

# Table of Contents

1. Tworzenie połączenia bazowanego.....	1
2. Tworzenie prostego szablonu projektu : Hibernate only.....	1
3. Tworzenie prostego szablonu projektu : JPA .....	1
4. Tworzenie prostego szablonu projektu : Spring + Spring Data .....	1
5. Stworzenie i zapisanie prostej encji oraz jej odczyt .....	1
6. Mapowania .....	1
7. MetaModel .....	1
8. JPQL.....	2
9. Pobieranie danych : Fetches .....	2
10. Cache.....	2
11. Criteria API .....	2
12. Przykłady.....	2
12.1. Tematy pokrewne : Json : ograniczanie wyników : BootRest : User + UserOptController ....	2
12.2. Setupowanie Spring/Spring Data/Hibernate oraz przykłady użycia i tworzenia repozytoriów	2
12.3. Zadanie przetestowanie : repozytorium i relacji .....	2
12.4. Utworzenie Specyfikacji oraz customizacja repozytorium .....	2
12.5. Dodanie cache .....	2
12.6. Metamodel : Jpametamodel : RepositoryTest .....	2
12.7. Lock : przykład : SpringDataKata : ItemRepository.....	3
12.8. Projections .....	3
12.9. Like .....	3
13. Zadanie skonfiguruj projekt min : baza , dostawca , transakcje .....	3
13.1. przykład :projekt HibernateKata : H2Database.....	3
13.2. przykład : HibernateKata : JpaConfigTest.....	3
13.3. przykład : Konfiguracja Hibernate SessionFactory : HibernateKata : AbstractHibTest ....	3
14. JDBC : HibernateKata : JdbcConfig.....	3
15. Prosta konfiguracja JPA do testów : HibernateKata : TestSelectorConfig .....	3
16. Konfiguracja pojedynczej encji w kontekście do testów : HibernateKata :	3
AbstractConcreteClazzHibTest	
16.1. użycie w praktyce : HibernateKata : ElementCollectionTest .....	3
17. konfiguracja z EntityManagerFactory .....	4
17.1. przykład HibernateKata : SimpleUseJpaTemplateTest .....	4
18. Podpinanie DS w Spring.....	4
18.1. przykład : SpringJpaKata : DataSourceTest .....	4
19. Integracja z Spring + audyt SQL <a href="http://przewidywalna-java.blogspot.com/2014/10/quick-spring-data-sql-audit.html">http://przewidywalna-java.blogspot.com/2014/10/quick-spring-data-sql-audit.html</a>	4
20. Klucze .....	4
20.1. przykład : HibernateKata : pl.java.scalatech.generator.....	4

20.2. uuid,hilo, table,seq,identity,assigned .....	4
20.3. złożone .....	4
21. JPA 2.1 features .....	4
21.1. konwerter : HibernateKata : SimpleProduct ,LuxuryItem.....	4
21.2. przykład : SpringDataKata : CartTest .....	4
21.3. namedQueryRuntime : SpringJpaKata : JpqlTest.shouldProgrammaticallyNamedQueries ..	5
21.4. ConstructorResult : SpringJpaKata : ConstructorResultTest .....	5
22. Walidator .....	5
22.1. przykład SpringJpaKata : ValidationPersonTest .....	5
22.2. zadanie : przeprowadz test wybranej encji w projekcie .....	5
23. Pierwsza encja .....	5
23.1. Zadanie stworzyć i dodać encje Person .....	5
23.2. Zadanie stworzyć i dodać encje Person .....	5
23.3. skorzystać z EntityManager .....	5
23.4. przykład : HibernateKata : MessageQueryTest .....	5
23.5. skorzystać z Session (Hibernate) z prawidłową obsługą .....	5
23.6. zakładanie ograniczeń .....	5
23.7. Zadanie użyć secondaryTable .....	6
23.8. Zadanie praca z Lob .....	6
23.9. Zadanie : selfRefence.....	6
24. Dziedziczenie .....	6
24.1. zadanie : zastosować każdy rodzaj dziedziczenia .....	6
25. Tworzenie zapytań .....	6
25.1. Spring Data .....	6
25.2. JPQL.....	6
25.3. Stworz zaytania natywne.....	6
25.4. Stwórz zapytania nazwane .....	6
25.5. Criteria API .....	7
25.6. Example .....	7
25.7. QUERY_DSL .....	7
25.8. CRUD .....	7
26. Praca z kolekcjami.....	7
26.1. Zastosować ElementCollection .....	7
27. Asocjacje .....	8
27.1. many2many extra fields.....	8
28. Wydajność.....	8
28.1. batch .....	9
29. Problem z Lazy.....	9
30. n+1 Problem .....	9
30.1. zadanie rozwiązanie problemu lazyInitializationException .....	10
30.2. FetchProfile .....	10

