für eine der drei Vererbungsoptionen entscheiden (`final`, `sealed` oder `non-sealed`).