



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Informatyka II

### Przedmiot

Kierunek studiów

Elektronika i Telekomunikacja

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/4

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

30

Ćwiczenia

Laboratoria

30

Projekty/seminaria

Inne (np. online)

### Liczba punktów ECTS

5

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

prof. dr hab. inż. Grzegorz Danilewicz,

grzegorz.danilewicz@put.poznan.pl

Odpowiedzialny za przedmiot/wykładowca:

### Wymagania wstępne

Posiada wiedzę z zakresu informatyki i zna składnię języków oprogramowania C, C++, umiejętność samodzielnego kształcenia się, ma świadomość konieczności poszerzania swoich kompetencji

### Cel przedmiotu

Zapoznanie studentów z zasadami programowania obiektowego i wykorzystania bibliotek środowiska .NET.

### 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 C#. 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 C#.



### Umiejętności

1. Potrafi się posługiwać językiem programowania wysokiego poziomu C#; 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 C#.

### 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

Wprowadzenie do programowania w języku C#, omówienie środowiska programistycznego. Platforma .NET. Typy, tablice i kontrola przepływu. Klasy i filary programowania obiektowego. Przeciążanie operatorów. Dziedziczenie i interfejsy. Delegaty i obsługa zdarzeń. Obsługa wyjątków. Struktury danych i typy generyczne. Programowanie wielowątkowe.

### Metody dydaktyczne

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

### Literatura

Podstawowa

1. Jesse Liberty "Programowanie C#", Helion 2005
2. <https://docs.microsoft.com/en-gb/dotnet/csharp/>

Uzupełniająca

- <https://docs.microsoft.com/en-us/dotnet/core/>
- <https://docs.microsoft.com/en-us/dotnet/architecture/>
- <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>
- <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>

### Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
łącznie nakład pracy	125	5,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	70	3,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwium) <sup>1</sup>	55	2,0

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności