„Wildcard-Zeichen `?`“

In Java wird das Wildcard-Zeichen `?` in Generics verwendet, um einen unbekannten Typ darzustellen. Dies ist besonders nützlich in Situationen, in denen der genaue Typparameter nicht von Bedeutung ist und du deinem Code mehr Flexibilität verleihen möchtest. Hier sind einige typische Anwendungsfälle für das Wildcard-Zeichen `?`:

1. Methodenparameter:

- Wildcards kommen häufig bei Methodenparametern zum Einsatz, wenn du eine Sammlung oder einen anderen generischen Typ akzeptieren möchtest, ohne dich auf einen spezifischen Typ festzulegen.
- Beispiel:

```java
public void printList(List<?> list) {
    for (Object element : list) {
        System.out.println(element);
    }
}
```

In diesem Beispiel kann `printList` eine `List` beliebigen Typs akzeptieren, sei es eine `List<Integer>`, `List<String>`, oder etwas anderes. Die Methode bleibt flexibel und funktioniert in jedem dieser Fälle.

2. Obere Begrenzung (`? extends Type`):Upper Bounded

- Diese Variante wird verwendet, wenn du mit einem generischen Typ arbeiten möchtest, ihn aber auf einen bestimmten Typ oder dessen Unterklassen beschränken willst.

- Beispiel:

```java
public void addNumbers(List<? extends Number> list) {
    for (Number number : list) {
        System.out.println(number);
    }
}
```

In diesem Fall kann die Methode `addNumbers` eine `List` von `Number` oder einer beliebigen Unterklasse von `Number` akzeptieren, wie beispielsweise `Integer` oder `Double`.

3. Untere Begrenzung (`? super Type`):Lower Bounded

- Diese Variante kommt zum Einsatz, wenn du mit einem generischen Typ arbeiten möchtest, ihn aber auf einen bestimmten Typ oder dessen Oberklassen beschränken willst.
- Beispiel:

```java
public void addToList(List<? super Integer> list) {
    list.add(42);
}
```

Hier kann die Methode `addToList` eine `List` von `Integer`, `Number` oder sogar `Object` akzeptieren, was dir erlaubt, einen `Integer` hinzuzufügen.

4. Rückgabewerte:Return Types

- Wildcards können auch in Rückgabewerten verwendet werden, wenn die Methode eine Sammlung von unbekannten Typen zurückgeben soll.
- Beispiel:

```java
public List<? extends Number> getNumbers() {
    return new ArrayList<Integer>();
}
```

Diese Methode könnte eine Liste von Untertypen von `Number` zurückgeben, wodurch die Flexibilität beim Rückgabetyp gewährleistet wird.

5. Unbeschränkte Wildcard (`?`):Unbounded Wildcard

- Dies ist die flexibelste Form der Wildcard und wird verwendet, wenn der spezifische Typ nicht von Bedeutung ist.
- Beispiel:

```java
public void processElements(List<?> list) {
    for (Object elem : list) {
        System.out.println(elem);
    }
}
```

Zusammenfassung:
- **Unbeschränkte Wildcard (`?`):** Steht für jeden beliebigen Typ.
- **Obere Begrenzung (`? extends Type`):** Begrenzt auf einen bestimmten Typ oder eine Unterklasse von `Type`.
- **Untere Begrenzung (`? super Type`):** Begrenzt auf einen bestimmten Typ oder eine Oberklasse von `Type`.