



Język Python na potrzeby GIS

Sylabus modułu zajęć

Informacje podstawowe

Kierunek studiów Geodezja i Kartografia	Cykl dydaktyczny 2020/2021
Specjalność Geoinformatyka, fotogrametria i teledetekcja	Kod przedmiotu DGIKGFS.IIi1S.c0fba39231b12ef5ea28d6087e911672.20
Jednostka organizacyjna Wydział Geodezji Górniczej i Inżynierii Środowiska	Języki wykładowe Polski
Poziom kształcenia studia magisterskie inżynierskie II stopnia	Obligatoryjność Obowiązkowy
Forma studiów Stacjonarne	Blok zajęciowy przedmioty specjalnościowe
Profil studiów Ogólnoakademicki	Przedmiot powiązany z badaniami naukowymi Tak
Koordynator przedmiotu	Mariusz Twardowski, Beata Hejmanowska
Prowadzący zajęcia	Mariusz Twardowski

Okres Semestr 1	Forma weryfikacji uzyskanych efektów uczenia się Zaliczenie	Liczba punktów ECTS 2.0
	Forma prowadzenia i godziny zajęć Wykład: 15 Ćwiczenia laboratoryjne: 15 Ćwiczenia projektowe: 10	

Cele kształcenia dla przedmiotu

C1	Moduł ma na celu zapoznanie studentów z możliwościami wykorzystania języka programowania Python do rozwiązywania zadań występujących przy przetwarzaniu danych występujących w GIS. Uświadamia ich o potencjalnych sposobach przyspieszenia pracy z danymi GIS przy użyciu własnych skryptów.
----	---

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	zasady obiektowego programowania w języku Python	GIK2A_W05, GIK2A_W06	Kolokwium
Umiejętności - Student potrafi:			
U1	tworzyć algorytmy i programy z interfejsem graficznym w języku Python	GIK2A_U11	Kolokwium
U2	rozszerzać funkcjonalność narzędzi geoinformatycznych z wykorzystaniem języka Python	GIK2A_U06, GIK2A_U11	Kolokwium
Kompetencji społecznych - Student jest gotów do:			
K1	kreatywnego podchodzenia do zagadnień programistycznych	GIK2A_K01	Aktywność na zajęciach

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Moduł pozwala na zapoznanie się z zasadami programowania w języku Python, używania jego wybranych bibliotek oraz projektowania i praktycznego wykonania interfejsu graficznego. Następnie uczy wykorzystania poznanych zasad do wykonania aplikacji graficznej z zastosowaniem programowania obiektowego.

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	15
Ćwiczenia laboratoryjne	15
Ćwiczenia projektowe	10
Przygotowanie do zajęć	5
Samodzielne studiowanie tematyki zajęć	9
Dodatkowe godziny kontaktowe	1
Przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	5
Łączny nakład pracy studenta	Liczba godzin 60
Liczba godzin kontaktowych	Liczba godzin 40

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	<p>Program laboratoria: 1. Wprowadzenie do języka Python Aktywne wykorzystanie interpretera do przypisywania zmiennych i ewaluacji wyrażeń matematycznych i logicznych. Sprawdzanie typów przypisanych zmiennych. Sposoby rozcinania ciągów. Metody alokacji list, zbiorów i słowników oraz ich odmiany. Wykorzystanie wybranych słów kluczowych.</p> <p>2. Skrypty i kontrola przepływu programu Zapoznanie z narzędziami do tworzenia skryptów i automatyzacja procesu ich uruchamiania. Interpretacja komunikatów o błędach, ich znajdowanie i poprawa. Wykorzystanie instrukcji warunkowej if...elif...else. Instrukcje pętli while oraz for, ich implementacja i kontrola. Obsługa wyjątków przy pomocy struktur try...except...else oraz with...as. Warianty definicji funkcji.</p> <p>3. Klasy, obiekty i moduły Tworzenie klas oraz implementacja ich metod. Przykłady dziedziczenia klasy bazowej. Inicjalizacja obiektów i odwołanie do ich metod z wykorzystaniem desygatora. Importowanie modułów własnych oraz standardowych.</p> <p>4. Wybrane biblioteki języka Przykłady najczęściej wykorzystywanych funkcji z biblioteki standardowej. Połączenie funkcjonalności bibliotek operacji przestrzennych w kodzie programu. Operacje na danych tekstowych, wektorowych i rastrowych.</p> <p>5. Projektowanie interfejsu QT Zapoznanie z narzędziem do projektowania interfejsu, dodawania elementów i modyfikacja ich zawartości. Wykorzystanie narzędzi do automatycznej konwersji pliku XML ze zdefiniowanym interfejsem do kodu Python. Przypisanie połączeń elementów interfejsu do definicji funkcji. Uruchamianie i testowanie interfejsu.</p>	U2, K1	Ćwiczenia laboratoryjne

