

Budowa aplikacji wielowarstwowych. Zastosowanie technologii Ajax

Laboratorium 8
Technologie internetowe
Zofia Kruczkiewicz

Wykaz pytań dotyczących materiału wykorzystanego w lab8, które należy opracować (m.in. wykłady: 8).

1. Należy opisać rolę znacznika

<f:ajax execute="cena promocja" render="brutto"/>

zagnieżdżonego w znaczniku **<h:commandButton** . Co oznaczają i jaką rolę pełnią atrybuty **execute** i **render**?

2. Należy opisać rolę znacznika

<f:ajax event="blur" execute="nazwa" render="messagePanel"/>

zagnieżdżonego w znaczniku **<h:inputText** . Co oznaczają i jaką pełnią rolę atrybuty **event**, **execute** oraz **render**?

3. Należy opisać rolę znacznika

<f:ajax execute="cena promocja" render="brutto"/>

zagnieżdżonego w znaczniku **<h:selectOneMenu** . Co oznaczają i jaką pełnią rolę atrybuty **event**, **execute** oraz **render**?

Zastosowanie znacznika f:ajax w celu poprawy wydajności programu

Czynności początkowe przed wykonaniem p. 1.
Wykonany program jest przeznaczony do rozwijania wg p.1-6.

- Należy wykonać kopię programu, wykonanego podczas lab6, jako **Sklep_7_Ajax** (wg **wskazówek z instrukcji do lab. 2, slajd 4:** <http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/ti /LAB TINT 2.pdf>). Nazwa projektu w tej instrukcji nie pokrywa się z proponowaną nazwą projektu.
- Ustawić kodowanie UTF-8; po zaznaczeniu nazwy projektu w oknie **Projects** prawym klawiszem myszy wybrać pozycję **Properties/Sources/Encoding/UTF-8**

1. Wyświetlanie ceny brutto przed przekazaniem wprowadzanych danych do przetwarzania

1.1. Należy dodać przycisk do strony `dodaj_produk2.xhtml`, który po naciśnięciu podaje aktualną cenę brutto wynikającą z podanej promocji i ceny netto. Cena brutto jest wyświetlana na stronie `dodaj_produk2.xhtml`, ale bez przeładowania tej strony.

W tym celu należy dodać jako ostatni fragment w znaczniku `panelGrid` strony znaczniki: `h:outputText` oraz `h:CommandButton` z zagnieżdżonym znacznikiem `f:ajax`, który po kliknięciu na przycisk powoduje wysłanie wartości znaczników `h:inputText` o `id="cena"` i `id="promocja"` na serwer, gdzie zostanie obliczona cena brutto za pomocą metody `getCena_brutto` w komponencie `managed_produk` i wysłanie jej do przeglądarki w znaczniku `h:outputText` o `id="brutto"`.

```
<h:panelGrid columns="2" >
```

```
.....
```

```
    <h:commandButton value="#{bundle['lista_produkow.cena_brutto']}">
```

```
        <f:ajax execute="cena promocja" render="brutto"/>
```

```
    </h:commandButton>
```

```
    <h:outputText id="brutto" value="#{managed_produk.cena_brutto_}" >
```

```
        <f:convertNumber currencySymbol="z&#322;" type="currency"/>
```

```
    </h:outputText>
```

```
</h:panelGrid>
```