30.3. Eager problem .....	11
31. Cache .....	11
31.1. firstLevelCache test : SpringJpaKata : FirstLevelCacheTest.....	11
31.2. wyjaśnienie zasady działania <a href="http://przewidywalna-java.blogspot.com/search?q=hibernate">http://przewidywalna-java.blogspot.com/search?q=hibernate</a>	11
31.3. zadanie : wprowadz cache do projektu .....	11
32. Lock.....	11
32.1. dodać wersjonowanie.....	11
33. DTO .....	11
33.1. Zadanie wprowadz DTO , spłaszcz model domenowy .....	11
34. Pułapki i dobre praktyki .....	11
35. Podsumowanie :.....	11
36. Koniec :) .....	12

# 1. Tworzenie połączenia bazowanego

## 1.1. Tworzenie puli połączeń na przykładzie HikariCP

# 2. Tworzenie prostego szablonu projektu : Hibernate only

operowanie na obiektach Session i SessionFactory oraz persistence.xml

# 3. Tworzenie prostego szablonu projektu : JPA

operowanie na obiektach EntityManager i EntityManagerFactory oraz persistence.xml

# 4. Tworzenie prostego szablonu projektu : Spring + Spring Data

integracja Hibernate po JPA w Spring

# 5. Stworzenie i zapisanie prostej encji oraz jej odczyt

# 6. Mapowania

## 6.1. Ćwiczenia:

- Stwórz relacje :
  - Person ma wiele samochodów (@OneToMany), mieszka w jednym z wielu krajów (@ManyToOne), na jedną z wielu ról (@ManyToMany)
  - Country
  - Car
  - Role
- Zasil obiekty poprzez test

# 7. MetaModel

## 7.1. Ćwiczenie stwórz sam metamodel do poprzenia

## **8. JPQL**

**8.1. Ćwiczenie napisz kwerendy do przykładów**

## **9. Pobieranie danych : Fetches**

## **10. Cache**

**10.1. Ćwiczenie wprowadź cache**

## **11. Criteria API**

## **12. Przykłady**

**12.1. Tematy pokrewne : Json : ograniczanie wyników :  
BootRest : User + UserOptController**

**12.2. Setupowanie Spring/Spring Data/Hibernate oraz  
przykłady użycia i tworzenia repozytoriów**

**12.2.1. Audit : SpringDataKata : AuditTest**

**12.2.2. Slice, Paging, Async, Stream : SpringDataKata:  
SpringDataRepositoryTest**

**12.2.3. QBE : SpringJpaKata : JpqlByExampleTest:shouldUseExample()**

**12.3. Zadanie przetestowanie : repozytorium i relacji**

**12.4. Utworzenie Specyfikacji oraz customizacja  
repozytorium**

**12.5. Dodanie cache**

**12.6. Metamodel : Jpametamodel : RepositoryTest**

## **12.7. Lock : przykład : SpringDataKata : ItemRepository**

## **12.8. Projections**

**12.8.1. przykład : SpringJpaKata : PersonRepoTest**

## **12.9. Like**

**12.9.1. przykład : SpringDataKata : PersonRepoTest.shouldLikeWork**

## **13. Zadanie skonfiguruj projekt min : baza , dostawca , transakcje**

**13.1. przykład :projekt HibernateKata : H2Database**

**13.2. przykład : HibernateKata : JpaConfigTest**

**13.3. przykład : Konfiguracja Hibernate SessionFactory : HibernateKata : AbstractHibTest**

## **14. JDBC : HibernateKata : JdbcConfig**

**15. Prosta konfiguracja JPA do testów : HibernateKata : TestSelectorConfig**

**16. Konfiguracja pojedynczej encji w kontekście do testów : HibernateKata : AbstractConcreteClazzHibTest**

**16.1. użycie w praktyce : HibernateKata : ElementCollectionTest**

## **17. konfiguracja z EntityManagerFactory**

**17.1. przykład HibernateKata :**  
**SimpleUseJpaTemplateTest**

## **18. Podpinanie DS w Spring**

**18.1. przykład : SpringJpaKata : DataSourceTest**

## **19. Integracja z Spring + audyt SQL**

<http://przewidywalna-java.blogspot.com/2014/10/quick-spring-data-sql-audit.html>

