

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim Indywidualny projekt programistyczny
2.	Nazwa przedmiotu/modułu w języku angielskim Individual programming project
3.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
4.	Kod przedmiotu/modułu 24-ISSP-S1-E3-Ipp
5.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub fakultatywny</i>) fakultatywny
6.	Kierunek studiów Informatyka Stosowana
7.	Poziom studiów (<i>I lub II stopień lub jednolite studia magisterskie</i>) I stopień
8.	Rok studiów (<i>jeśli obowiązuje</i>) 2
9.	Semestr (<i>zimowy lub letni</i>) letni
10.	Forma zajęć i liczba godzin Laboratorium 30 godz
11.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia mgr Mateusz Bancewicz
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu oraz zrealizowanych przedmiotów <ul style="list-style-type: none">• Podstawy tworzenia aplikacji z interfejsem użytkownika w dowolnej technologii, przy użyciu dowolnej biblioteki / frameworka, np. html, python (django, pygame), C++ (QT, QtQuick), Swift (język do urządzeń Apple), Unity3D
13.	Cele przedmiotu Studenci wykorzystują posiadane umiejętności i realizują własne, indywidualne projekty. W czasie zajęć samodzielnie poznają i wykorzystują wybrane przez siebie API do interakcji z użytkownikiem (GUI). Projekt powinien składać się przynajmniej z dwóch części. Pierwsza z nich

	<p>to dokument będący propozycją realizowanego projektu. Dokument powinien mieć taką formę i zawierać takie informacje, aby po jego przeczytaniu zachęcić czytelnika (prowadzącego / inwestora) do zainwestowania w projekt (formą zapłaty jest ocena). Im forma bardziej atrakcyjna, tym lepiej (wzór: projekty z KickStarter.com) Dokument powinien zawierać co najmniej opis pomysłu (uzasadnienie, że realizuję ten projekt w danej technologii, bo chcę się jej nauczyć, też jest ok), wykorzystanie (<i>use cases</i>), opis i uzasadnienie wykorzystanych technologii, harmonogram, koncepcyjne schematy GUI, informacje o repozytorium. Druga część to sam działający projekt. Projekt musi zostać zaakceptowany przez prowadzącego, aby przejść do fazy jego realizacji.</p> <p>Wyniki projektu prezentowane są w formie seminarium. Oceniana jest przede wszystkim zgodność programu z przedstawionym planem oraz zgodność realizacji z harmonogramem. Wszelkie zmiany w zaplanowanych funkcjonalnościach lub u użytych technologiach muszą być skonsultowane z prowadzącym. Zmiany mogą być wprowadzone tylko po jego akceptacji. Kod źródłowy projektu oraz jego dokumentacja muszą być umieszczone w repozytorium w wybranym systemie wersjonowania (np. git, mercurial), a prowadzący od początku musi mieć do niego dostęp.</p> <p>Cele:</p> <ul style="list-style-type: none"> - Uświadomienie studentom istotności planowania projektu - Pokazanie, w jaki sposób planować projekt i jego podzadania - Pokazanie, w jaki sposób proponować i wprowadzać zmiany projektowe - Utrwalenie nawyków związanych z prowadzeniem projektów programistycznych (planowanie projektu, wykorzystanie systemu wersjonowania, tworzenie dokumentacji, prezentacja gotowego produktu) 	
14.	<p>Zakładane efekty kształcenia</p> <p>wiedza:</p> <ul style="list-style-type: none"> * rozumie, jak działa proces powstawania oprogramowania * zna systemy wersjonowania i sposoby ich używania w praktyce * zna wybrane biblioteki graficznego interfejsu użytkownika (GUI) * rozumie istotę i rolę prototypu w cyklu tworzenia oprogramowania <p>umiejętności:</p> <ul style="list-style-type: none"> * umie zaprojektować, zaplanować, wykonać i ukończyć prosty projekt programistyczny * potrafi go dobrze „sprzedać” w ramach prezentacji <p>kompetencje społeczne:</p> <ul style="list-style-type: none"> * potrafi dzielić się swoją wiedzą z innymi 	K_W04, K_U06, K_U08, K_U16, K_U17, K_K02, K_K04, K_K05

15.	Treści programowe	
	<ul style="list-style-type: none"> • Projektowanie aplikacji i planowanie pracy nad projektem. • Systemy wersjonowania i ich użycie w praktyce. • Przegląd wybranych bibliotek GUI. • Prototypowanie. • Prezentacja prototypów projektu. 	
16.	Zalecana literatura (<i>podręczniki</i>)	
	<ul style="list-style-type: none"> • Dokumentacja wybranej technologii dostępna w Internecie 	
17.	Forma zaliczenia poszczególnych komponentów przedmiotu/modułu, sposób sprawdzenia osiągnięcia zamierzonych efektów kształcenia: wykład: seminarium: laboratorium: ocena projektów na podstawie - zgodności projektów z zaakceptowanymi planami - prezentacji końcowej konwersatorium: inne:	
18.	Język wykładowy	
	polski	
19.	Obciążenie pracą studenta	
	Forma aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności
	Godziny zajęć (wg planu studiów) z nauczycielem: - wykład: - ćwiczenia: - laboratorium: - inne:	- 0 - 0 - 30 - 0
	Praca własna studenta np.: - przygotowanie do zajęć: - opracowanie wyników: - czytanie wskazanej literatury: - napisanie raportu z zajęć: - przygotowanie do egzaminu:	- 75 - 0 - 10 - 10 - 0
	Suma godzin	125
	Liczba punktów ECTS	5

*objaśnienie symboli:

K (przed podkreśleniem) - kierunkowe efekty kształcenia

W - kategoria wiedzy

U - kategoria umiejętności

K (po podkreśleniu) - kategoria kompetencji społecznych

01, 02, 03 i kolejne - numer efektu kształcenia