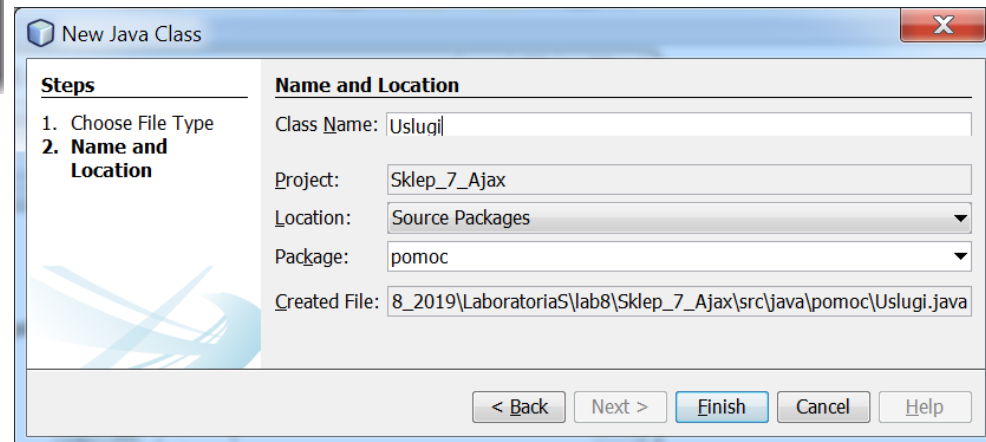
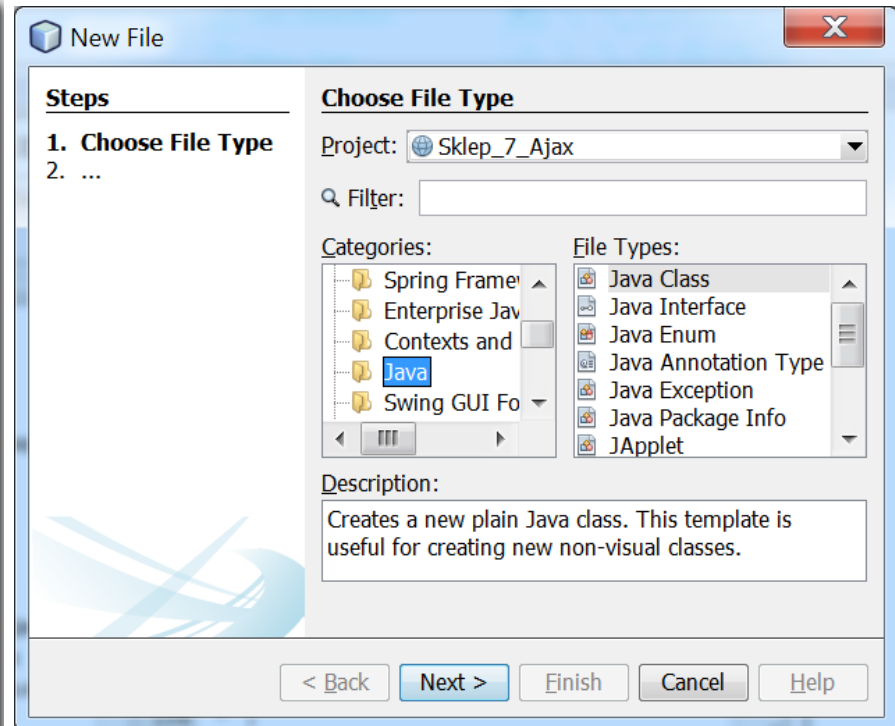
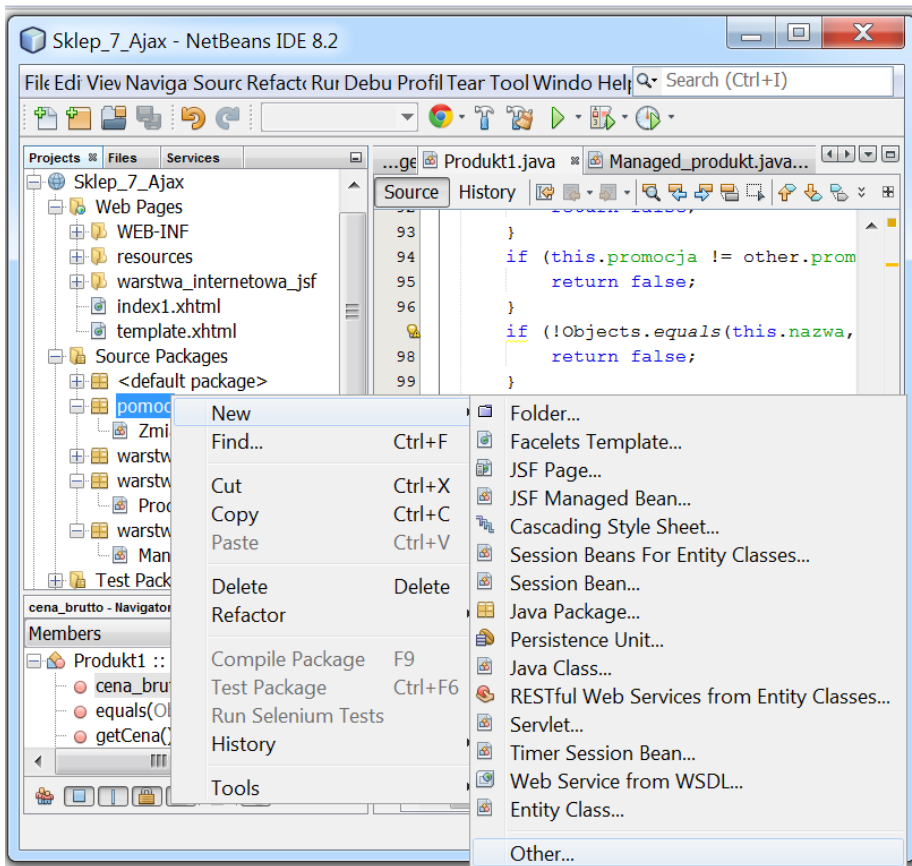
W atrybucie `execute` znacznika `f:ajax` znajduje się łańcuch nazw jako `id` poszczególnych znaczników `h:inputText`, służących do wprowadzania danych na stronie `dodaj_produk2.xhtml`:

```
<h:inputText id="cena"
```

```
<h:inputText id="promocja"
```

Aby ten kolejny element strony działał należy wykonać następujące czynności, podane dalej.