## **20. Klucze**

**20.1. przykład : HibernateKata :**  
**pl.java.scalatech.generator**

**20.2. uuid,hilo, table,seq,identity,assigned**

**20.3. złożone**

**20.3.1. przykład : HibernateKata : UserId , test : MapKeyTest**

## **21. JPA 2.1 features**

**21.1. konwerter : HibernateKata : SimpleProduct  
,LuxuryItem**

**21.2. przykład : SpringDataKata : CartTest**

**21.3. namedQueryRuntime : SpringJpaKata :  
JpqlTest.shouldProgrammaticallyNamedQueries**

**21.4. ConstructorResult : SpringJpaKata :  
ConstructorResultTest**

## **22. Walidator**

**22.1. przykład SpringJpaKata : ValidationPersonTest**

**22.2. zadanie : przeprowadz test wybranej encji w projekcie**

## **23. Pierwsza encja**

**23.1. Zadanie stworzyć i dodać encje Person**

**23.2. Zadanie stworzyć i dodać encje Person**

**23.3. skorzystać z EntityManager**

**23.3.1. przykład: HibernateKata :  
EntityFactoryTest,SimpleUseJpaTempateTest**

**23.4. przykład : HibernateKata : MessageQueryTest**

**23.5. skorzystać z Session (Hibernate) z prawidłową obsługą**

**23.6. zakładanie ograniczeń**

**23.6.1. Zadanie założyć ograniczenia po stronie bazy i po stronie encji**

**23.6.2. przykład : HibernateKata : BidCheck , User ,UserConstraintTwo,  
UserTableConststaint**



## **23.7. Zadanie użyć secondaryTable**

23.7.1. przykład : HibernateKata: Address

## **23.8. Zadanie praca z Lob**

23.8.1. przykład : HibernateKata : pl.java.scalatech.domain.exercise2.Item

## **23.9. Zadanie : selfRefence**

23.9.1. Przykład : HibernateKata SelfCategoryTest

23.9.2. Przykład 2 : HibernateKata : SelfReferenceTest

## **24. Dziedziczenie**

### **24.1. zadanie : zastosować każdy rodzaj dziedziczenia**

24.1.1. HibernateKata :pl.java.scalatech.domain.inherit

24.1.2. SpringJpaKata :test : pl.java.scalatech.exercise.inheritance

## **25. Tworzenie zapytań**

### **25.1. Spring Data**

### **25.2. JPQL**

25.2.1. Natywne

przykład :SpringJpaKata : JpaNativeTest

przykład : SpringJpaKata : Todo , ConstructorResultTest

### **25.3. Stworz zaytania natywne**

### **25.4. Stwórz zapytania nazwane**

## 25.5. Criteria API

25.5.1. Zadanie : stwórz metamodel

25.5.2. przykład : jpaMetaModel

<https://github.com/przodownikR1/jpaMetaModel>

25.5.3. proste przykłady : HibernateKata : SelectTest

25.5.4. Zadanie stwórz implementacje DAO API do testu

25.5.5. przykład : SpringJpaKata : JpaCriteriaTest

## 25.6. Example

25.6.1. Zadanie stwórz zapytanie QBE

25.6.2. przykład : SpringJpaKata : JpqlByExampleTest

## 25.7. QUERY\_DSL

25.7.1. Stwórz prosty DSL

[https://github.com/przodownikR1/queryDSL\\_springData](https://github.com/przodownikR1/queryDSL_springData)

## 25.8. CRUD

25.8.1. trwałość przez osiągalność przykład projekt : HibernateKata :  
HibernateFactoryServiceTest.java : test (shouldSaveThenLoadItem)

