

## Firma SoftSystem sp. z o.o. zaprasza na praktyczny kurs programowania obiektowego

Firma SoftSystem Sp. z o.o. pragnie zaprosić studentów Politechniki Lubelskiej na serie bezpłatnych zajęć z dziedziny tworzenia oprogramowania komputerowego. Zajęcia będą miały formę wykładów z prezentacją omawianych zagadnień w praktyce. Prowadzący będzie implementował i uruchamiał przykłady podczas zajęć oraz omawiał szczegóły implementacji. Dla studentów będą udostępnione materiały szkoleniowe do pobrania. Zachęcamy do aktywnego uczestnictwa w zajęciach poprzez równoległe tworzenie aplikacji na własnym sprzęcie komputerowym.

### *Cel zajęć*

Celem zajęć jest przekazanie wiedzy praktycznej z zakresu tworzenia oprogramowania w różnych technologiach, z uwzględnieniem dobrych praktyk i wzorców.

### *Nagrody dla aktywnych*

Wśród regularnie uczęszczających na zajęcia rozlosowane zostaną nagrody książkowe.

### *Informacje o firmie*

Firma SoftSystem Sp. z o.o. została założona w 1990 roku i zatrudnia obecnie 440+ specjalistów. Dostarczamy innowacyjne i wysokiej jakości rozwiązania informatyczne dla klientów z branży medycznej. Nasze oprogramowanie jest używane przez 377+ klinik i laboratoriów medycznych w USA i Kanadzie. Większość naszych rozwiązań jest zorientowana na architekturze Unix serwer – Windows klient. Stwarza to możliwość osiągnięcia optymalnych rezultatów poprzez niezawodność, wysoką wydajności i skalowalności platformy UNIX. Korzystanie z systemu Windows, jako platformy klient pozwala na tworzenie aplikacji z powszechnym i przyjaznym dla użytkownika interfejsem.

W grudniu 2012 roku firma SoftSystem Sp. z o.o. otworzyła swój oddział w Lublinie.

## Część 1 - Object Oriented Programming – dobre praktyki

3 zajęcia po 120 minut

Zdobywanie biegłości w programowaniu składa się z dwóch rzeczy: wiedzy i pracy. Konieczne jest zdobywanie wiedzy na temat zasad, wzorców oraz heurystyk. Następnie należy poprzez pracę i praktykę przekuć wiedzę w umiejętności. W części tej zostaną przedstawione procesy myślenia dobrego programisty oraz sztuczki i techniki przez niego wykorzystywane. Omówimy dobre praktyki programowania zorientowanego obiektowo oraz podstawowe wzorce projektowe. Teoria będzie zaprezentowana i wyjaśniona na praktycznych przykładach w języku C# oraz Java.

### 1. Dobre praktyki programowania zorientowanego obiektowo.

- Wprowadzenie do OOP (enkapsulacja, polimorfizm, dziedziczenie).
- Znaczenie nazw w programowaniu.
- Zasada KISS, DRY.
- Zasada SOLID:
  - S** - SRP Single responsibility principle
  - O** - OCP Open/closed principle
  - L** - LSP Liskov substitution principle
  - I** - ISP Interface segregation principle
  - D** - DIP Dependency inversion principle

### 2. Dobre praktyki programowania zorientowanego obiektowo c.d.

- Refaktoring przykładowego programu z wykorzystaniem wcześniej poznanych zasad.
- Omówienie Dependency Injection – z przykładami (IoC container).

### 3. Wzorce projektowe w praktyce.

- Omówienie wybranych wzorców projektowych:
  - Prototype
  - Adapter
  - Factory
  - Chain of Responsibility
  - Memento
  - Inne ...

### 4. C# oraz Java dla zaawansowanych – nowe mechanizmy języków.

- C#: LINQ, functional programming, dynamic language, multithreading
- Java: WatchService, Fork/Join Task, udogodnienia na poziomie składni języka.
- Omówienie i uruchomienie przykładów.

## Część 2 – Technologia ASP.NET

3 zajęcia po 120 minut

Technologia ASP.NET MVC 3 z zastosowaniem Entity Framework Code First, Scaffolding, Dependency Injection. Podczas zajęć zostanie od podstaw stworzona i uruchomiona kompletna aplikacja internetowa. Aplikacja umożliwi zarządzanie bazą pacjentów oraz zapewni autoryzowany dostęp do wybranych funkcjonalności.

### 1. Wprowadzenie do ASP.NET MVC

- Środowisko Visual Studio oraz IIS.
- Kontroler MVC.
- Widoki Razor.
- Deployment i debugowanie aplikacji.

### 2. Tworzenie modelu danych z użyciem Entity Framework Code First

- Wprowadzenie do Entity Framework.
- Stworzenie bazy danych na podstawie modelu.
- Użycie modelu w kontrolerze i widoku – scaffolding.

### 3. Infrastruktura ASP.NET, bezpieczeństwo, unit testing, deployment

- Dodanie możliwości logowania do aplikacji.
- Cache, Resources, Globalization.
- Unit testing.
- Bezpieczeństwo aplikacji internetowej.

## Część 3 – Połączenie technologii Java z .NET

4 zajęcia po 120 minut

W tej części zostanie przedstawiony proces tworzenia aplikacji klient-serwer. Wykorzystując zintegrowane środowisko programistyczne NetBeans oraz takie technologie jak: EJB, Hibernate oraz JAX-WS stworzymy część serwerową aplikacji. Jako serwera aplikacji użyjemy Oracle WebLogic Server. Technologia .NET WinForms posłuży nam do zaimplementowania części klienta, który będzie się komunikował z serwerem przy pomocy stworzonych wcześniej Java WebServices. Na zajęciach omówimy projektowanie oraz budowanie aplikacji od podstaw. Dodatkowo będzie zaprezentowana lekka aplikacja web'owa w technologii JSP, korzystająca z istniejących serwisów (przesyłanie danych w formacie JSON). Podczas zajęć zostanie przygotowane środowisko developerskie, zaimplementowane Unit Testy, omówiony tryb debug oraz deployment.

### 1. Tworzenie aplikacji server-side w technologii Java Część 1

- Przygotowanie środowiska – NetBeans.
- Ant – skrypty do kompilacji/budowania aplikacji.
- Hibernate – wstęp do ORM.
- EJB – wstęp do technologii.

### 2. Tworzenie aplikacji server-side w technologii Java Część 2

- Stworzenie modelu danych – od projektowania do realizacji.
- Deployment – udostępnienie aplikacji poprzez Application Server.
- Wystawienie web-services.
- Testowanie – przygotowanie podstawowych Unit Testów.

### 3. Aplikacja klienta .NET WinForms

- Przygotowanie środowiska programistycznego .NET – Visual Studio.
- Projekt aplikacji GUI, a następnie implementacja.
- Konsumowanie udostępnionych wcześniej web-services.
- Bindowanie danych z użyciem DevExpress.

### 4. Konsumowanie istniejących serwisów przez aplikację HTML

- Projekt strony opartej na istniejących web-services.
- Struts – MVC web framework dla JSP.
- Podstawy CSS, JavaScript, JQuery.
- Deployment na application server, uruchomienie oraz debug.