

<p>1. ma rozszerzoną i pogłębioną wiedzę z matematyki przydatną do formułowania i rozwiązywania złożonych zadań informatycznych dotyczących m. in. programowania w logice, formalnej specyfikacji i weryfikacji oprogramowania - [K1st_W1]</p> <p>2. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie kluczowych zagadnień informatyki oraz wiedzę szczegółową w zakresie wybranych zagadnień tej dyscypliny - [K1st_W4]</p> <p>3. zna podstawowe techniki, metody oraz narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych, głównie o charakterze inżynierskim - [K1st_W7]</p>
<p>Umiejętności:</p>
<p>1. potrafi, formułując i rozwiązując zadania informatyczne, zastosować odpowiednio dobrane metody, w tym metody analityczne, symulacyjne lub eksperymentalne - [K1st_U4]</p> <p>2. ma umiejętności formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi - [K1st_U11]</p> <p>3. potrafi organizować, współdziałać i pracować w grupie, przyjmując w niej różne role oraz potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K1st_U18]</p>
<p>Kompetencje społeczne:</p>
<p>1. na podstawie historii rozwoju języka Python rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K1st_K1]</p> <p>2. ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych - [K1st_K2]</p>

<p style="text-align: center;">Sposoby sprawdzenia efektów kształcenia</p>
<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:</p> <p>Ocena formująca:</p> <ul style="list-style-type: none">- na podstawie odpowiedzi udzielanych odnośnie realizacji zadań w ramach ćwiczeń; <p>Ocena podsumowująca:</p> <p>Sprawdzanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- ocenę umiejętności związanych z realizacją zadań laboratoryjnych (w ciągu semestru każdy student ma do napisania pięć programów o stopniowo wzrastającej skali trudności, w tym związanego z utworzeniem komponentu i jego wykorzystaniem),- ocenę wiedzy i umiejętności związanych z realizacją ćwiczeń laboratoryjnych poprzez sprawdziany,- ocenę wiedzy i umiejętności wykazanych na pisemnym kolokwium zaliczeniowym o charakterze problemowym i praktycznym (kolokwium składa się z 6 tematów/zadań o różnej skali trudności i wynikającej stąd różnej punktacji; maksymalnie można uzyskać 20 punktów; na ocenę 3.0 trzeba zdobyć 11 punktów). <p>Studenci, którzy wyróżniająco wykonali programistyczne zadania laboratoryjne oraz uzyskali bardzo dobre wyniki ze sprawdzianów częściowych są zwalniani z kolokwium zaliczeniowego.</p>
<p style="text-align: center;">Treści programowe</p>

Program przedmiotu obejmuje następujące zagadnienia:

- podstawowe pojęcia związane z programowaniem (programowanie, algorytm, program, język programowania, język ukierunkowany maszynowo, rozkaz, język wyższego rzędu, język uniwersalny, język specjalizowany),
- przegląd języków programowania (Ada, Algol, asemblery, Basic, C, C++, Cobol, Fortran, HTML, Java, Lisp, Logo, Pascal, PHP, PL/1, Prolog),
- sieci działań (schematy blokowe) i symbole stosowane w nich,
- ogólne zasady programowania zorientowanego obiektowo (dziedziczność, hermetyczność i polimorfizm),
- ogólna charakterystyka pakietu PyCharm,
- podstawowe pojęcia związane z konstruowaniem programów w zintegrowanym systemie programowania PyCharm (projekt, pakiet, moduł),
- posługiwanie się zintegrowanym pakietem programowania PyCharm,
- przegląd konstrukcji języka Python (program, moduł, pakiet, funkcje, klasy i obiekty, typy danych, zmienne, instrukcje),
- struktura programu i pakietu,
- podstawowe elementy języka (symbole podstawowe, słowa kluczowe i dyrektywy języka, identyfikatory, liczby, łańcuchy, w tym łańcuchy znaków Unicode, literały logiczne, komentarze i separatory),
- typy danych i ich opis (definiowanie typów, typy proste, łańcuchowe, opisujące obiekty, zgodność typów, dekoratory),
- zmienne (deklaracje zmiennych, zmienne indeksowane, obiektowe, dynamiczne, funkcyjne, z początkową wartością, nakładanie zmiennych, literały stałe i zmienne),
- wyrażenia (rodzaje operatorów i ich priorytet, składnia wyrażenia, wyrażenie stałe),
- instrukcje (proste, strukturalne, wyrażenia lambda),
- funkcje (definicje, rodzaje parametrów, przeciążanie, wywoływanie),
- przetwarzanie obiektów (konstruktory i destruktory, metody statyczne, wirtualne, dynamiczne i abstrakcyjne, obsługa wiadomości, własności),
- publikowanie pakietów w Python Package Index,
- przetwarzanie plików, serializacja,
- wielowątkowość (synchronizacja wątków, priorytety, oczekiwanie na zakończenie).

Na zajęciach laboratoryjnych studenci, po zapoznaniu się ze zintegrowanym środowiskiem programowania PyCharm, piszą programy wykorzystujące poznane elementy języka.

Metody dydaktyczne:

1. Wykład: prezentacja multimedialna (każdy wykład) oraz prezentacja pisania i wykonywania wybranych programów bezpośrednio w pakiecie PyCharm.
2. Ćwiczenia laboratoryjne: ćwiczenia praktyczne dotyczące elementów języka Python, pisanie programów w tym języku.

Literatura podstawowa:

1. Python 3: kompletne wprowadzenie do programowania, Mark Summerfield, Helion, 2010

Literatura uzupełniająca:

1. Python. Wprowadzenie. Wydanie IV, Mark Lutz, Helion 2010

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. Udział w zajęciach laboratoryjnych : 3 x 4 godz.	12
2. Przygotowanie do ćwiczeń laboratoryjnych: 3 x 8 godz.	24
3. Udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	3
4. Napisanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	11
5. Przygotowanie do sprawdzianów	5
6. Udział w wykładach	12
7. Zapoznanie się ze wskazaną literaturą, 300 stron	30
8. Przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym (28 + 2 godz.)	30

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	127	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	27	2
Zajęcia o charakterze praktycznym	47	3