1.2. Należy zdefiniować w pakiecie **pomoc** projektu klasę o nazwie **Usługi**



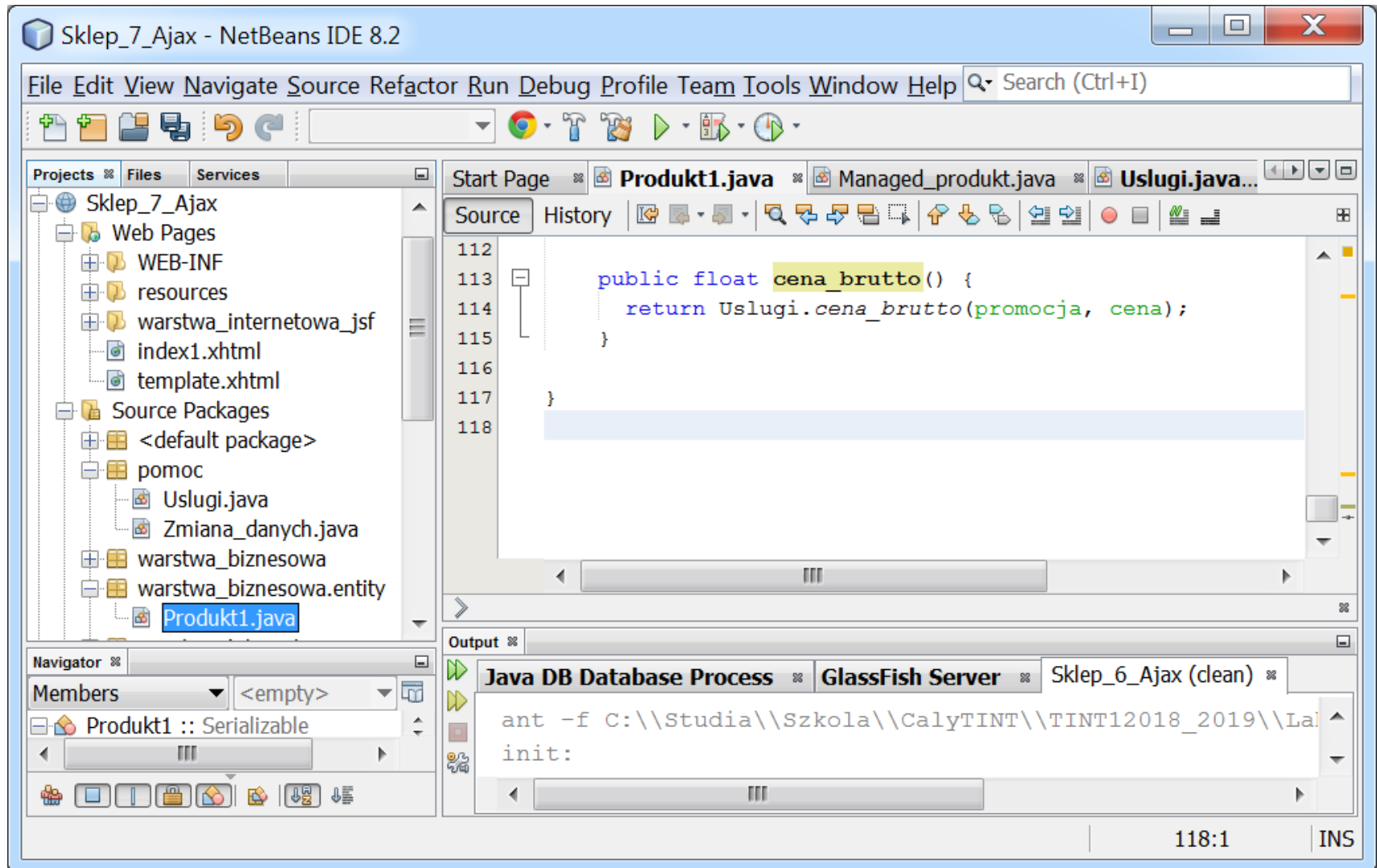
1.3. Zdefiniowanie uniwersalnej metody `cena_brutto` w klasie `Uslugi`

The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows the source code for the `Uslugi` class in the `Uslugi.java` file. The code defines a public static method `cena_brutto` that calculates the gross price based on a discount and a net price.

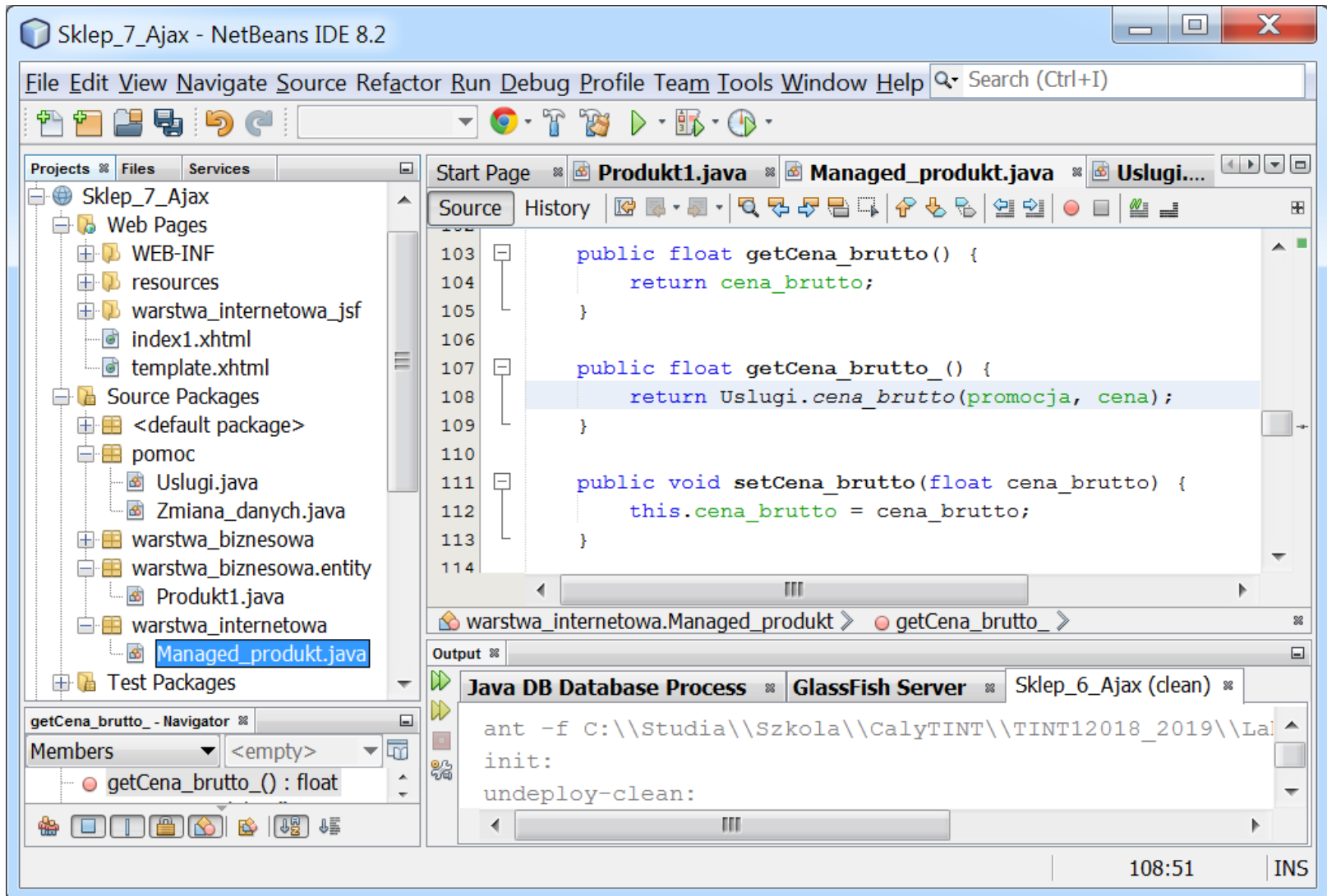
```
1 package pomoc;  
2  
3  
4 public class Uslugi {  
5  
6     public static float cena_brutto(int promocja, float cena)  
7     {  
8         return cena * (1 - (float) promocja / 100);  
9     }  
10 }  
11  
12  
13
```

