

Kod szkolenia: **JS/BP**

Tytuł szkolenia: **Wzorce i praktyki programistyczne w języku JavaScript**

Dni: 4

Opis:

Adresaci szkolenia

Szkolenie przeznaczone dla osób znających język JavaScript na poziomie średnio zaawansowanym, którzy chcą poszerzyć posiadaną wiedzę poprzez poznanie wzorców, dobrych praktyk oraz powszechnie wykorzystywanych narzędzi wspierających pracę programisty.

Cel szkolenia

Celem szkolenia jest zapoznanie uczestników dobrymi praktykami stosowanymi na co dzień przez doświadczonych programistów języka JavaScript, a także nabycie praktycznych umiejętności budowania złożonych aplikacji.

Mocne strony szkolenia

Szkolenie ma charakter warsztatowy. Taki sposób prowadzenia zajęć umożliwia efektywne przyswojenie wiedzy oraz zetknięcie się z realnymi problemami, które mogą wystąpić w czasie codziennej pracy.

Wymagania

Od uczestników wymagana jest znajomość języka JavaScript na poziomie średnio zaawansowanym.

Parametry szkolenia

3*8 godzin (3*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Program szkolenia:

1. Wprowadzenie

- o Podsumowanie istotnych koncepcji języka
- o Standardy ECMAScript 2015/2016/2017 i ich wsparcie
- o Zasady budowania zarządzalnego kodu
- o Stosowanie konwencji kodowania



- Praktyki prowadzące do niskiego sprzężenia kodu
 - Tworzenie dokumentacji
 - Walidowanie poprawności składni skryptu z JSLint
2. Praktyki programistyczne
 - Poprawne wykorzystanie podstawowych elementów języka
 - Funkcje i związane z nimi konstrukcje
 - Tworzenie i reużywanie obiektów
 - Obsługa zdarzeń
 - Praca z przeglądarką
 - Wykorzystanie DOM API
 3. Wzorce w JavaScript
 - Reużywanie kodu
 - Uzyskiwanie modularnych i bezpiecznych rozwiązań
 - Wybrane wzorce GOF w oparciu o praktyczne przykłady
 - Wzorce / rozwiązania stosowane w popularnych bibliotekach
 4. Testowanie
 - Planowanie, tworzenie i uruchamianie testów
 - Wykorzystanie obiektów typu mock i stub
 - Testy jednostkowe (JUnit, Jasmine, Mochajs)
 - Automatyzacja testów z użyciem Karma i Buster.js
 - Testy end-to-end z użyciem Protractor
 5. Wydajność
 - Przegląd narzędzi profilujących / testujących wydajność
 - Efektywne ładowanie i wykonywanie skryptów
 - Praca z obiektami, zasięg zmiennych oraz kontekst wykonania
 - Algorytmy i sterowanie przepływem
 - DOM Scripting
 - Praca z tekstem i wyrażeniami regularnymi
 - Komunikacja sieciowa
 - Najlepsze praktyki
 6. Automatyzacja
 - Zarządzanie strukturą aplikacji
 - Łączenie, kompresowanie, zaciemnianie skryptów
 - Generowanie dokumentacji
 - Budowanie produkcyjnej wersji rozwiązania

