

## Pro Java №22

```
public <T> void printElement(T element) {  
    System.out.println(element);  
}
```

В данном примере метод `printElement` параметризован типом `T`, что позволяет использовать его с любым типом данных.

Дженерики не работают с массивами, но работают с коллекциями.

Вайлдкарты (wildcards) в дженериках используются для обеспечения гибкости и позволяют параметризованным типам принимать более общие или более конкретные типы данных. В Java есть три основных вида вайлдкарт:

1. **\*\*Неограниченные вайлдкарты\*\*** (`?``)
2. **\*\*Ограниченные верхней границей вайлдкарты\*\*** (`? extends Type``)
3. **\*\*Ограниченные нижней границей вайлдкарты\*\*** (`? super Type``)

### Неограниченные вайлдкарты (`?``)

Неограниченные вайлдкарты используются, когда тип не имеет значения. Например:

```
List<?> list = new ArrayList<String>();
```

Здесь список может содержать элементы любого типа.

### Вайлдкарты, ограниченные верхней границей (`? extends Type``)

Эти вайлдкарты позволяют использовать типы, которые являются подтипами указанного типа. Например:

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

В данном случае метод `process` может принимать список чисел (`List<Number>``) или список подтипов чисел, таких как `List<Integer>`` или `List<Double>``.

### Вайлдкарты, ограниченные нижней границей (`<? super Type>`)  
Эти вайлдкарты позволяют использовать типы, которые являются суперклассами указанного типа. Например:

```
public void addNumbers(List<? super Integer> list) {  
    list.add(10);  
    list.add(20);  
}
```

Lombok – это библиотека для Java, которая помогает уменьшить шаблонный код с помощью аннотаций. Некоторые из полезных аннотаций Lombok включают:

- `@Getter` и `@Setter`: автоматически создают геттеры и сеттеры для полей класса.
- `@NoArgsConstructor`: создаёт конструктор без аргументов.
- `@AllArgsConstructor`: создаёт конструктор со всеми аргументами.
- `@ToString`: создаёт метод `toString()`.
- `@EqualsAndHashCode`: создаёт методы `equals()` и `hashCode()`.