The left sidebar shows the project structure for `Sklep_7_Ajax`, with the `warstwa_internetowa` package selected. The bottom status bar indicates the current zoom level is 12:1 and the cursor is in the Insert (INS) mode.

1.4. Przedefiniowanie metody `cena_brutto` w klasie `Produkt1` (wprowadzenie wieloużywalności kodu)



1.5. Dodanie metody `getCena_brutto_` w klasie `Managed_produk`t (wprowadzenie wielouzywalności kodu), wywołanej w atrybucie **value**:
<h:outputText id="brutto" **value="#{managed_produk.t.cena_brutto_}"** > (str. 5)



1.6. Wynik działania znacznika `<f:ajax execute="cena promocja" render="brutto"/>`

Uwaga: w celu uzyskania wyniku należy kliknąć na przycisk **Cena brutto**

Wstawianie nowego produktu x Stan przed naciśnięciem przycisku **Cena brutto**

localhost:14308/

Dodaj produkt
Lista produktów

Podaj nazwę produktu Produkt1

Podaj cenę netto produktu 100 zł

Podaj promocję produktu 10

Podaj datę produkcji 5-11-2019

Cena brutto 0 zł

OK

Bottom

Wstawianie nowego produktu x Stan po naciśnięciu przycisku **Cena brutto**

localhost:14308/

Dodaj produkt
Lista produktów

Podaj nazwę produktu

Podaj cenę netto produktu 0 zł

Podaj promocję produktu 0

Podaj datę produkcji

Cena brutto 0 zł

OK

Bottom

Wstawianie nowego produktu x Stan po naciśnięciu przycisku **Cena brutto**

localhost:14308/

