



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Informatyka II [S1EiT1>INF3]

Przedmiot

| | |
|---|--|
| Kierunek studiów Elektronika i telekomunikacja | Rok/Semestr 2/4 |
| Studia w zakresie (specjalność) – | Profil studiów ogólnoakademicki |
| Poziom studiów pierwszego stopnia | Język oferowanego przedmiotu polski |
| Forma studiów stacjonarne | Wymagalność obligatoryjny |

Liczba godzin

| | | |
|----------------|-------------------------|------------------------|
| Wykład 30 | Laboratorium 30 | Inne (np. online) 0 |
| Ćwiczenia 0 | Projekty/seminaria 0 | |

Liczba punktów ECTS

5,00

Koordynatorzy

prof. dr hab. inż. Grzegorz Danilewicz
grzegorz.danilewicz@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student przystępując do przedmiotu ma podstawową wiedzę o budowie komputera oraz wiedzę z zakresu systemów liczbowych. Potrafi się posługiwać językami programowania wysokiego poziomu C, C++. Potrafi pozyskiwać informacje z literatury i baz danych oraz innych źródeł w języku polskim lub angielskim; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, wyciągać wnioski i uzasadniać opinie.

Cel przedmiotu

Zapoznanie studentów z podstawowymi koncepcjami programowania obiektowego, w tym klasami, obiektami, dziedziczeniem, polimorfizmem i hermetyzacją. Rozwijanie biegłości w korzystaniu z podstawowych bibliotek Pythona do różnych zadań. Porównanie i zestawienie paradygmatów programowania obiektowego i funkcjonalnego na praktycznych przykładach w języku Python, z podkreśleniem ich mocnych i słabych stron w różnych kontekstach rozwiązywania problemów.

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma podstawową wiedzę o trendach rozwojowych w zakresie: języków programowania wysokiego poziomu wchodzących w skład platformy .NET, organizacji współczesnych platform programistycznych

oraz integracji języków programowania.

2. Zna język programowania wysokiego poziomu Python. Zna zasady konstrukcji programów komputerowych z rozbudowanym interfejsem graficznym, tworzenia oprogramowania wielowątkowego, współpracy z bazami danych oraz podstawowych mechanizmów związanych z programowaniem aplikacji sieciowych z wykorzystaniem języka Python.

Umiejętności:

1. Potrafi posługiwać się językiem programowania wysokiego poziomu Python; potrafi projektować i programować aplikacje z rozbudowanym interfejsem graficznym; umie tworzyć oprogramowanie implementujące podstawowe protokoły sieciowe i współpracujące z podstawowymi dostawcami baz danych; potrafi projektować i tworzyć oprogramowanie wielowątkowe.
2. Potrafi rozwiązać typowe inżynierskie zadania programistyczne z wykorzystaniem języka Python.

Kompetencje społeczne:

1. Zna ograniczenia własnej wiedzy i umiejętności w zakresie współczesnych języków programowania wysokiego poziomu, rozumie konieczność dalszego kształcenia się w zakresie języków i platform programistycznych
2. Posiada świadomość konieczności profesjonalnego podejścia do rozwiązywanych problemów technicznych z wykorzystaniem współczesnych języków i platform programistycznych; jest świadomy podejmowanej odpowiedzialności za tworzone przez siebie oprogramowanie.
3. Jest świadomy niebezpieczeństw wynikających z błędnie zaprojektowanego i tworzonego oprogramowania, zarówno dla użytkowników, jak i dla urządzeń.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykłady. Wiedza sprawdzana na dwóch kolokwium. Pisemne sprawdziany dotyczące treści wykładowych - jedno w połowie wykładów, drugie na zakończenie wykładów. Każde z kolokwium musi być zaliczone na co najmniej ocenę dostateczną. Ocena dostateczna jest wystawiana, gdy liczba punktów z kolokwium jest większa niż 50%. Ocena końcowa jest średnią arytmetyczną ocen z kolokwium. Dla ocen z kolokwium i dla oceny końcowej obowiązuje skala ocen od 2 (niedostateczny - ocena negatywna) do 5 (bardzo dobry). Zasady przeliczania oceny końcowej:

Zakres średniej : ocena

- 0,00 - 2,99 : 2,0
- 3,00 - 3,24 : 3,0
- 3,24 - 3,74 : 3,5
- 3,75 - 4,24 : 4,0
- 4,25 - 4,74 : 4,5
- 4,75 - 5,00 : 5,0

Laboratoria. Ocenie podlegają następujące składowe: wiedza studenta przed wykonaniem ćwiczenia, odpowiedzi na pytania w trakcie realizacji ćwiczeń, sprawozdania pisemne z realizacji ćwiczeń, pisemne kolokwium na koniec semestru. Ocena końcowa z laboratorium jest wypadkową ocen składowych, przy czym każda z ocen składowych musi być pozytywna. Dla ocen składowych i dla oceny końcowej obowiązuje skala ocen od 2 (niedostateczny - ocena negatywna) do 5 (bardzo dobry).

Treści programowe

Treści programowe obejmują następujące zagadnienia:

1. Kompilacja a interpretacja kodu
2. Programowanie strukturalne a obiektowe
3. Python jako przykład języka badań i prototypowania
4. Podstawy programowania w języku Python

Tematyka zajęć

1. Wprowadzenie do programowania w języku Python
2. Omówienie środowiska programistycznego
3. Komentowanie kodu
4. Proste typy danych w Python: integers, floats, booleans
5. Typ łańcucha znaków, jego formatowanie, metody

6. Wykorzystanie bibliotek
 7. Definiowanie i wykorzystanie funkcji: wywoływanie, parametry, wartości zwracane
 8. Funkcje anonimowe (lambda)
 9. Parametry wywołania funkcji
 10. Komunikacja z programem
 11. Literały liczbowe
 12. Instrukcje warunkowe i operatory logiczne
 13. Instrukcje pętli
 14. Struktury złożone: listy, kolejki, słowniki, zbiory
 15. Obsługa plików tekstowych i binarnych
 16. Przekazywanie parametrów przez wartości i referencję
 17. Obsługa błędów
 18. Podstawy programowania obiektowego w Python: klasy, obiekty, dziedziczenie, wielopostaciowość
 19. Tworzenie i wykorzystanie własnych bibliotek
 20. Formaty wymiany danych (JSON, XML)
 21. Podstawy tworzenia wykresów (Matplotlib)
 22. Zaawansowane zagadnienia w Python
- Laboratorium: Praktyka programowania w języku Python.

Metody dydaktyczne

1. Wykład

Wykład z użyciem tablicy i/lub projektora, wykład z użyciem notatnika Jupyter, wykład konwersatoryjny, eksperyment, studium przypadku, pisanie oprogramowania.

2. Ćwiczenia

wykonywanie zadań podanych przez prowadzącego (ćwiczenia praktyczne realizowane z wykorzystaniem środowisk uruchomieniowych dla języków programowania Python) uzupełniane prezentacjami multimedialnymi.

Literatura

Podstawowa

1. Jesse Liberty "Programowanie C#", Helion 2005
 2. Charles R. Severance, Python dla wszystkich, Andrzej Wójtowicz, 2021
- Uzupełniająca
3. www.python.org

Bilans nakładu pracy przeciętnego studenta

| | Godzin | ECTS |
|--|--------|------|
| Łączny nakład pracy | 125 | 5,00 |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem | 70 | 3,00 |
| Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) | 55 | 2,00 |