2.	<p>Program wykładów: 1. Wprowadzenie do języka Python Zapoznanie się z interpreterem języka. Porównanie z innymi językami programowania. Przypisania zmiennych, operatory matematyczne i logiczne. Podstawowe typy danych i ich reprezentacja. Słowa kluczowe i podstawowe pojęcia z zakresu języka. Struktury złożone. Pojęcie obiektów niemutowanych.</p> <p>2. Skrypty i kontrola przepływu programu Formatowanie poprawnych skryptów i ich uruchamianie. Struktury warunkowe, zapętlenia i definicje funkcji kontrolujące przepływ programu. Różnice pomiędzy interpreterem a kompilatorem. Sposoby wykrywania wystąpień wyjątków w programie. Obsługa wejścia/wyjścia dla różnych urządzeń. Definiowanie funkcji różnych typów.</p> <p>3. Klasy, obiekty i moduły Pojęcie poziomów abstrakcji. Wprowadzenie do programowania obiektowego w języku Python. Budowa klas, obiektów i metod, ich dziedziczenie i polimorfizm. Sposoby modularyzacji skryptów i importowanie bibliotek.</p> <p>4. Wybrane biblioteki języka Omówienie ważniejszych elementów bibliotek standardowych. Biblioteki zawierające przekształcenia geospacialne GDAL, OGR, OSR. Sposoby odczytywania, zapisywania i przetwarzania danych rastrowych i wektorowych.</p> <p>5. Projektowanie interfejsów Wykorzystanie narzędzi do szybkiego projektowania aplikacji (RAD), automatyczne generowanie kodu programu, połączenia elementu interfejsu z kodem wykonawczym. Wprowadzenie pojęcia programu sterowanego przez wydarzenia. Zapoznanie z bibliotekami do budowy interfejsu graficznego.</p>	W1	Wykład
3.	<p>Program ćwiczeń projektowych: 1. Budowa aplikacji QT Zapoznanie ze sposobami budowania i uruchamiania aplikacji. Mechanizmy komunikacji API QT z metodami. Tworzenie klasy kompatybilnej z interfejsem Wykorzystanie poznanych elementów języka do rozbudowy funkcjonalności interfejsu poprzez połączenie metod klasy z interfejsem. Tworzenie metod do zarządzania danymi różnych typów.</p> <p>2. Budowa aplikacji WSGI Konfiguracja środowiska aplikacji Python uruchomianej przez WWW. Wykorzystanie wybranych narzędzi w procesie projektowania i implementacji aplikacji WSGI. Wdrożenie aplikacji online na serwerze prywatnym lub publicznym. Metody debugowania aplikacji w cyklu deweloperskim.</p> <p>3. Programowanie w zespole Narzędzia i usługi wspomagania programowania w zespole Konfiguracja środowiska do pracy w zespole. Wersjonowanie źródeł. Projektowanie API i budowa aplikacji wielomodułowej z wykorzystaniem bibliotek GIS. Zastosowanie najlepszych praktyk w programowaniu zespołowym.</p>	U1, U2	Ćwiczenia projektowe

Informacje rozszerzone

Metody i techniki kształcenia:

Wykład tablicowy, Wykonanie ćwiczeń laboratoryjnych, Prezentacja multimedialna, Projekt, Zdalne kształcenie