Dodaj produkt
Lista produktów

Podaj nazwę produktu Produkt1

Podaj cenę netto produktu 100 zł

Podaj promocję produktu 0

Podaj datę produkcji

Cena brutto 100 zł

OK

Bottom

Wstawianie nowego produktu x Stan po naciśnięciu przycisku **Cena brutto**

localhost:14308/

Dodaj produkt
Lista produktów

Podaj nazwę produktu Produkt1

Podaj cenę netto produktu 100 zł

Podaj promocję produktu 10

Podaj datę produkcji 5-11-2019

Cena brutto 90 zł

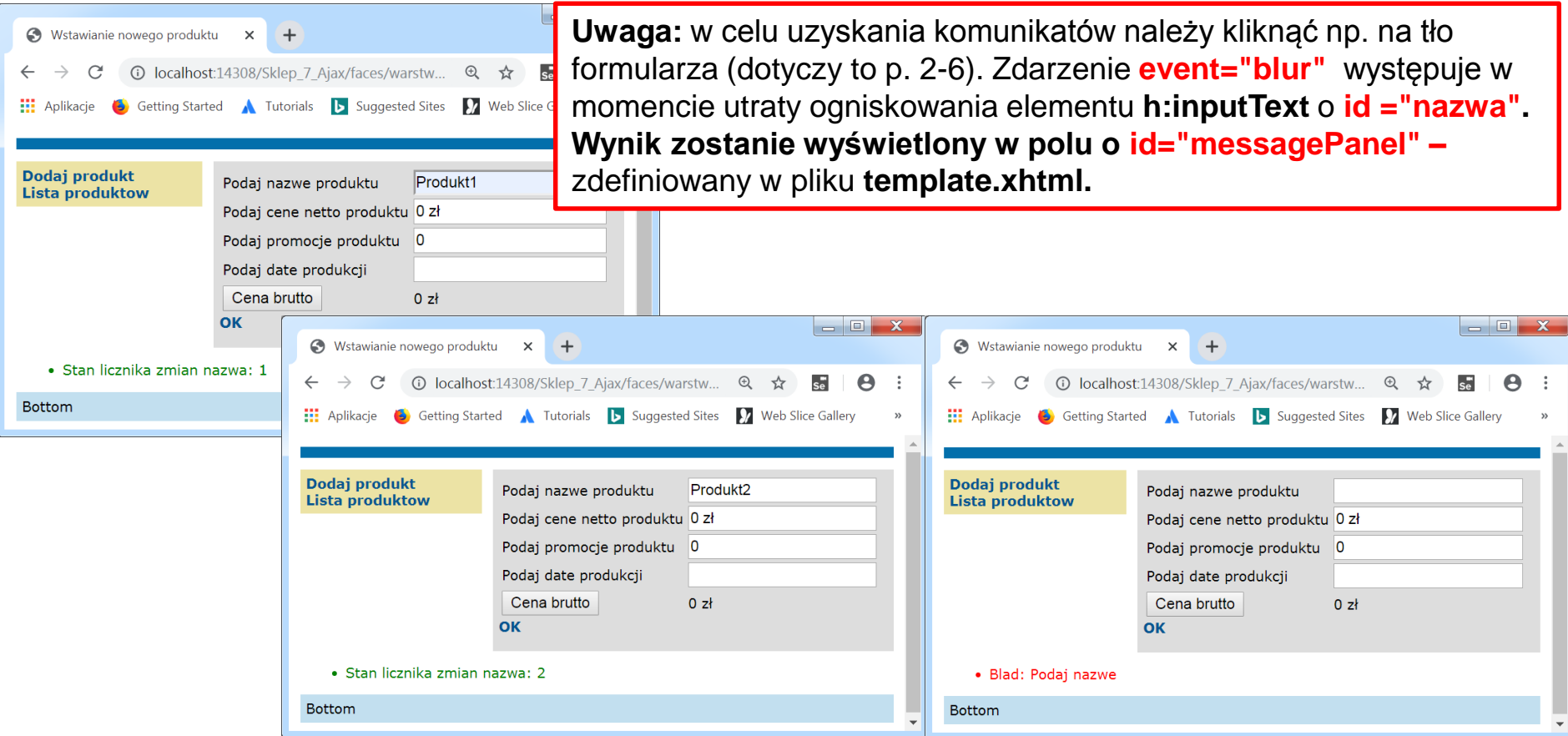
OK

Bottom

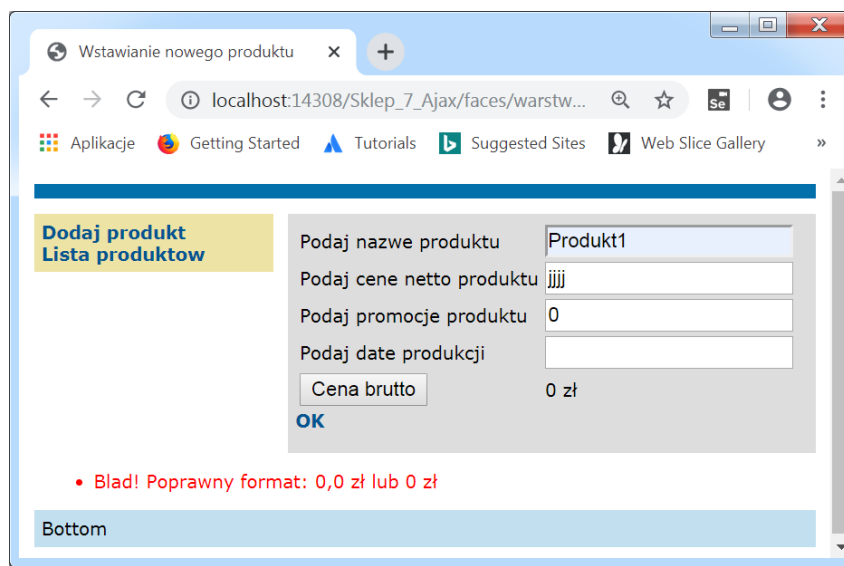
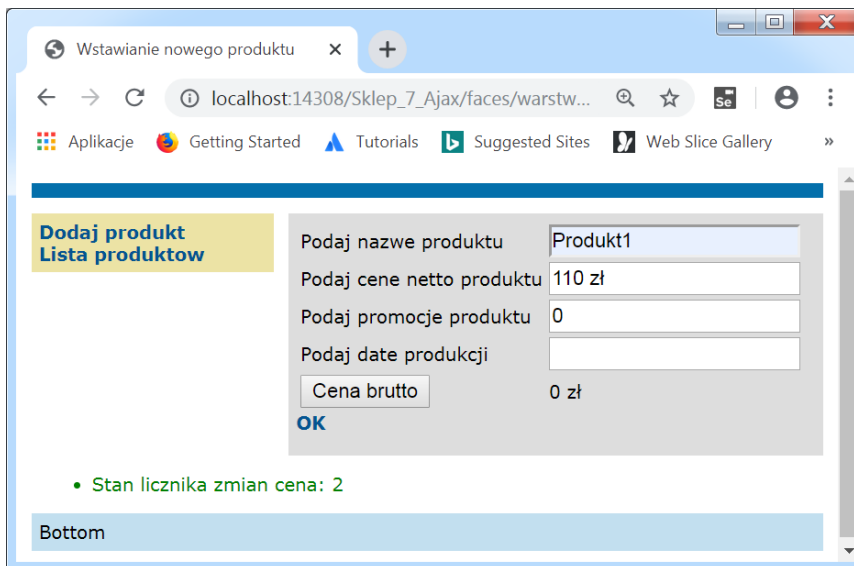
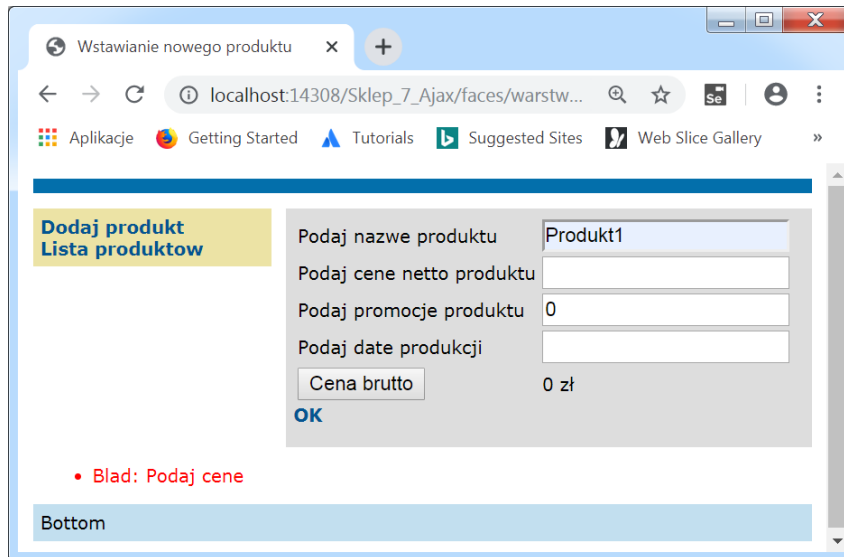
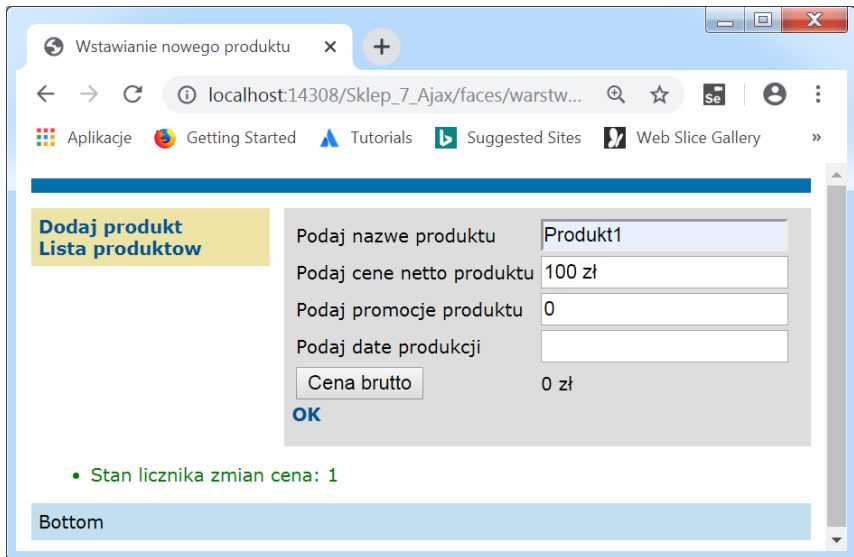
2. Wyświetlanie informacji o **liczbie zmian lub braku wprowadzonych danych** w polu wejściowym o etykiecie **Podaj nazwe produktu** na stronie **dodaj_produk2.xhtml** bez przeładowania całej strony (za pomocą znacznika **f:ajax**) – należy zmodyfikować kod **h:inputText** podanej poniżej.

```
<h:outputLabel value="#{bundle['dodaj_produk2.nazwa']}" for="nazwa" />
<h:inputText id="nazwa" title="#{bundle['dodaj_produk2.nazwa1']}"
value="#{managed_produk2.nazwa}"
required="true" requiredMessage="#{bundle['dodaj_produk2.blad_nazwa']}" >
<f:valueChangeListener binding="#{managed_produk2.zmiana1}"/>
<f:ajax event="blur" execute="nazwa" render="messagePanel"/>
</h:inputText>
```

