# Nazwa przedmiotu

Technologie Programowania Adaptacyjnego

# Nazwa w języku angielskim

Technologies of Adaptive Programming

# Formy zajęć i liczba godzin w semestrze

1. Wykład: 30
2. Ćwiczenia: 0
3. Laboratorium: 0
4. Projekt: 30
5. Seminarium: 0

# Cel przedmiotu

Celem przedmiotu jest rozszerzenie wiedzy i umiejętności w zakresie programowania o technologie ułatwiające adaptację tworzonego programu do zmieniających się wymogów środowiska w trakcie jego użytkowania. Cel ten obejmuje zagadnienia związane z praktycznym poznaniem technologii pozwalających na odroczeniu decyzji w zakresie współdziałaniu programu ze środowiskiem .NET i otoczeniem zewnętrznym. Studenci poznają wybrane wzorce projektowe, architekturę rozwiązań, biblioteki i konstrukcje językowe oraz ich praktyczną implementację w środowisku .NET w zakresie umożliwiającym projektowanie i opracowywanie własnych programów zgodnie z paradygmatami programowania adaptacyjnego.

# Efekty kształcenia

Po skończeniu kursu studenci będą:

1. ocenić i posługiwać się konstrukcjami języka C# dotyczących programowania dynamicznego,
2. posługiwać się technologiami komponowania programu z niezależnie opracowanych modułów,
3. analizować i wykorzystywać dostępne technologie w zakresie adaptacji istniejącego programu do różnych repozytoriów danych procesowych,
4. rozumieć potrzebę i stosować spójne mechanizmy konfiguracji programu złożonego z niezależnie utworzonych modułów,
5. analizować potrzeby śledzenia aktywności programu i wybierać technologie programowania gwarantujące możliwość ich adaptacji w trakcie użytkowania,
6. testować stosowane rozwiązania uwzględniając architekturę programu jako zbioru luźno powiązanych modułów,
7. analizować i dyskutować na temat cech i zakresu stosowania poszczególnych technologii programowania adaptacyjnego.

# Metody weryfikacji efektów kształcenia

* Aktywne uczestnictwo w dyskusji moderowanej przez prowadzącego.
* Prezentacja na zadany temat.
* Kolokwium.
* Zaliczenie zadań projektowych.

# Wymagania wstępne

Do zrozumienia treści wymagana jest wiedza w zakresie:

* Programowania obiektowego.
* Technologii programowania.
* Programowania komponentowego.

# Organizacja przedmiotu i treści kształcenia

W trakcie zajęć programowanie adaptacyjne jest przedstawione jako pewien katalog technologii programistycznych, które stosuje się na etapie projektowania i wdrażania programu by zwiększyć możliwości jego adaptacji do zmieniających się warunków środowiska produkcyjnego w czasie, gdy jest on użytkowany.

## Wykład

W ramach wykładu prezentacja wybranego zagadnienia jest ilustrowana kodem źródłowym przykładowego programu. Podsumowaniem prezentacji jest dyskusja wszystkich obecnych studentów moderowana przez prowadzącego. W ramach dyskusji omawiane są scenariusze, definiowane najważniejsze cele i analizowany zakres stosowania omawianych rozwiązań. Udział w dyskusji jest oceniany.

Zajęcia koncertują się na dyskusji dostępnych artefaktów, które dotyczą i wspierają programowanie adaptacyjne, a w tym:

* najczęściej występujących w praktyce scenariuszy,
* architektury i dostępnych wzorców projektowych,
* dostępnych bibliotek wspierających omawiane rozwiązania,
* składni i semantyki wybranych konstrukcji języka programowania,
* metod i narzędzi testowania,

W trakcie wykładu omawiane są następujące zagadnienia:

* składnia i semantyka przydatnych dla programowania adaptacyjnego konstrukcji języka C#, a w szczególności:
  + programowanie równoległe i asynchroniczne,
  + refleksja,
  + programowanie atrybutowe,
  + programowanie dynamicznego,
* reprezentacja wyrażeń i ich konwersja do postaci wymaganej przez wybrane systemy zewnętrzne,
* architektura i wzorce projektowe związane z dostępem do repozytoriów danych procesowych:
  + serializacja do zachowania stanu obiektów i ich relacji z wykorzystaniem takich formatów jak XML, JSON, itp.
  + mapowanie obiektowo-relacyjne
* komponowanie programu z niezależnych modułów:
  + wstrzykiwanie zależności
  + rozszerzanie funkcjonalności z wykorzystaniem komponentów opracowanych niezależnie (plug-in)
  + zapewnienie separacji odpowiedzialności,
  + wersjonowanie komponentów składowych,
  + domeny aplikacji,
* śledzenia aktywności programu w środowisku produkcyjnym:
  + wieloźródłowe systemy śledzenia,
  + semantyczne śledzenie,
  + dynamiczne konfigurowanie mechanizmów śledzenia,
  + śledzenie z wykorzystaniem przetwarzania w chmurze,
  + współużywalne wzorce projektowe promujące podział uwagi (ang. separation of concerns) przy tworzeniu komponentów programu,
* projektowanie spójnego systemu konfigurowania:
  + programu jako pewnej całości,
  + niezależnie utworzonych komponentów programu,
  + właściwości w kontekście poszczególnych użytkowników,
* testy jednostkowe i integracyjne:
  + wymagania środowiska testowania w zakresie budowy programu złożonego z niezależnie opracowanych komponentów,
  + symulowanie danych procesowych na potrzeby testów,
  + symulowanie funkcjonalności środowiska produkcyjnego na potrzeby testowania,

## Projekt

W ramach projektu studenci realizują niezależne zadania lub jeden etapowany projekt. Celem tej formy zajęć jest potwierdzenie umiejętności zastosowania w praktyce wiedzy teoretycznej omawianej w trakcie wykładu. Zadaniem studentów jest zaprojektowanie i uruchomienie własnych programów komputerowych realizujących wybrane scenariusze i testowanie proponowanych rozwiązań z wykorzystaniem testów jednostkowych. Każde zadanie lub etap projektu jest oceniany niezależnie.

# Formy zaliczenia

Aktywność na zajęciach, jedno pisemne kolokwium na koniec semestru oraz realizacja projektów. Końcowa ocena bazuje na średniej ważonej 20% aktywność, 40% za kolokwium i 40% za realizację zadań projektowych.

Ocenianymi elementami prac na zajęciach laboratoryjnych są:

* poprawność,
* terminowość,
* umiejętność zrozumiałego omówienia zaproponowanego rozwiązania.

# Literatura podstawowa

* C# 6.0. Kompletny przewodnik dla praktyków. Wydanie V; Mark Michaelis, Eric Lippert; Wydawnictwo: Helion; 2016
* Współbieżność w języku C#. Receptury; STEPHEN CLEARY; Wydawnictwo: Helion; 2017
* Dependency Injection in .NET; Mark Seemann; Manning Publications; 2011

## Literatura uzupełniająca

* Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku; Erich Gamma; Helion; 2017
* Oprogramowanie łatwe w utrzymaniu. Pisz kod podatny na przyszłe zmiany; JOOST VISSER; Wydawnictwo: Helion; 2016
* Reactive Programming for .NET Developers; Antonio Esposito; Michael Ciceri; Packt Publishing; 2016
* Entity Framework Core Cookbook - Second Edition; Ricardo Peres; Packt Publishing; 2016