



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie obiektowe [S2Elmob1-SSP>PO1]

Przedmiot

Kierunek studiów

Elektromobilność

Rok/Semestr

1/2

Studia w zakresie (specjalność)

Samochodowe systemy pokładowe

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

0

Laboratorium

15

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

1,00

Koordynatorzy

dr inż. Stanisław Mikulski

stanislaw.mikulski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Znajomość podstaw programowania w dowolnym języku programowania, w tym zagadnienia takie jak: pętle, instrukcje warunkowe, typy zmiennych, funkcje oraz ich definicje. Podstawowa wiedza z matematyki dot. zagadnień: aproksymacji, rachunku macierzowego oraz analizy statystycznej.

Cel przedmiotu

Celem przedmiotu jest poszerzenie wiedzy dotyczącej techniki programowania obiektowego, w taki sposób aby student potrafił tworzyć programy obiektowe charakteryzujące się dużą zwinnością i efektywnością, wykorzystując przy tym już opracowane pakiety języka Python.

Przedmiotowe efekty uczenia się

Wiedza:

Student ma poszerzoną wiedzę w zakresie technik programowania oraz stosowania nowoczesnych narzędzi informatycznych do analizy i syntezy układów elektrycznych pojazdów hybrydowych i elektrycznych w tym trakcyjnych

Umiejętności:

Student potrafi stosować nowoczesne narzędzia informacyjno-komunikacyjne, zaawansowane techniki programowania oraz metody uczenia maszynowego w celu gromadzenia, przetwarzania i analizy danych

Kompetencje społeczne:

Student rozumie, że w obszarze techniki wiedza i umiejętności szybko się dewaluują co wymaga ciągłego ich uzupełniania

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Laboratorium:

poprzez zadania wykonywane samodzielnie podczas zajęć.

Treści programowe

Laboratorium:

Paradygmaty programowania, mechanizmy wykonywania kodu, interpretacja a kompilacja kodu, omówienie podstawowej struktury języka Python, w tym m. in.: klasy, obiekty, abstrakcja, dziedziczenie i hermetyzacja. Omówienie wybranych wzorców projektowych i ich przykładowa implementacja. Diagramy klas UML.

Tematyka zajęć

W ramach laboratoriów omawiane są następujące zagadnienia:

- 1) wprowadzenie do języka Python oraz omówienie środowiska Visual Studio Code
- 2) typy danych oraz operatory stosowane w języku Python
- 3) tworzenie klas
- 4) stosowanie hermetyzacji (definicja właściwości klasy, oraz funkcji dostępowych)
- 5) dziedziczenie
- 6) klasy abstrakcyjne
- 7) tworzenie kodu na podstawie diagramów UML

Metody dydaktyczne

Laboratorium:

Prezentacje, quizy i zadania e-learningowe, inicjowanie dyskusji w trakcie zajęć laboratoryjnych.

Literatura

Podstawowa:

1. Johansson R., Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion 2021
2. Dokumentacja programu Python, <https://docs.python.org/3/>
3. Lott S. F., Phillips D., Programowanie zorientowane obiektowo w Pythonie. Tworzenie solidnych i łatwych w utrzymaniu aplikacji i bibliotek. Wydanie IV, Helion 2023

Uzupełniająca:

1. Chollet F., Deep learning. Praca z językiem Python i biblioteką Keras, Helion 2019

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	29	1,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	15	0,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	14	0,50