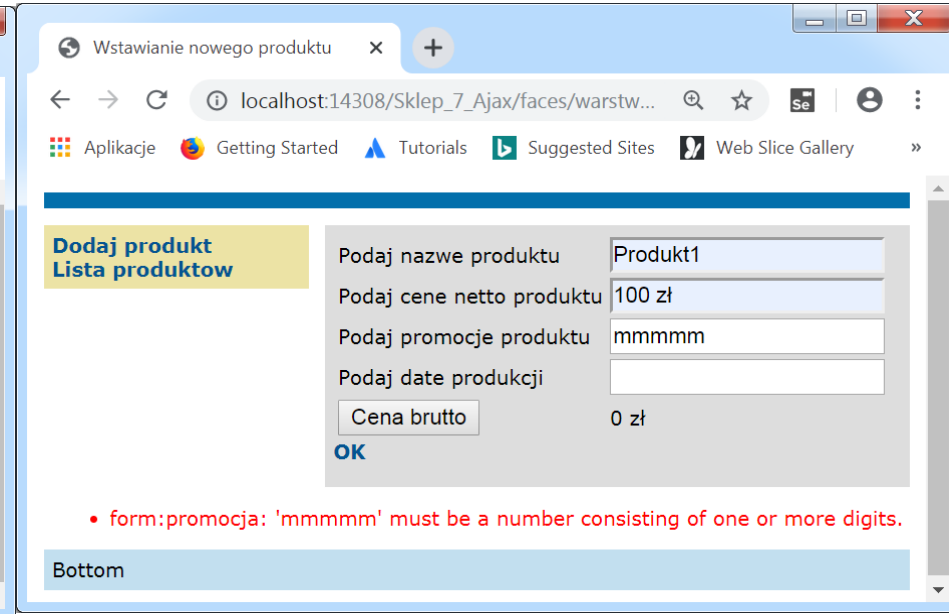
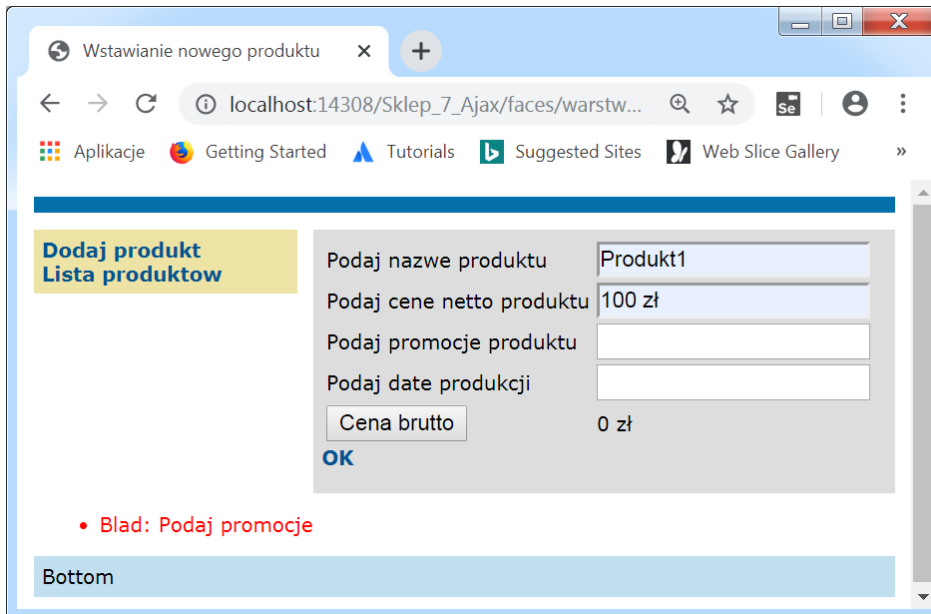
Uwaga: w celu uzyskania komunikatów należy kliknąć np. na tło formularza (dotyczy to p. 2-6). Zdarzenie **event="blur"** występuje w momencie utraty ogniskowania elementu **h:inputText** o **id="nazwa"**. Wynik zostanie wyświetlony w polu o **id="messagePanel"** – zdefiniowany w pliku **template.xhtml**.



