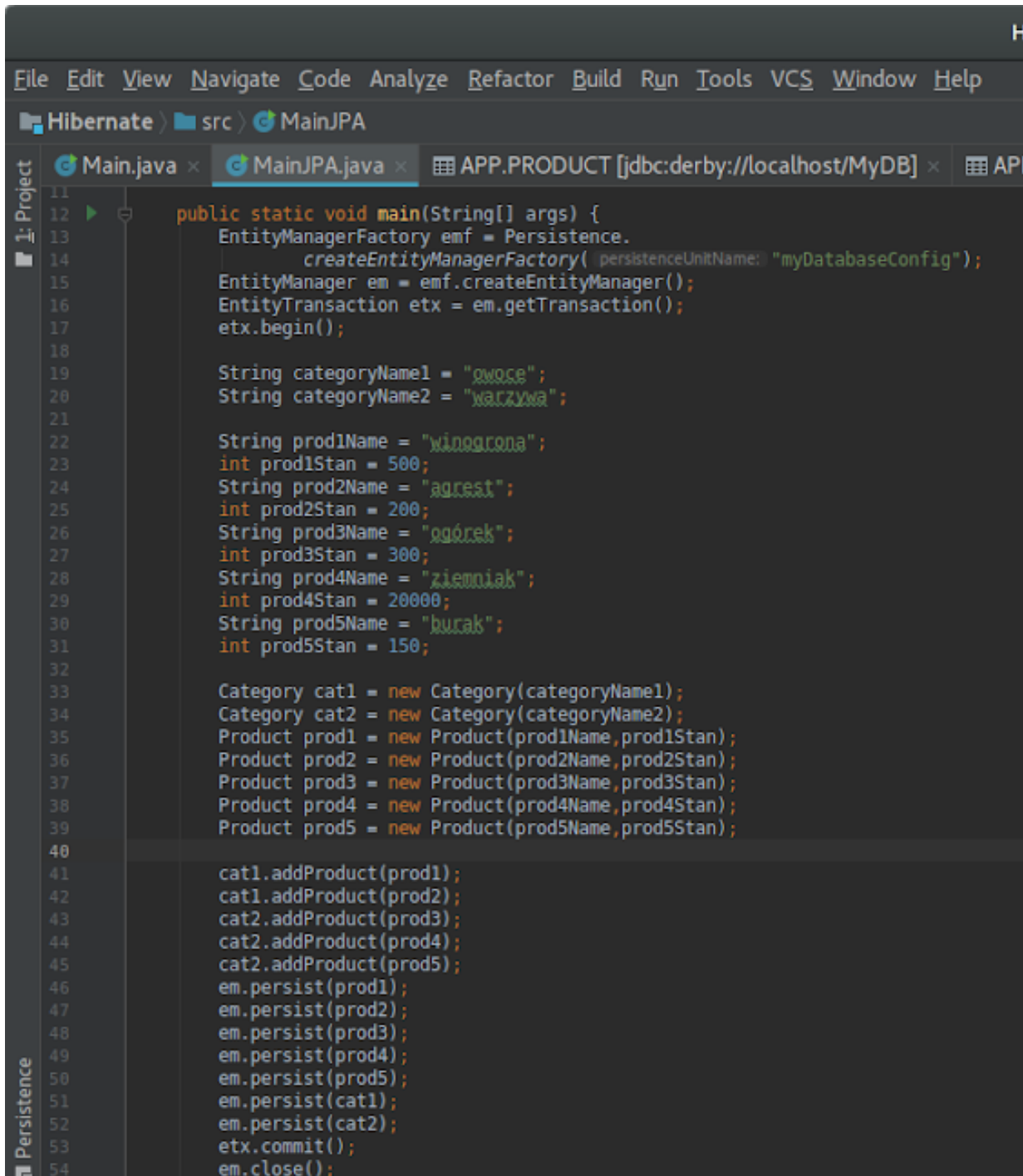


Zadanie domowe z Hibernate'a

Krzysztof Makiela

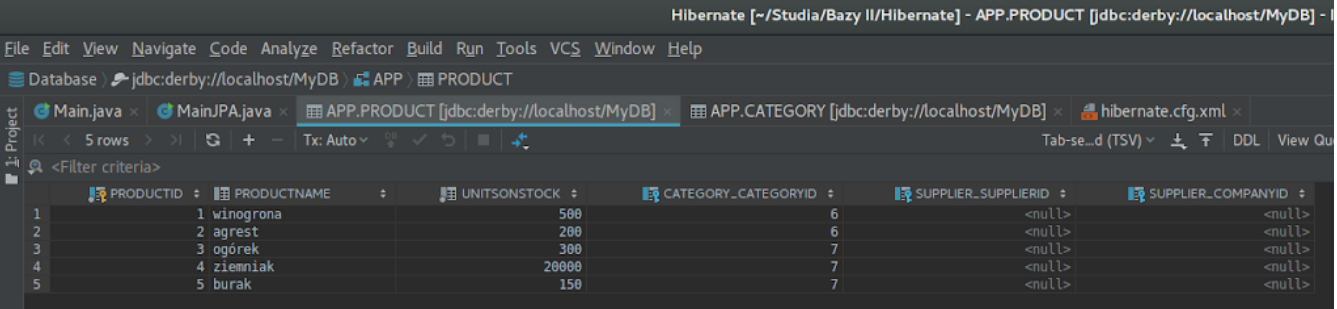
IX

Wykonanie kroku VI przy użyciu JPA:

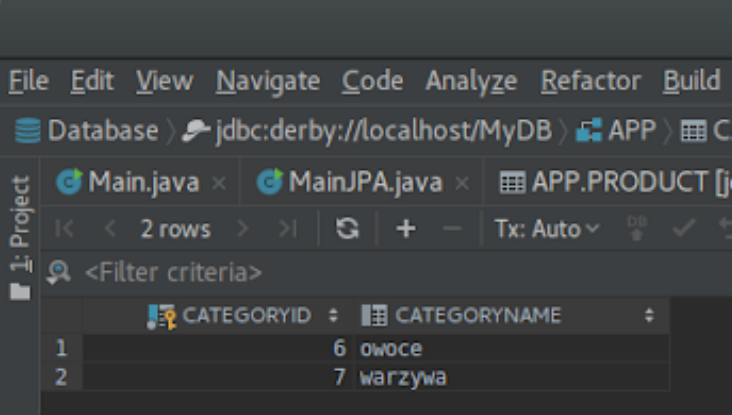


```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Hibernate > src > MainJPA
Main.java x MainJPA.java x APP.PRODUCT [jdbc:derby://localhost/MyDB] x AP
11
12 public static void main(String[] args) {
13     EntityManagerFactory emf = Persistence.
14         createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
15     EntityManager em = emf.createEntityManager();
16     EntityTransaction etx = em.getTransaction();
17     etx.begin();
18
19     String categoryName1 = "owoc";
20     String categoryName2 = "warzywa";
21
22     String prod1Name = "winogrona";
23     int prod1Stan = 500;
24     String prod2Name = "agrest";
25     int prod2Stan = 200;
26     String prod3Name = "ogórek";
27     int prod3Stan = 300;
28     String prod4Name = "ziemniak";
29     int prod4Stan = 20000;
30     String prod5Name = "burak";
31     int prod5Stan = 150;
32
33     Category cat1 = new Category(categoryName1);
34     Category cat2 = new Category(categoryName2);
35     Product prod1 = new Product(prod1Name, prod1Stan);
36     Product prod2 = new Product(prod2Name, prod2Stan);
37     Product prod3 = new Product(prod3Name, prod3Stan);
38     Product prod4 = new Product(prod4Name, prod4Stan);
39     Product prod5 = new Product(prod5Name, prod5Stan);
40
41     cat1.addProduct(prod1);
42     cat1.addProduct(prod2);
43     cat2.addProduct(prod3);
44     cat2.addProduct(prod4);
45     cat2.addProduct(prod5);
46     em.persist(prod1);
47     em.persist(prod2);
48     em.persist(prod3);
49     em.persist(prod4);
50     em.persist(prod5);
51     em.persist(cat1);
52     em.persist(cat2);
53     etx.commit();
54     em.close();
```