## 26. Praca z kolekcjami

### 26.1. Zastosować ElementCollection

#### 26.1.1. Set

przykład : HibernateKata : EmbeddedUser

przykład : HibernateKata : UserColl

#### 26.1.2. Collection

przykład : HibernateKata : Item

### 26.1.3. Map

przykład : HibernateKata : UserMap

embedded : HibernateKata : Employee

przykład kodu : ElementCollectionTest

## 27. Asocjacje

### 27.1. many2many extra fields

przykład : pl.java.scalatech.domain.manyToMany

#### 27.1.1. many2manyBidirectional

przykład : ManyToManyTestBi

#### 27.1.2. many2one

przykład : ManyOneTest

przykład : SelfReferenceTest

#### 27.1.3. one2many

przykład : One2ManyTest

przykład : One2ManyBiTest

przykład : SelfReferenceTest

#### 27.1.4. one2one

przykład : One2OneBiTableTest

przykład : One2OneJoinTableTest

przykład : One2OneJoinTableTest , One2OneOrdinaryTest

przykład OneToOneTest

przykład : One2OnePrimaryTest

## 28. Wydajność

## 28.1. batch

### 28.1.1. zadanie : Dodaj masowe encje za pomocą Criteria i JPQL

przykład : SpringJpaKata : JpaBulkTest

### 28.1.2. Zadanie włączenie statystyk

przykład : HibernateKata : StatisticsJPATest

### 28.1.3. Fetch: subselect i batch

przykład : SpringJpaKata : JpaFetchTest

## 29. Problem z Lazy

## 30. n+1 Problem

przykład : SpringJpaKata : N1Test

przykład : SpringJpaKata : JpaPerformanceN1Test2

Standard 6 kwerend :

```
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=1; {executed in 0 msec}
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=2; {executed in 0 msec}
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=3; {executed in 0 msec}
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=4; {executed in 0 msec}
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=5; {executed in 0 msec}
select jobs0_.customerId as customer4_1_0_, jobs0_.jobId as jobId1_1_0_, jobs0_.jobId
as jobId1_1_1_, jobs0_.version as version2_1_1_, jobs0_.name as name3_1_1_ from Job
jobs0_ where jobs0_.customerId=6; {executed in 0 msec}
```

Po @Fetch tuningu:

```

select
    jobs0_.customerId as customer4_1_1_,
    jobs0_.jobId as jobId1_1_1_,
    jobs0_.jobId as jobId1_1_0_,
    jobs0_.version as version2_1_0_,
    jobs0_.name as name3_1_0_
from
    Job jobs0_
where
    jobs0_.customerId in (
        select
            customer0_.customerId
        from
            Customer customer0_

```

Po @BatchSize

```

select
    jobs0_.customerId as customer4_1_1_,
    jobs0_.jobId as jobId1_1_1_,
    jobs0_.jobId as jobId1_1_0_,
    jobs0_.version as version2_1_0_,
    jobs0_.name as name3_1_0_
from
    Job jobs0_
where
    jobs0_.customerId in (
        ?, ?, ?, ?, ?, ?
    )

```

```

select jobs0_.customerId as customer4_1_1_, jobs0_.jobId as jobId1_1_1_, jobs0_.jobId
as jobId1_1_0_, jobs0_.version as version2_1_0_, jobs0_.name as name3_1_0_ from Job
jobs0_ where jobs0_.customerId in (1, 2, 3, 4, 5, 6);

