

## PrimeFaces

Celem ćwiczenia jest przygotowanie aplikacji JSF przy użyciu biblioteki komponentów PrimeFaces. Do wykonania ćwiczenia potrzebne jest środowisko NetBeans w wersji 13.

### Ćwiczenie 1

1. Otwórz środowisko NetBeans.
2. Utwórz nowy projekt typu Web Application z kategorii Java with Maven. Kliknij przycisk **Next >**. Podaj własną nazwę projektu, wyczyść pole z nazwą pakietu i kliknij przycisk **Next >**. Wybierz wersję Java EE Jakarta EE 8 Web i upewnij się, że jako serwer aplikacji wybrany jest serwer Payara (jeśli nie jest dostępny do wyboru, to kliknij **Add...** i pobierz oraz zainstaluj najnowszą dostępną wersję). Kliknij przycisk **Finish**.
3. Otwórz do edycji plik konfiguracyjny projektu Mavena (pom.xml). Odszukaj w nim sekcję <dependencies> zawierającą odwołanie do biblioteki Java EE. Wstaw poniższy zapis dodający do zależności projektu bibliotekę PrimeFaces:

```
<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>primefaces</artifactId>
  <version>11.0.0</version>
</dependency>
```

4. Odszukaj w pliku pom.xml wersję maven-war-plugin. Jeśli podana jest wcześniejsza niż 3.3.2, to zmień ją na 3.3.2. Zapisz zmiany.
5. Usuń z projektu (z gałęzi Web Pages) utworzoną przez kreator projektu stronę index.html.
6. Utwórz w projekcie nową stronę JSF (New File→JavaServer Faces/JSF Page). Jako jej nazwę podaj index. Zwróć uwagę aby powstała strona ze składnią Facelets (a nie JSP). Rozszerzenie pliku strony.xhtml zostanie dodane automatycznie.
7. W ciele (h:body) strony index.xhtml umieść formularz JSF z komponentem Spinner z biblioteki PrimeFaces wklejając poniższy kod:

```
<h:form>
  <p:spinner id="minMax" min="0" max="42" />
</h:form>
```

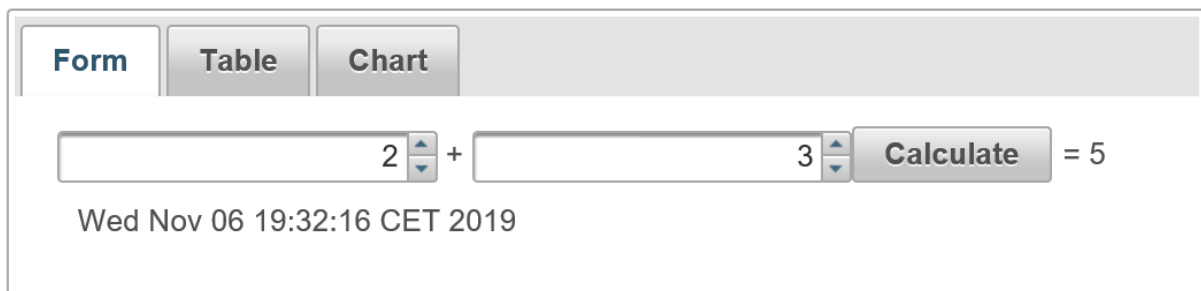
8. Aby znaczniki reprezentujące komponenty PrimeFaces były prawidłowo zinterpretowane, dodaj w elemencie <html> następującą deklarację przestrzeni nazw ze zwyczajowym dla biblioteki PrimeFaces prefiksem p:

```
xmlns:p="http://primefaces.org/ui"
```

9. Przebuduj projekt. Biblioteka PrimeFaces powinna zostać pobrana z repozytorium Mavena zgodnie z zapisem z pliku pom.xml i NetBeans powinien przestać zgłaszać błędy w kodzie strony.
10. Uruchom stronę i przetestuj działanie komponentu Spinner w przeglądarce.

## Ćwiczenie 2

1. Zmodyfikuj stronę `index.xhtml` tak aby zawierała komponent układu `TabView` z trzema zakładkami: „Form”, „Table”, „Chart”.
2. Na pierwszej zakładce zbuduj formularz do dodawania dwóch liczb całkowitych. Do wprowadzania liczb wykorzystaj komponenty `Spinner`. Zwróć uwagę by przycisk (`CommandButton`) był z biblioteki `PrimeFaces`, a nie ze zbioru standardowych komponentów JSF. Uwaga: Przycisk powinien mieć ustawiony atrybut `update`, wskazujący na komponent `OutputText` prezentujący wynik dodawania. Operacja dodawania powinna być realizowana przez komponent `managed bean`.
3. Umieść po formularzem dodatkowy komponent `OutputText` prezentujący bieżącą datę. Data powinna być pobierana z komponentu `managed bean`.



The screenshot shows a web application with a `TabView` component containing three tabs: "Form", "Table", and "Chart". The "Form" tab is selected. Inside the "Form" tab, there is a calculation interface. It consists of two `Spinner` input fields. The first field contains the value "2" and the second field contains the value "3". Between the fields is a plus sign "+". To the right of the second field is a `CommandButton` labeled "Calculate". To the right of the button is an equals sign "=" followed by the result "5". Below the input fields and the button, there is an `OutputText` component displaying the current date and time: "Wed Nov 06 19:32:16 CET 2019".

4. Sprawdź czy data się odświeża po kliknięciu przycisku realizującego dodawanie. O czym to świadczy?
5. Obejrzyj źródło strony w przeglądarce zwracając uwagę na kod JavaScript i związane z nim biblioteki.
6. Ustaw dla przycisku atrybut `ajax="false"` i ponownie przetestuj działanie formularza.
7. Przywróć dla przycisku domyślnie ustawienie `ajax="true"`. Dopisz do wartości atrybutu `update` przycisku (po spacji) `id` komponentu prezentującego datę. Czy teraz też data się odświeża przy dodawaniu?
8. Dodaj na stronie komponent `Growl`. Spraw by po każdej operacji dodawania był w nim wyświetlany komunikat pokazujący wykonane działanie (np. „2 + 3 = 5”).

## Ćwiczenie 3

1. Na zakładce „Table” umieść tabelkę (`DataTable`) prezentującą dane o studentach (imię, nazwisko, średnia). Dane te powinny być generowane programowo przez komponent `managed bean`. Przetestuj działanie aplikacji.
2. Dodaj możliwość sortowania danych w tabelce wg nazwiska i średniej. Przetestuj działanie aplikacji.
3. Rozszerz tabelkę o mechanizm paginacji.

## Ćwiczenie 4

1. Na zakładce „Chart” dodaj wykres liniowy (`LineChart`) prezentujący wartości sinusa i cosinusa kątów od 0 do 360 stopni (z krokiem 10 stopni).

2. Zmodyfikuj wykres by pozwalał na zooming myszą. Dodaj przycisk resetujący powiększenie.

### **Ćwiczenie 5**

1. Dodaj do projektu bibliotekę darmowych motywów dla biblioteki PrimeFaces edytując plik konfiguracyjny projektu Mavena (pom.xml):

```
<dependency>
  <groupId>org.primefaces.themes</groupId>
  <artifactId>all-themes</artifactId>
  <version>1.0.10</version>
</dependency>
```

2. Otwórz do edycji plik deskryptora aplikacji webowej (web.xml) i zmień motyw aplikacji dodając odpowiedni wpis <context-param>. Zaobserwuj różnice w wyglądzie aplikacji dla kilku przykładowych motywów.

```
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>luna-pink</param-value>
</context-param>
```