Efekt w bazie danych:



PRODUCTID	PRODUCTNAME	UNITSONSTOCK	CATEGORY_CATEGORYID	SUPPLIER_SUPPLIERID	SUPPLIER_COMPANYID
1	winogrona	500	6	<null>	<null>
2	agrest	200	6	<null>	<null>
3	ogórek	300	7	<null>	<null>
4	ziemniak	20000	7	<null>	<null>
5	burak	150	7	<null>	<null>

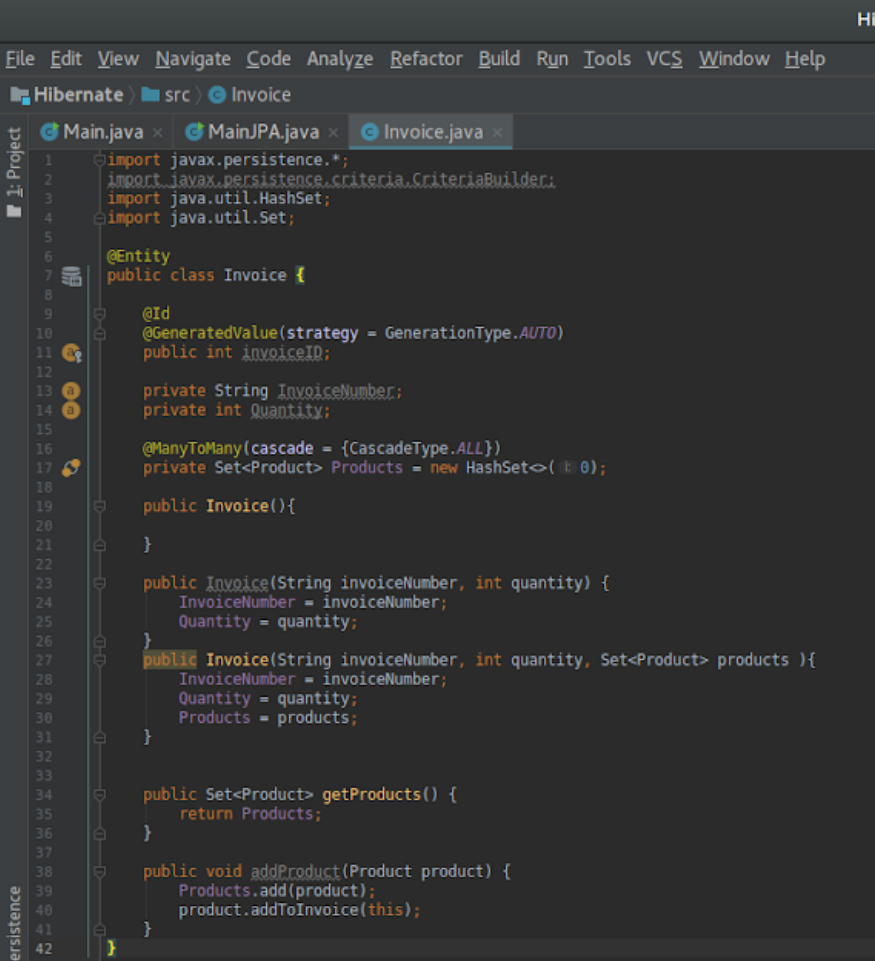


CATEGORYID	CATEGORYNAME
6	owoce
7	warzywa

X

Kaskady

klasa Invoice:



```
1 import javax.persistence.*;
2 import javax.persistence.criteria.Criteria, CriteriaBuilder;
3 import java.util.HashSet;
4 import java.util.Set;
5
6 @Entity
7 public class Invoice {
8
9     @Id
10     @GeneratedValue(strategy = GenerationType.AUTO)
11     public int invoiceID;
12
13     private String invoiceNumber;
14     private int quantity;
15
16     @ManyToMany(cascade = {CascadeType.ALL})
17     private Set<Product> products = new HashSet<>();
18
19     public Invoice(){
20
21     }
22
23     public Invoice(String invoiceNumber, int quantity) {
24         invoiceNumber = invoiceNumber;
25         quantity = quantity;
26     }
27
28     public Invoice(String invoiceNumber, int quantity, Set<Product> products) {
29         invoiceNumber = invoiceNumber;
30         quantity = quantity;
31         products = products;
32     }
33
34     public Set<Product> getProducts() {
35         return products;
36     }
37
38     public void addProduct(Product product) {
39         products.add(product);
40         product.addToInvoice(this);
41     }
42 }
```

W klasie main:

```
53  
54  
55     Product p1 = new Product( ProductName: "prod1", UnitsOnStock: 5000);  
56     Product p2 = new Product( ProductName: "prod2", UnitsOnStock: 3000);  
57     Set<Product> prodset= new HashSet<>( 0);  
58     prodset.add(p1);  
59     prodset.add(p2);  
60     Invoice inv1 = new Invoice( invoiceNumber: "inv1", quantity: 2,prodset);  
61     em.persist(inv1);  
62     etx.commit();  
63     em.close();  
64  
65 }
```

Efekt w bazie danych:

Hibernate [~/Studia/Bazy II/Hibernate] - APP.PRODUCT [jdbc:derby://localhost/MyDB] - IntelliJ

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Database > jdbc:derby://localhost/MyDB > APP > PRODUCT

Main.java x MainJPA.java x APP.INVOICE [jdbc:derby://localhost/MyDB] x APP.PRODUCT [jdbc:derby://localhost/MyDB] x Invoice.java x

< 7 rows > Tx: Auto ✓ DDL View Query

<Filter criteria>

	PRODUCTID	PRODUCTNAME	UNITSONSTOCK	CATEGORY_CATEGORYID	SUPPLIER_SUPPLIERID	SUPPLIER_COMPANYID
1	1	winogrona	500	6	<null>	<null>
2	2	agrest	200	6	<null>	<null>
3	3	ogórek	300	7	<null>	<null>
4	4	ziemniak	20000	7	<null>	<null>
5	5	burak	150	7	<null>	<null>
6	9	prod2	3000	<null>	<null>	<null>
7	10	prod1	5000	<null>	<null>	<null>

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window

Database > jdbc:derby://localhost/MyDB > APP > INVOICE

Main.java x MainJPA.java x APP.INVOICE [jdbc:derby://localhost/MyDB] x

< 1 row > Tx: Auto ✓ DDL View Query

<Filter criteria>

	INVOICEID	INVOICENUMBER	QUANTITY
1	8	inv1	2

File Edit View Navigate Code Analyze Refactor Build Run Tools V

Database > jdbc:derby://localhost/MyDB > APP > INVOICE_PROD

Main.java x MainJPA.java x APP.INVOICE [jdbc:derby://local

< 2 rows > Tx: Auto ✓ DDL View Query

<Filter criteria>

	INVOICES_INVOICEID	PRODUCTS_PRODUCTID
1	8	9
2	8	10

XI

Embedded

Klasa Address:

```
Address.java [Hibernate] - IntelliJ IDEA

DUCT [jdbc:derby://localhost/MyDB] x Address.java x

1  import javax.persistence.Column;
2  import javax.persistence.Embeddable;
3
4  @Embeddable
5  public class Address {
6
7      @Column(name = "street")
8      private String StreetA;
9      @Column(name = "city")
10     private String CityA;
11
12
13     public Address(){
14
15     }
16
17     public Address(String streetA, String cityA){
18         this.StreetA = streetA;
19         this.CityA = cityA;
20     }
21
22     public String getStreetA() {
23         return StreetA;
24     }
25
26     public void setStreetA(String streetA) {
27         StreetA = streetA;
28     }
29
30     public String getCityA() {
31         return CityA;
32     }
33
34     public void setCityA(String cityA) {
35         CityA = cityA;
36     }
37 }
```

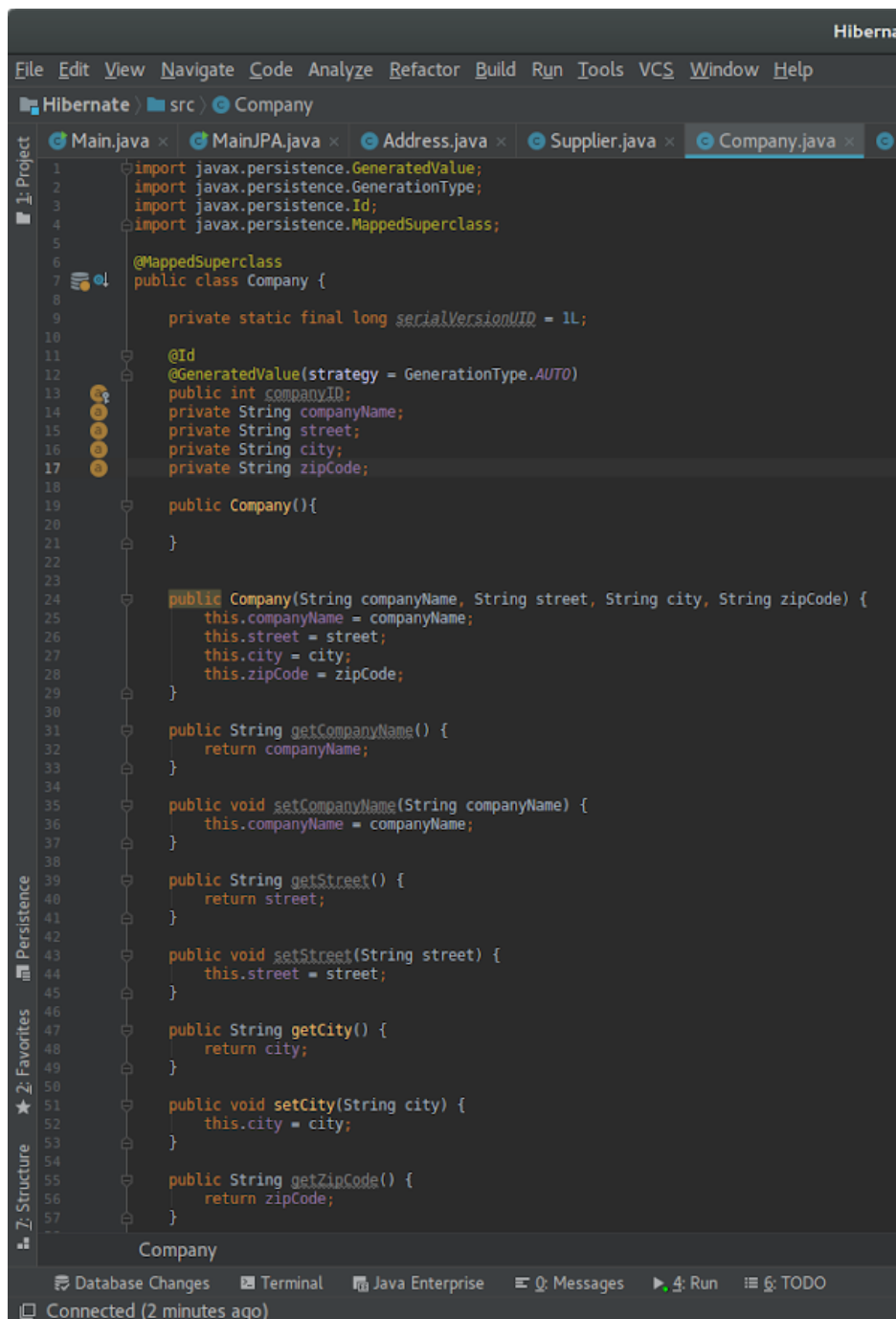
Klasa Supplier:

```
15
16     @Embedded
17     private Address address;
18
19     @Embedded
20     @AttributeOverrides({
21         @AttributeOverride(name = "CityA", column = @Column(name = "cityD")),
22         @AttributeOverride(name = "StreetA", column = @Column(name = "streetD"))
23     })
24     private Address delivery_address;
25
26
27
```

