

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Nowoczesne techniki programowania D1_17
Nazwa przedmiotu (j. ang.):	Modern programming techniques
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Bezpieczeństwo systemów informatycznych
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne
Obszar kształcenia:	nauki techniczne
Dziedzina:	nauki techniczne
Dyscyplina nauki:	informatyka
Koordinator przedmiotu:	Dr inż. Bartosz Trybus Dr Marcin Skuba Dr inż. Piotr Wais

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	kształcenia specjalnościowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	IV, 7
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 15 h, ćw. laboratoryjne 30 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Programowanie niskopoziomowe, Programowanie I, II, Bazy danych

3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):	2	stacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiąganych na tych zajęciach	obecność na wykładach obecność na ćwiczeniach laboratoryjnych ćwiczenia projektowe udział w konsultacjach W sumie: ECTS	15 30 2 47 1,6
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)	przygotowanie ogólne opracowanie dokumentacji (sprawozdań) praca nad projektem studiowanie zalecanej literatury praca w sieci w sumie: ECTS	3 5 2 2 12 0,4
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):	udział w zajęciach praca samodzielna w sumie: ECTS	30 7 37 1,3

4. Opis przedmiotu

<p>Cel przedmiotu: Zdobycie wiedzy i umiejętności praktycznych z zakresu tworzenia systemów informatycznych z użyciem nowoczesnych języków i technologii programowania.</p>
<p>Metody dydaktyczne: wykład informacyjny, ćwiczenia laboratoryjne, symulacja</p>
<p>Treści kształcenia: Wykłady i laboratoria: Charakterystyka wybranych technologii: JEE, .NET, PHP, Web Services i WebAPI, frameworki JavaScript Implementacja warstwy danych systemu informatycznego Implementacja warstwy logiki system informatycznego Implementacja warstwy prezentacji systemu informatycznego</p>

Wzorce architektoniczne: MVC, MVVM
Zarządzanie projektami metodyką zwinną (SCRUM)

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia (w sumie wymienić ok. od 3 do 9 efektów - podać numery efektów z listy dla danego kierunku/specjalności – opublikowane na stronie uczelni; podać TYLKO te efekty (tam gdzie to możliwe i stosowne w trzech kategoriach, np. kompetencje społeczne mogą nie być realizowane w tym przedmiocie), na których osiągnięcie kładzie się nacisk w ramach przedmiotu, wybrane efekty kierunkowe powinny być bardziej szczegółowo sformułowane niż te dla całej specjalności, tak aby były weryfikowalne – dlatego mają osobne symbole jako efekty przedmiotu)

Efekt przedmiotu (kod przedmiotu + kod efektu kształcenia)	Student, który zaliczył przedmiot (spełnił minimum wymagań)	Efekt kierunkowy
D1_17_K_W01 D1_17_K_W02 D1_17_K_W03	Wiedza: 1. Zna cechy oraz obszary zastosowań współczesnych narzędzi programowania. 2. Zna techniki implementacji warstwy danych, logiki i prezentacji systemów informatycznych. 3. Zna wzorce architektoniczne i zasady zarządzania projektami.	K_W05 K_W06 K_W08 K_W12
D1_17_K_U01 D1_17_K_U02 D1_17_K_U03	Umiejętności 1. Potrafi zaprogramować aplikację webową z użyciem technologii JEE, .NET, PHP. 2. Potrafi utworzyć warstwę prezentacji wykorzystując w niej język JavaScript z odpowiednimi frameworkami. 3. Potrafi zastosować wzorce architektoniczne MVC lub MVVM w tworzonej aplikacji.	K_U02 K_U11 K_U15 K_U25
D1_17_K_K01	Kompetencje społeczne 1. Potrafi pracować w zespole projektującym system informatyczny składający się z warstw danych, logiki i prezentacji.	K_K04 K_K08

Sposoby weryfikacji efektów kształcenia:

(np. dyskusja, gra dydaktyczna, zadanie e-learningowe, ćwiczenie laboratoryjne, projekt indywidualny/ grupowy, zajęcia terenowe, referat studenta, praca pisemna, kolokwium, test zaliczeniowy, egzamin, opinia eksperta zewnętrznego, etc. Dodać do każdego wybranego sposobu symbol zakładanego efektu, jeśli jest ich więcej)

Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca – przykładowe sposoby jej wystawienia poniżej	Ocena końcowa przykładowe sposoby jej wystawienia poniżej
1	D1_17_K_W01 D1_17_K_W02 D1_17_K_W03 D1_17_K_U01 D1_17_K_U02 D1_17_K_U03	Zaliczenie	sprawdzian wiedzy, sprawdzian umiejętności	rozwiązanie zadania problemowego (zadań), analiza przypadku
2	D1_17_K_U01 D1_17_K_U02 D1_17_K_U03 D1_17_K_K01	ćwiczenia laboratoryjne	kolokwia zaliczające partie materiału, rozwiązania zadań domowych	średnia z ocen formujących

Kryteria oceny (oceny 3,0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):		
w zakresie wiedzy		Efekt kształcenia
Na ocenę 3,0	Student uzyskał min. 50% wymaganej wiedzy w zakresie obowiązującego materiału. Student: 1. Zna obszary zastosowań współczesnych narzędzi programowania, 2. Zna wybraną technikę implementacji warstwy danych, logiki i prezentacji systemów informatycznych, 3. Zna wzorzec architektoniczny MVC.	D1_17_K_W01 D1_17_K_W02 D1_17_K_W03
Na ocenę 5,0	Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student: Zna podstawowe oraz zaawansowane cechy współczesnych narzędzi programowania i techniki implementacji warstwy danych, logiki i prezentacji systemów informatycznych.	D1_17_K_W01 D1_17_K_W02 D1_17_K_W03
w zakresie umiejętności		
Na ocenę 3,0	Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi: 1. Zaprogramować warstwę danych i logiki w wybranej technologii. 2. Utworzyć warstwę prezentacji dla aplikacji webowej. 3. Zastosować język JavaScript do usprawnienia warstwy prezentacji.	D1_17_K_U01 D1_17_K_U02 D1_17_K_U03
Na ocenę 5,0	Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie: Utworzyć aplikację webową z wykorzystaniem MVC i WebAPI, zastosować biblioteki jQuery, Bootstrap, AngularJS i in. oraz metodykę zwinną w projekcie informatycznym.	D1_17_K_U01 D1_17_K_U02 D1_17_K_U03
w zakresie kompetencji społecznych		
Na ocenę 3,0	Student osiągną wymagane kompetencje społeczne na poziomie min. 50%. Potrafi pracować w zespole projektującym system informatyczny składający się z warstw danych, logiki i prezentacji.	D1_17_K_K01
Na ocenę 5,0	Student osiągną wymagane kompetencje społeczne na poziomie wyższym niż 90%. 1. Potrafi pracować w zespole projektującym system informatyczny składający się z warstw danych, logiki i prezentacji. 2. Jest w stanie pełnić rolę kierownika zespołu realizującego zadanie projektowe.	D1_17_K_K01
<p>Zalecana literatura (w podziale na literaturę podstawową i uzupełniającą):</p> <p>Podstawowa:</p> <ol style="list-style-type: none"> 1. Strona WWW: emateriały.pwsz.krosno.pl. 2. C# 6.0 i MVC 5 : tworzenie nowoczesnych portali internetowych / Krzysztof Żydzik, Tomasz Rak 3. Microsoft Visual Studio 2012 : Programowanie w C# / Dawid Farbaniec 4. Java : kompendium programisty / Herbert Schildt 5. Visual Studio 2013 : podręcznik programowania w C# z zadaniami / Jacek Matulewski 6. PHP i MySQL : od nowicjusza do wojownika ninja / Kevin Yank <p>Uzupełniająca:</p> <ol style="list-style-type: none"> 1. TDD [Test-Driven Development] : programowanie w Javie sterowane testami : naucz się podstaw metodyki TDD / Viktor Farcic, Alex Garcia 		

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin: (np. indywidualne konsultacje, poprawa prac, przygotowanie projektu zaliczeniowego, egzaminu, przygotowanie ćwiczeń e-learningowych). Przykład poniżej

Aktualizacja wykładów, ćwiczeń i zadań domowych – 30 godzin

Ocena sprawozdań i zadań domowych – 20 godzin

Konsultacje – 10 godzin

Przygotowanie zadań zaliczeniowych – 5 godzin

W sumie: 65 godzin