



3. KARTA PRZEDMIOTU

Kod przedmiotu	Z-IDN-U-510b
Nazwa przedmiotu	Język programowania Python
Nazwa przedmiotu w języku angielskim	The Python Programming Language
Obowiązuje od roku akademickiego	2019/2020

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	INŻYNIERIA DANYCH
Poziom kształcenia	I stopień
Profil studiów	Praktyczny
Forma i tryb prowadzenia studiów	Studia niestacjonarne
Zakres	Inżynieria zasobów danych
Jednostka prowadząca przedmiot	Katedra Informatyki i Matematyki Stosowanej
Koordinator przedmiotu	Dr inż. Paweł Stąpór
Zatwierdził	Dr hab. inż. Artur Bartosik, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Przedmiot specjalnościowy
Status przedmiotu	Obowiązkowy
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	Semestr V
Wymagania wstępne	Podstawy informatyki, Algorytmy i struktury danych, Projektowanie relacyjnych baz danych
Egzamin (TAK/NIE)	TAK
Liczba punktów ECTS	5

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	12		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student ma poszerzoną wiedzę nt. programowania proceduralnego i obiektowego. Rozumie pojęcia dziedziczenia, kompozycji i polimorfizmu obiektów w odniesieniu do języka Python.	ID1_W09 ID1_W10 ID1_W11
	W02	Student rozumie i zna zasady budowy aplikacji obsługujących bazy danych.	ID1_W11
	W03	Student ma wiedzę nt. zasad projektowania interfejsu GUI aplikacji zgodnego z architekturą systemu operacyjnego.	ID1_W11
Umiejętności	U01	Student potrafi budować aplikacje w języku Python dla system Windows z wykorzystaniem plików.	ID1_U09
	U02	Student potrafi ocenić przydatność narzędzi programowania do rozwiązywania zagadnień z zakresu inżynierii danych.	ID1_U06
Kompetencje społeczne	K01	Student rozumie potrzebę stałego uzupełniania wiedzy z obszaru nowoczesnych narzędzi i idei informatyki.	ID1_K01

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Pojęcie modułu i atrybutów. Sposoby wykonywania programu w języku Python, importowanie i przeładowywanie modułów, interfejs IDLE.
	2. Typy obiektów wbudowanych, instrukcje, aspekty programowania funkcjonalnego, specjalne tryby przekazywania argumentów funkcji, zakresy nazw i ich użycie.
	3. Sposoby przetwarzania plików tekstowych, zapis obiektów Pythona do pliku, metoda <i>pickle</i> .
	4. Programowanie zorientowane obiektowo, hierarchia dziedziczenia, narzędzie <i>klasy</i> , wyszukiwanie atrybutów obiektu w drzewie dziedziczenia klas, definicja metod klasy, argument <i>self</i> , metody specjalne <i>__init__</i> , <i>__add__</i> i <i>__str__</i> . Narzędzia introspekcji klas: atrybuty <i>__dict__</i> , <i>__class__</i> , <i>__name__</i> . Przykład tworzenia programu zorientowanego obiektowo.
	5. Przechowywanie obiektów w bazie danych <i>shelve</i> , interaktywna obsługa bazy danych. Przykład konsolowego interfejsu bazy danych <i>shelve</i> . Nawiązywanie połączenia z bazami danych <i>SQL</i> , użycie modułu <i>sqlite3</i> i <i>mysql</i> .
	6. Graficzny interfejs użytkownika GUI, moduł <i>tkinter</i> , układ widżetów – metody <i>pack</i> i <i>grid</i> . Funkcje obsługi zdarzeń – metody <i>bind</i> i <i>command</i> . Przykład graficznego interfejsu użytkownika bazy danych <i>shelve</i> .
	7. Przykłady zaawansowanych aplikacji bazodanowych z zakresu inżynierii danych.
laboratorium	1. Sposoby wykonywania programu w języku Python, importowanie i przeładowywanie modułów, interfejs IDLE.
	2. Przetwarzanie typowych struktur danych języka Python: listy, krotki, słowniki zbiorów. Instrukcje sterujące: <i>if</i> , <i>for</i> , <i>while</i> .
	3. Aspekty programowania funkcyjnego: funkcje, argumenty, zakresy nazw, specjalne tryby dopasowania argumentów.
	4. Przetwarzanie danych zawartych w plikach tekstowych, zapis obiektów Pythona w pliku: metoda <i>pickle</i> .
	5. Programowanie zorientowane obiektowo. Definicja klasy, konstruktor <i>__init__</i> , atrybut <i>self</i> . Narzędzia introspekcji klas: atrybuty <i>__dict__</i> , <i>__class__</i> , <i>__name__</i> . Atrybuty specjalne: <i>__add__</i> , <i>__str__</i> . Rola dziedziczenia i przeciążania.
	6. Tworzenie i obsługa bazy danych <i>shelve</i> , tworzenie konsolowego interfejsu bazy danych.

	7. Graficzny interfejs użytkownika (GUI) – moduł <i>tkinter</i> . Tworzenie graficznego interfejsu użytkownika (GUI) bazy danych <i>shelve</i> .
	8. Obsługa bazy danych SQL w języku Python. Nawiązywanie połączenia z bazami danych <i>SQL</i> , użycie modułu <i>sqlite3</i> i <i>mysql</i> . Tworzenie bazy i tabel. Wypełnianie tabel danymi.
	9. Realizacja przykładowej aplikacji bazodanowej SQL z wykorzystaniem interfejsu GUI.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X	X			
W02		X	X			
W03		X	X			
U01			X			
U02			X			
K01						X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% punktów z kolokwiów w trakcie zajęć i 50% punktów z teoretycznego testu egzaminacyjnego.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z kolokwiów w trakcie zajęć.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	12		18			h
2.	Inne (konsultacje, egzamin)	4		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	36					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,4					ECTS
5.	Liczba godzin samodzielnej pracy studenta	89					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	3,6					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	75					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	3					ECTS
9.	Sumaryczne obciążenie pracą studenta	125					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	5					ECTS

LITERATURA

1. Dawson M., *Python dla każdego. Podstawy programowania*, Helion, Gliwice 2014.
2. Lutz M., *Python. Wprowadzenie*, wydanie IV, Helion, Gliwice 2011.
3. <http://pl.python.org>
4. <https://docs.python.org/3/>
5. www.tu.kielce.pl/~stapor/materialy.htm