```

przykład : SpringJpaKata : SaveEmTest

## 30.1. zadanie rozwiązanie problemu lazyInitializationException

Przykład : SpringJpaKata : JpaLazyTest

## 30.2. FetchProfile

przykład : SpringJpaKata : JpaFetchTest

## 30.3. Eager problem

### 30.3.1. Cartesian problem

przykład : SpringJpaKata : JpaCartasianTest

## 31. Cache

### 31.1. firstLevelCache test : SpringJpaKata : FirstLevelCacheTest

### 31.2. wyjaśnienie zasady działania

<http://przewidywalna-java.blogspot.com/search?q=hibernate>

### 31.3. zadanie : wprowadz cache do projektu

## 32. Lock

### 32.1. dodać wersjonowanie

przykład : projekt HibernateKata : VersionTest

springData lock : przykład : SpringDataKata : ItemRepository

## 33. DTO

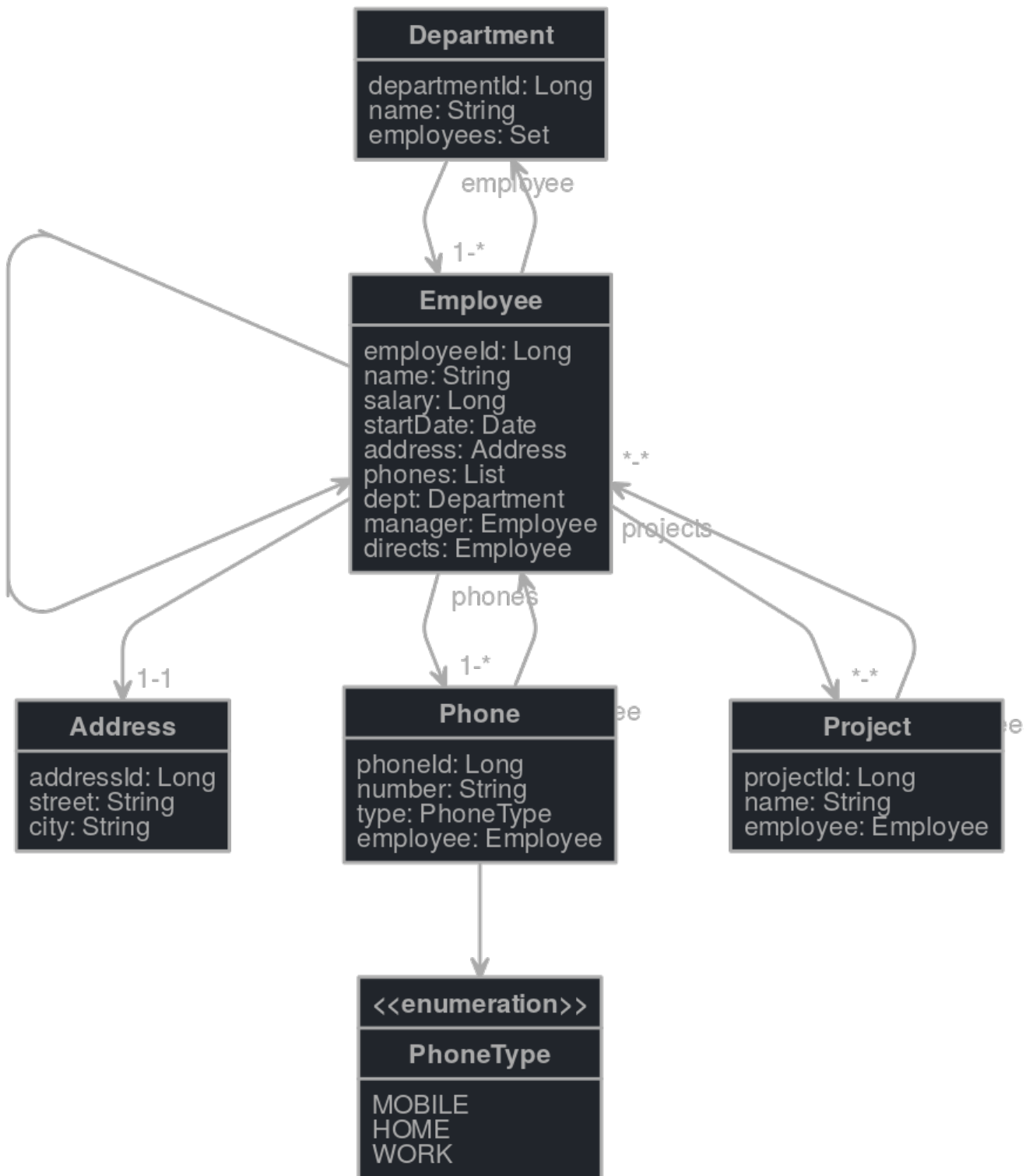
### 33.1. Zadanie wprowadz DTO , spłaszcz model domenowy

## 34. Pułapki i dobre praktyki

przykład : SpringJpaKata : FilterTest

## 35. Podsumowanie :

Pełny model domenowy z relacjami i DAO. Uzupełnić testy



36. Koniec :)