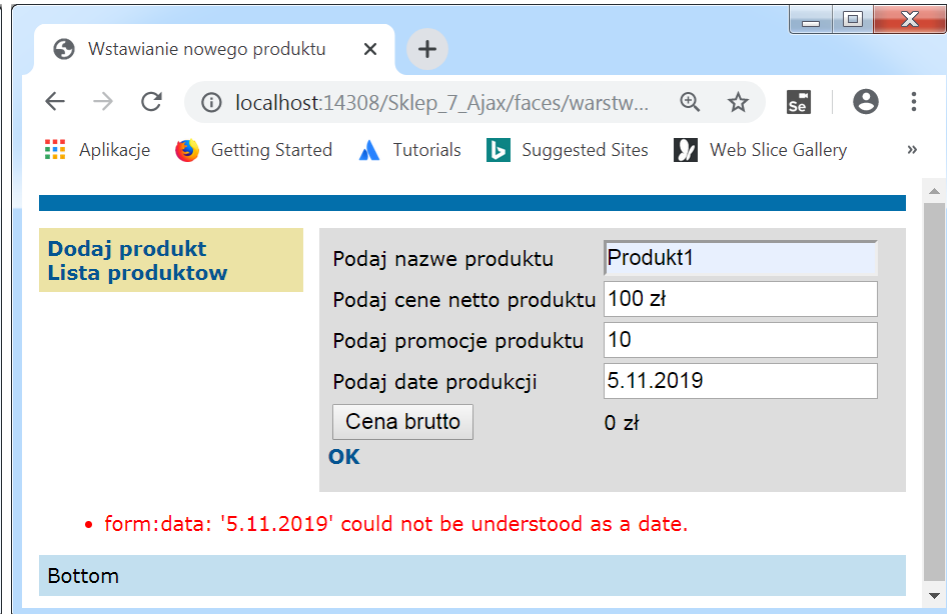
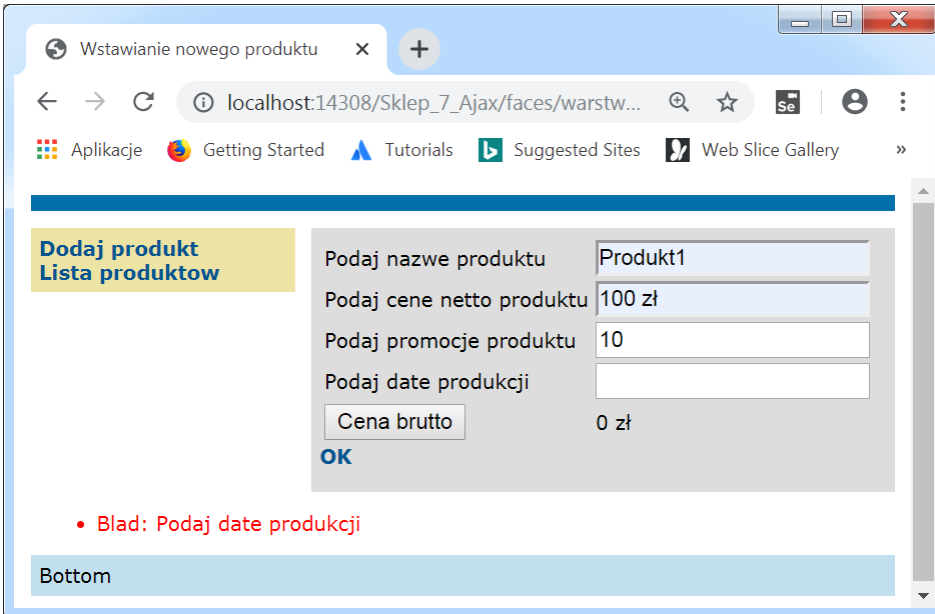
3. Należy wprowadzić zmiany do kodu strony dodaj_produk2.xhtml w celu wyświetlania informacji o **liczbie zmian, braku wprowadzonych danych oraz błędy formatu danych** w polu wejściowym o etykiecie **Podaj cene netto produktu** na stronie **podaj_produk2.xhtml** bez przeładowania strony za pomocą znacznika znacznika f:ajax (przykład z p.2).



