

Adnotacje

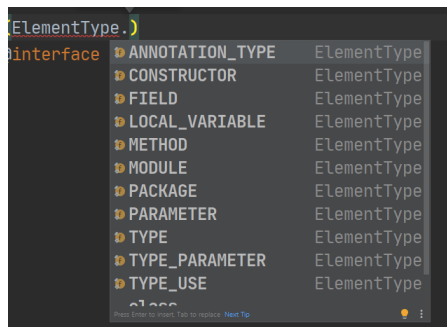
Adnotacje to meta dane, które dostarczają informacje o kodzie, którymi można później manipulować. Nie mają bezpośredniego wpływu na napisany kod oznaczony tą adnotacją.

Wykorzystanie adnotacji:

Tworzenie własnych adnotacji

Adnotacje nie mogą dziedziczyć, ale wszystkie rozszerzają **interface Annotations**.

@Target - pozwala zdefiniować typy deklaracji, dla których będziemy mogli stosować naszą adnotację.



Możemy użyć więcej niż jednej adnotacji, ale musimy wtedy zastosować klamry.

W przypadku pominięcia `@Target`, przyjmuje się, że może być stosowane wszędzie poza adnotacją typów.

@Retention - by nasza adnotacja cechowała się konkretnym zachowaniem. Możliwe są 3 tryby:



SOURCE – występuje tylko w kodzie źródłowym (informacje dla kompilatora)

Np.

```
@Override
```

czyli sprawdź czy metoda jest faktycznie nadpisana

CLASS – występuje tam gdzie SOURCE + w pliku skompilowanym class (a więc przetwarzane w trakcie kompilacji)

Np.

```
@SuppressWarnings("unchecked")
```

Czyli nie ostrzegaj nas o zagrożeniach typu **unchecked**.

RUNTIME – przetwarzane w trakcie uruchomienia programu

Służy do tego mechanizm refleksji.

Np.

```
@PostConstruct.
```

W przypadku pominięcia *@Retention* przyjmuje się, że została użyta CLASS.

Przykładowa adnotacja

<https://github.com/idzikpro/JavaBasics/blob/master/src/main/java/pl/idzikpro/annotation/MyAnnotation.java>

```
@Target({
    ElementType.FIELD,
    ElementType.METHOD,
    ElementType.CONSTRUCTOR
})
@Retention(RetentionPolicy.RUNTIME)
public @interface MyAnnotation {
    String value1() default "hello";

    String value2();
}
```

i jej zastosowanie w klasie

<https://github.com/idzikpro/JavaBasics/blob/master/src/main/java/pl/idzikpro/annotation/Person.java>

```
@MyAnnotation(value1 = "field", value2 = "name")
private String name;
private int age;

@MyAnnotation(value2 = "Person")
public Person(String name, int age) {
    this.name = name;
    this.age = age;
}

@Override
```

```
@MyAnnotation(value1 = "method", value2 = "toString")
public String toString() {
    return "Person{ " +
        "name=" + name + " " +
        ", age=" + age +
        '}';
}
```

Teraz, aby odczytać adnotację wystarczy użyć mechanizmu refleksji.

<https://github.com/idzikpro/JavaBasics/blob/master/src/main/java/pl/idzikpro/annotation/GetAnnotationFromClass.java>

```
for (Annotation annotation : Person.class.getMethod("toString").getDeclaredAnnotations()
) {
    System.out.println(annotation.annotationType().getName());
}
```

Przykładowe adnotacje

Android : @ButterKnife

Java : rodzina Lombok

Hibernate : @Entity