

# Domain-Specific Languages

## Internal DSL case study

### in Groovy Language

Tomasz Janek

*Ośrodek Przetwarzania Informacji Instytut Badawczy  
Laboratorium Inteligentnych Systemów Informatycznych*

[tomasz.janek@opi.org.pl](mailto:tomasz.janek@opi.org.pl)

# Co to jest DSL ?

---

**Język dziedzinowy (Domain-Specific Language) -**  
specjalnie utworzony język programowania z  
myślą o rozwiązaniu pewnej specjalistycznej  
dziedziny problemów

# Cele języka dziedzinowego

---

- Skuteczny i przejrzysty interfejs w celu zapewnienia sprawnej interakcji użytkowników z aplikacją
- Mały i prosty język ( działający na wyższym poziomie abstrakcji niż język ogólnego zastosowania)
- Oddzielenie logiki biznesowej od kodu aplikacji
- Eksperti dziedzinowi mogą implementować logikę biznesową

# Typy języków dziedzinowych

---

- **Zewnętrzne:**

*XML, SQL, Ant, makefile*

- **Wewnętrzne:**

*Spock, Geb, EasyB, Rake*

# Zalety

---

- Zwiększa produktywność
- Przenośny
- Samo dokumentujący się kod
- Łatwy do opanowania język
- Walidacja na poziomie dziedziny

# Wady

---

- Wysoki koszt zaprojektowania, implementacji oraz późniejszego utrzymania dobrego DSL
- Słabe wsparcie narzędzi lub ich brak
- Ograniczone zastosowanie

# Dlaczego Groovy

---

- W pełni obiektowy, dynamicznie typowany, skryptowy język programowania
- Wsparcie dla meta programowania:
  - „Otwarte klasy” (meta klasy, kategorie)
  - Builders
  - Transformacje AST
- Command Chain Expressions
- Domknięcia
- Przeciążanie operatorów

# Przykład

---

## Język dziedzinowy dla walidacji poprawności modelu danych



# Przykład

---

## Walidacja poprawności modelu danych - przykład użycia

```
walidacja dla sekcji 1 modelu Pierwszego {  
  ostrzezenie "Komunikat - ostrzezenie", P01 > P02  
  informacja "Komunikat - informacja", P01 < 5  
  blad "Komunikat - blad", P03 >= 40  
} wyniki << wynik
```

# Przykład

---

## Walidacja poprawności modelu danych - integracja z aplikacją

```
def shell = new GroovyShell(binding)
def messages = shell.evaluate(
    new File('validator.groovy').text +
    new File('validation.gdsl').text
)
messages.flatten().each { message ->
    // operacje związane z przetwarzaniem wyników walidacji
```

# Bibliografia

---

- *Martin Fowler* „**Domain-Specific Languages**”, Addison-Wesley Professional, 2010
- *Debasish Ghosh* „**DSLs in Action**”, Manning Publications, 2010
- *Terence Parr* „**Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Language**”, Pragmatic Bookshelf, 2010
- *Terence Parr* “**The Definitive Antlr Reference: Building Domain-Specific Languages**”, Pragmatic Bookshelf, 2007