

KARTA PRZEDMIOTU**I. Dane podstawowe**

Nazwa przedmiotu	Laboratorium programowania: aplikacje w różnych środowiskach programistycznych
Nazwa przedmiotu w języku angielskim	Programming laboratory: applications in various development environments
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	I stopnia
Forma studiów (stacjonarne, niestacjonarne)	stacjonarne
Dyscyplina	Informatyka, informatyka techniczna i telekomunikacja
Język wykładowy	polski

Koordinator przedmiotu	mgr Krzysztof Buszowski
------------------------	-------------------------

Forma zajęć (<i>katalog zamknięty ze słownika</i>)	Liczba godzin	semestr	Punkty ECTS
wykład			3+3
konwersatorium			
ćwiczenia			
laboratorium	30+30	V+VI	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	Wstęp do programowania Podstawy informatyki i programowania Programowanie obiektowe
-------------------	---

II. Cele kształcenia dla przedmiotu

Wykorzystanie rozproszonego systemu kontroli wersji (GIT), ciągłej integracji (CI) oraz narzędzi zarządzania projektem
Omówienie wybranych framework'ów i bibliotek umożliwiających implementację części klienta (desktop application)
Omówienie wybranych framework'ów oraz bibliotek umożliwiających implementację części serwerowej (back-end)
Omówienie dobrych praktyk programistycznych

III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
WIEDZA		
W_01	Student potrafi stosować rozproszony system kontroli wersji oraz narzędzia zarządzania projektem	K_W08
W_02	Student potrafi wykorzystać biblioteki oraz framework'i	K_W06, K_W08
UMIEJĘTNOŚCI		
U_01	Student potrafi stworzyć aplikację składającą się z dwóch funkcjonujących oddzielnie komponentów (desktop client/application; back-end service)	K_U08, K_U17
KOMPETENCJE SPOŁECZNE		
K_01	Ma świadomość ograniczenia swojej wiedzy i umiejętności, rozumie potrzebę dokształcania się i podnoszenia kompetencji zawodowych i osobistych	K_K06

IV. Opis przedmiotu/ treści programowe

<ol style="list-style-type: none"> 1. Wstęp do zajęć. Opis technologii, które będą wykorzystywane na zajęciach. Wstęp do git'a. Kontynuacja git. Praca z repozytoriami. Podstawowe komendy git. Praca z gałęziami. 2. Wstęp do Java. Krótko o JVM i ekosystemie. Omówienie składni i proste zadania do zrobienia. 3. Zarządzanie zależnościami oraz proces budowania z wykorzystaniem Gradle'a. Build lifecycle. Testy (unit + it). 4. Implementacja aplikacji GUI (JavaFX). Omówienie wzorców projektowych ułatwiających implementację i utrzymanie. 5. Testowanie kodu. 4. Persystowanie stanu aplikacji (bazy danych). 6. Protokół HTTP/2. Jak wygląda żądanie oraz odpowiedź HTTP. Implementacja prostego serwera HTTP w oparciu o framework Spark. 7. REST. Co to jest i po co tego używać. W jaki sposób projektować API. Zabezpieczanie API (uwierzelnianie użytkownika, CSRF) 8. Wprowadzenie do Spring + Spring MVC. Implementacja serwera HTTP za pomocą kontenera używając adnotacji. Testy integracyjne.

V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne (lista wyboru)	Metody weryfikacji (lista wyboru)	Sposoby dokumentacji (lista wyboru)
WIEDZA			
W_01	Ćwiczenia praktyczne, Praca pod kierunkiem	Projekt	Karta oceny projektu
W_02	Ćwiczenia praktyczne, Praca pod kierunkiem	Projekt	Karta oceny projektu
UMIEJĘTNOŚCI			
U_01	Ćwiczenia praktyczne design thinking	Projekt	Karta oceny projektu

KOMPETENCJE SPOŁECZNE			
K_01	Dyskusja, Metoda problemowa PBL design thinking	Projekt	Karta oceny projektu

VI. Kryteria oceny, wagi...

Przygotowanie projektu zaliczeniowego w wybranej technologii.

VII. Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	100
Liczba godzin indywidualnej pracy studenta	60

VIII. Literatura

Literatura podstawowa
Benjamin J. Evans, David Flanagan, "Java w pigułce. Wydanie VI", Helion 2015 Zasoby internetowe (w tym dokumentacja techniczna stosowanych technologii) Materiały przygotowane przez prowadzącego zajęcia
Literatura uzupełniająca
Benjamin J. Evans, Martijn Verburg, "The Well-Grounded Java Developer", Manning 2012