

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Programowanie współbieżne z wykorzystaniem platformy .NET		Cykl kształcenia rozpoczynający się w roku akademickim: 2023/24	
Nazwa kierunku studiów, poziom i profil kształcenia: informatyka, studia I-go stopnia, inżynierskie			
Język wykładowy: polski		Rodzaj zajęć: specjalistyczne	
Rok studiów: III		Semestr: VI	
Liczba punktów ECTS przypisana zajęciom: 4		Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail	
Jednostka organizacyjna: Instytut Inżynierii Technicznej, Zakład Informatyki			
FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN			
Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:			
Studia stacjonarne		Studia niestacjonarne	
Wykład:	15	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	30	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	45	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			
Wymagania wstępne i dodatkowe:wymagana jest wiedza i umiejętności zdobyte na zajęciach kształcenia kierunkowego - <i>podstawy programowania, algorytmy i struktury danych, programowanie obiektowe</i> . Ponadto wymagana jest umiejętność programowania w języku C#.			

Cel (cele) kształcenia dla zajęć: poznanie technik programowania współbieżnego z wykorzystaniem języka C# i biblioteki TPL platformy .NET oraz realizacja aplikacji implementujących obliczenia współbieżne, w tym aplikacji WPF			
EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW			
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się			
Symbol efektów uczenia się określonego dla zajęć	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:		Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie			
W_01	podstawowe aspekty teoretyczne związane z programowaniem współbieżnym		K_W06, K_W08
W_02	techniki programowania wielowątkowego oraz ich implementację w środowisku .NET		K_W06, K_W08, K_W10
Umiejętności - potrafi			
U_01	zaprojektować i utworzyć aplikację wykorzystującą metody i właściwości klas: <i>Thread</i> , <i>ThreadPool</i> , <i>Task</i> oraz <i>Parallel</i>		K_U01, K_U03, K_U05, K_U14
U_02	wdrożyć mechanizmy synchronizacji wątków, zadań		K_U01, K_U03, K_U11
U_03	potrafi wykorzystać mechanizmy wielowątkowości w aplikacjach z GUI w tym w aplikacjach z zapytaniami do bazy danych		K_U01, K_U03, K_U05, K_U06, K_U12
Kompetencji społecznych - jest gotów do			
K_01	podnoszenia swoich kwalifikacji poprzez samokształcenie		K_K01
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA			
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):			
Symbol treści programowych	Opis treści programowych	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć
wykład			

TP-01	Ogólna idea obliczeń współbieżnych, równoległych oraz rozproszonych. Ogólna charakterystyka platformy .NET i języka C# w kontekście programowania współbieżnego, rola biblioteki TPL. Wątki i procesy, hermetyzacja aplikacji w platformie .NET - koncepcja domen aplikacji DA w .NET, domeny aplikacji a procesy.	Wykład podający, wykład problemowy, pogadanka	egzamin ustny
TP-02	Tworzenie wątków i przeniesienie do nich obliczeń, wykorzystanie klasy Thread. Problem zagłódzenia wątków. Zarządzanie wątkami, sekcje krytyczne i operacje atomowe. Przykładowe rozwiązania i ich analiza.	Wykład podający, wykład problemowy, pogadanka	egzamin ustny
TP-03	Wybrane mechanizmy synchronizacji wątków - problem uczujących filozofów, czytelników i pisarzy. Komunikacja między wątkami. synchronizacja wątków z różnych procesów, analiza przykładowych rozwiązań.	Wykład podający, wykład problemowy, pogadanka	egzamin ustny
TP-04	Tworzenie zadań oraz operacje na zadaniach i ich synchronizacja - wykorzystanie klasy Task. Klasa <i>Parallel</i> - dystrybucja obliczeń w pętlach zrównoleglonych.	Wykład podający, wykład problemowy, pogadanka	egzamin ustny
TP-05	Zapytania do bazy danych z wykorzystaniem PLINQ.	Wykład podający, wykład problemowy, pogadanka	egzamin ustny

TP-06	Wykorzystanie wielowątkowości w aplikacjach z interfejsem GUI (dla WPF). Analiza przykładowych rozwiązań	Wykład podający, wykład problemowy, pogadanka	egzamin ustny
zajęcia praktyczne			
TP-07	Realizacja aplikacji konsolowych - tworzenie domen aplikacji, badanie rozkładu procesów i wątków w kontekście tworzonych domen DA.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-08	Realizacja aplikacji konsolowych operujących na wątkach (zastosowanie klasy <i>Thread</i>), przeniesienie metody do oddzielnego wątku.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-09	Operacje na wątkach: usypianie wątku, przerwanie działania, wstrzymywanie i wznowianie działania, wątki w tle, zmiana priorytetu wątku. Budowanie wątków w oparciu o delegaty <i>ThreadStart</i> oraz <i>ParametrizedThreadStart</i> .	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-10	Realizacja aplikacji wielowątkowych - badanie czasu rozwiązywania zadania przez zaimplementowanie wielowątkowości.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-11	Realizacja aplikacji z pulą wątków z zastosowaniem klasy <i>ThreadPool</i> - badanie funkcjonowania puli z wykorzystaniem metod tej klasy.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-12	Tworzenie aplikacji konsolowych z mechanizmami sygnalizacji ukończenia pracy wątków - przypadek bez puli oraz z pulą wątków.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów

TP-13	Tworzenie aplikacji implementujących mechanizmy synchronizacji zapisu do zmiennych współdzielonych - tworzenie sekcji krytycznych (<i>lock</i>) oraz definiowanie operacji atomowych (klasa <i>Interlocked</i>). Badanie wpływu zastosowania atrybutu [<i>ThreadStatic</i>] na wartości zmiennych globalnych w wątkach.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-14	Realizacja aplikacji implementujących mechanizmy synchronizacji wątków - rozwiązywanie problemu uczących filozofów, problem czytelników i pisarzy, problem producenta i konsumenta.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-15	Realizacja aplikacji implementujących klasę <i>Task</i> . Praca z zdaniami, przekazywanie danych do- oraz z zadań, synchronizacja zadań.	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-16	Wykorzystanie pętli zrównoleglonych <i>for</i> oraz <i>foreach</i> .	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-17	Realizacja aplikacji z wykorzystaniem zapytań do bazy danych (PLINQ i SQL)	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
TP-18	Realizacja aplikacji z interfejsem graficznym WPF - uruchamianie metody zdarzeniowej kontrolki w dodatkowym wątku	realizacja projektów w postaci aplikacji	bieżąca kontrola realizacji zadań, ocena wykonanych projektów
ZALECANA LITERATURA (w tym pozycje w języku obcym)			

Literatura podstawowa:

1. Matulewski J i inni.: *Visual Studio 2010 dla programistów C#*, Helion 2011r.
2. Michaelis M., Lippert E.: *C# 7.0 Kompletny przewodnik dla praktyków*, Helion 2017r.
3. Sharp J.: *Microsoft Visual C# 2017 krok po kroku*, wyd. APNPromise 2017r.

Literatura uzupełniająca:

1. Oficjalny serwis: <https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/>
2. Oficjalny serwis: <https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/task-based-asynchronous-programming>

III. INFORMACJE DODATKOWE**BILANS PUNKTÓW ECTS****OBCIĄŻENIE PRACĄ STUDENTA (godziny)**

Forma aktywności	Liczba godzin
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	45
Praca własna studenta	50
SUMA GODZIN:	95

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 4	2
	Praca własna studenta		2

OPIS PRACY WŁASNEJ STUDENTA:**OPIS PRACY WŁASNEJ STUDENTA:**

Przygotowanie do zajęć praktycznych:

- 1) ugruntowana podstawowa wiedza teoretyczna w zakresie realizowanej tematyki zajęć: U_01, U_02, U_03 (ocena realizowanych aplikacji)
 - 2) znajomość klas dedykowanych do zastosowania w poszczególnych problemach W_01, W_02, U_01, U_02, U_03 (ocena realizowanych aplikacji)
- Czytanie wskazanej literatury - poszerzenie wiedzy dotyczące zagadnień związanych zagadnień ujętych w W_01, W_02.

KRYTERIA OCENIANIA

Ocena kształtująca: podjęta będzie na podstawie zajęć praktycznych, które kończą się zaliczeniem na ocenę. Warunkiem uzyskania oceny pozytywnej z zajęć praktycznych jest realizacja wszystkich przewidzianych ćwiczeń.

Na ocenę dostateczną student opanowuje umiejętności praktyczne w realizacji aplikacji konsolowych i aplikacji z GUI implementujących mechanizmy programowania wielowątkowego w zakresie podstawowym, z pomocą prowadzącego zajęcia. Na ocenę dobrą student sprawnie i samodzielnie realizuje projekty, a także wykazuje się dobrą wiedzą teoretyczną.

Na ocenę bardzo dobrą student prezentuje wysokie umiejętności praktyczne, biegle posługując się wszystkimi podstawowymi i zaawansowanymi aspektami programowania aplikacji wielowątkowych, posiada bardzo dobrą wiedzę teoretyczną.

Ocena podsumowująca: podjęta na podstawie końcowego egzaminu ustnego, dotyczącego treści programowych, realizowanych na wykładach (W_01, W_02). Warunkiem przystąpienia do egzaminu jest uzyskanie oceny pozytywnej z zajęć praktycznych.

**INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA
KSZTAŁCENIA NA ODLEGŁOŚĆ**