XII

Dziedziczenie

Klasa Company:



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Hibernate | src | Company
Main.java x MainJPA.java x Address.java x Supplier.java x Company.java x
1 import javax.persistence.GeneratedValue;
2 import javax.persistence.GenerationType;
3 import javax.persistence.Id;
4 import javax.persistence.MappedSuperclass;
5
6 @MappedSuperclass
7 public class Company {
8
9     private static final long serialVersionUID = 1L;
10
11     @Id
12     @GeneratedValue(strategy = GenerationType.AUTO)
13     public int companyID;
14     private String companyName;
15     private String street;
16     private String city;
17     private String zipCode;
18
19     public Company(){
20
21     }
22
23
24     public Company(String companyName, String street, String city, String zipCode) {
25         this.companyName = companyName;
26         this.street = street;
27         this.city = city;
28         this.zipCode = zipCode;
29     }
30
31     public String getCompanyName() {
32         return companyName;
33     }
34
35     public void setCompanyName(String companyName) {
36         this.companyName = companyName;
37     }
38
39     public String getStreet() {
40         return street;
41     }
42
43     public void setStreet(String street) {
44         this.street = street;
45     }
46
47     public String getCity() {
48         return city;
49     }
50
51     public void setCity(String city) {
52         this.city = city;
53     }
54
55     public String getZipCode() {
56         return zipCode;
57     }
58
59 }
Company
Database Changes Terminal Java Enterprise Messages Run TODO
Connected (2 minutes ago)
```

Klasa Supplier:

```
3
4 @Entity
5 public class Supplier extends Company implements Serializable {
6
7     private static final long serialVersionUID = 1L;
8
9     @Id()
10    @GeneratedValue(strategy = GenerationType.AUTO)
11    public int supplierID;
12    private String bankAccountNumber;
13
14    public Supplier(){
15    }
16
17    public Supplier(String companyName, String street, String city, String zipCode, String bankAccountNumber) {
18        super(companyName, street, city, zipCode);
19        this.bankAccountNumber = bankAccountNumber;
20    }
21
22    public String getBankAccountNumber() {
23        return bankAccountNumber;
24    }
25
26    public void setBankAccountNumber(String bankAccountNumber) {
27        this.bankAccountNumber = bankAccountNumber;
28    }
29
30 }
```

Klasa Customer:

```
6
7 @Entity
8 public class Customer extends Company implements Serializable {
9
10    private static final long serialVersionUID = 1L;
11
12    @Id
13    @GeneratedValue(strategy = GenerationType.AUTO)
14    public int customerID;
15    public double discount;
16
17    public Customer(){
18    }
19
20    public Customer(String companyName, String street, String city, String zipCode, float discount) {
21        super(companyName, street, city, zipCode);
22        this.discount = discount;
23    }
24
25    public double getDiscount() {
26        return discount;
27    }
28
29    public void setDiscount(float discount) {
30        this.discount = discount;
31    }
32
33
34
35 }
```

Wywołania w main:

```
63
64 Customer cust1 = new Customer( companyName: "Customer1", street: "customerStreet", city: "customerCity", zipCode: "666-666", discount: 15);
65 Supplier suppl1 = new Supplier( companyName: "supplier1", street: "supplierstreet", city: "suppcity", zipCode: "66-666-66", bankAccountNumber: "12345676853256856");
66 em.persist(cust1);
67 em.persist(suppl1);
68
```

Efekt w bazie:

<Filter criteria>							
	SUPPLIERID	COMPANYID	CITY	COMPANYNAME	STREET	ZIPCODE	BANKACCOUNTNUMBER
1	13	14	suppcity	supplier1	supplierstreet	66-666-66	12345676853256856

<Filter criteria>							
	CUSTOMERID	COMPANYID	CITY	COMPANYNAME	STREET	ZIPCODE	DISCOUNT
1	11	12	customerCity	Customer1	customerStreet	666-666	15