Rodzaj zajęć	Sposób weryfikacji i oceny efektów uczenia się	Warunki zaliczenia przedmiotu
Wykład	Kolokwium	
Ćwiczenia laboratoryjne	Aktywność na zajęciach, Kolokwium	
Ćwiczenia projektowe	Kolokwium	

Dodatkowy opis

W celu usprawnienia zajęć, przy użyciu języka Python i biblioteki Qt napisany został program służący do konwersji współrzędnych zapisanych w formacie Krakowskiego Układu Lokalnego na format EPSG2180, oraz generowania pliku w formacie SHP, który może zostać wyświetlony w programach typu GIS. Studenci podczas zajęć projektowych mogą użyć wygenerowanych współrzędnych do wykonania własnych opracowań.

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Ćwiczenia projektowe i laboratoryjne zaliczane są na podstawie kolokwium. W przypadku braku zaliczenia kolokwium możliwa będzie jego poprawa.

Sposób obliczania oceny końcowej

Ocena końcowa jest wystawiana na podstawie kolokwium zaliczeniowego.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Student w przypadku nieobecności uzupełnia braki samodzielnie.

Wymagania wstępne i dodatkowe

Znajomość modułu podstaw informatyki.

Znajomość obsługi komputera.

Znajomość obsługi edytora tekstowego. Na pierwszych zajęciach (ćwiczeniach) prowadzący określi:

- dopuszczalną liczbę nieobecności studenta na zajęciach oraz sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności,

- zasady uzyskania zaliczenia w terminie podstawowym.

Przed końcem semestru prowadzący ustali zasady i terminy uzyskania zaliczeń poprawkowych. Student może dwukrotnie przystąpić do poprawkowego zaliczenia zajęć.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

Ćwiczenia projektowe: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez sylabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

Literatura

Obowiązkowa

1. Oficjalna dokumentacja języka i kursy online: <http://pl.python.org>
2. Dokumentacja PyQt: <http://pyqt.sourceforge.net/Docs/PyQt5/>

Dodatkowa

1. Lutz Mark: "Python. Wprowadzenie". Helion 2011,
2. Dawson Michael: "Python dla każdego.Podstawy programowania". Helion 2014
3. Summerfield Mark: "Rapid GUI Programming with Python and Qt". Prentice Hall 2008

Badania i publikacje

Badania

1. Integracja danych teledetekcyjnych na potrzeby kontroli w systemie dopłat bezpośrednich do rolnictwa (IACS)
http://home.agh.edu.pl/~galia/research_bh.html

Publikacje

1. K. Pyka, M. Twardowski: "Miejsce wolnego oprogramowania w nauczaniu geoinformatyki". Archiwum Fotogrametrii, Kartografii i Teledetekcji. 2007.
2. K.Pyka, M. Słota, M. Twardowski -"Usage of stereo orthoimage in GIS: old concept, modern solution". XXII ISPRS congress. 2012
3. Twardowski M., Pastucha E., Kolecki J., 2016: Performance of the automatic bundle adjustment in the virtualized environment

Kierunkowe efekty uczenia się

Kod	Treść
GIK2A_K01	działania w sposób kreatywny i przedsiębiorczy z uwzględnieniem krytycznej oceny posiadanej wiedzy i potrzeby konsultacji eksperckich
GIK2A_U06	posługiwać się technikami informacyjno-komunikacyjnymi właściwymi do realizacji zadań typowych dla działalności inżynierskiej,
GIK2A_U11	zapisywać obiekty świata rzeczywistego w systemie informacji przestrzennej oraz tworzyć i realizować procedury postępowania w języku formalnym za pomocą złożonych narzędzi programowych, dokonując krytycznej analizy stosowanych metod, uwzględniając komponent badawczy i aspekty pozatechniczne
GIK2A_W05	metody, techniki, narzędzia i materiały stosowane przy rozwiązywaniu złożonych zadań inżynierskich z dziedziny geodezja i kartografia oraz dziedzin pokrewnych
GIK2A_W06	specjalistyczne metody pozyskiwania, analizowania, modelowania i wizualizowania danych przestrzennych i zmian tych danych spowodowanych procesami naturalnymi i technologicznymi