4. Należy wprowadzić zmiany do kodu strony **dodaj_produk2.xhtml** w celu wyświetlania informacji o **braku wprowadzonych danych oraz błędu formatu** w polu wejściowym o etykiecie **Podaj promocje produktu** na stronie **podaj_produk2.xhtml** bez przeładowania strony za pomocą znacznika **f:ajax** (przykład z p. 2).



5. Należy wprowadzić zmiany do kodu strony **dodaj_produk2.xhtml** w celu wyświetlania informacji o **błędach formatu daty lub braku wprowadzonych danych daty** w polu wejściowym o etykiecie **Podaj date produktu** na stronie **podaj_produk2.xhtml** bez przeładowania strony za pomocą znacznika **f:ajax** (przykład z p.2)



6. Należy wprowadzić zmiany do kodu strony **dodaj_produk2.html** w celu wyświetlania informacji o **częstotliwości prowadzonych danych, braku danych** w polu wejściowym dodanym w ramach poprzednich laboratoriów bez przeładowania strony za pomocą znacznika f:ajax (przykład z p. 2)

Czynności początkowe przed wykonaniem p. 7

- Należy wykonać kopię programu, wykonanego podczas lab7, jako **Sklep_8_Ajax** (wg wskazówek z instrukcji do lab. 2, slajd 4: http://zofia.kruckiewicz.staff.iiar.pwr.wroc.pl/wyklady/ti_/LAB_TINT_2.pdf). Nazwa projektu w tej instrukcji nie pokrywa się z proponowaną nazwą projektu.
- Ustawić kodowanie UTF-8; po zaznaczeniu nazwy projektu w oknie **Projects** prawym klawiszem myszy wybrać pozycję **Properties/Sources/Encoding/UTF-8**

7. Należy wprowadzić zmiany w programie **Sklep_8_Ajax** wykonanym jako kopia **Sklep_8** zrealizowanego w ramach **lab7** – należy dokonać zmiany podane poniżej w pliku **dodaj_produkt2.xhtml**

7.1. W znaczniku wyboru promocji dodano znacznik **f:ajax** w celu wybrania wartości wprowadzonych w znacznikach **h:inputText** o **id="cena"** i **id="promocja"** i wyświetlenia wartości ceny brutto w komponencie **h:outputText** o **id="brutto"** **bez konieczności ponownego załadowania całej strony.**

```
<h:panelGrid columns="2">
```

```
.....  
<h:outputLabel value="#{bundle['dodaj_produkt2.promocja']}" for="promocja" />
```

```
<h:selectOneMenu
```

```
    id="promocja" title="#{bundle['dodaj_produkt2.promocja1']}"
```

```
    value="#{managed_produkt.promocja}"
```

```
    required="true" requiredMessage="#{bundle['dodaj_produkt2.blad_promocja']}" >
```

```
    <f:selectItems value="#{managed_produkt.itemsAvailableSelectOne}"/>
```

```
    <f:ajax execute="cena promocja" render="brutto"/>
```

```
</h:selectOneMenu>
```

```
.....  
<h:outputLabel for="brutto" value="#{bundle['lista_produktow.cena_brutto']}" />
```

```
<h:outputText id="brutto" value="#{managed_produkt.cena_brutto_}" >
```

```
    <f:convertNumber currencySymbol="zł;" type="currency"/>
```

```
</h:outputText>
```

```
</h:panelGrid>
```

7.2. Należy skopiować plik z definicją klasy **Uslugi.java** do pakietu **pomoc** (z pakietu pomoc z programu **Sklep_7_Ajax**) i następnie wykonać zmiany wg p. **1.4, 1.5**:

Klasa **Uslugi**

```
package pomoc;  
public class Uslugi {  
    public static float cena_brutto(int promocja, float cena) {  
        return cena * (1 - (float) promocja / 100);  
    }  
}
```

Nowa metoda w klasie **Managed_produk**t:

```
public float getCena_brutto_() {  
    return Uslugi.cena_brutto(promocja, cena);  
}
```

Zmiana metody w klasie **Produkt1**:

```
public float cena_brutto() {  
    return Uslugi.cena_brutto(promocja, cena);  
}
```

7.3. Wynik zmiany zawartości strony **dodaj_produk2.xhtml** bez konieczności przeładowania całej strony – zmiana jedynie pola z etykietą **Cena brutto**

Uwaga: w celu uzyskania wyniku należy wybrać promocję z listy rozwijanej

The screenshots illustrate the following sequence of events:

- Screenshot 1:** The form is initially empty. The 'Cena brutto' field is set to 0 zł.
- Screenshot 2:** The 'Podaj nazwę produktu' field is filled with 'Produkt1'.
- Screenshot 3:** The 'Podaj cenę netto produktu' field is filled with '100 zł'.
- Screenshot 4:** The 'Podaj promocję produktu' dropdown menu is open, showing options 0, ---, 0, 10, 20, and 50. The '10' option is selected, and the 'Cena brutto' field is updated to 90 zł.
- Screenshot 5:** The 'Podaj promocję produktu' dropdown menu is closed, and the 'Cena brutto' field is updated to 80 zł.