



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie obiektowe

### Przedmiot

Kierunek studiów

Elektrotechnika

Studia w zakresie (specjalność)

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

0

Laboratoria

30

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

2

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

mgr inż. Damian Burzyński

Odpowiedzialny za przedmiot/wykładowca:

email: damian.burzynski@put.poznan.pl

tel. 61 665 24 49

Instytut Elektrotechniki i Elektroniki

Przemysłowej

ul. Piotrowo 3A, 60-965 Poznań

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu informatyki i programowania. Dodatkowo powinien również posiadać umiejętności myślenia abstrakcyjnego, pozyskiwania informacji ze wskazanych źródeł oraz wykazywać gotowość do podjęcia współpracy w zespole.

### Cel przedmiotu

Przekazanie studentom teoretycznych i praktycznych zagadnień programowania wysokopoziomowego z elementami programowania obiektowego, nabycie umiejętności tworzenia aplikacji w środowisku



Microsoft Visual Studio (w języku C#). Rozwijanie u studentów umiejętności rozwiązywania problemów występujących podczas projektowania i implementacji aplikacji technicznych.

### Przedmiotowe efekty uczenia się

#### Wiedza

Student zna zasady programowania wysokopoziomowego. Posiada wiedzę z zakresu programowania obiektowego przydatną podczas tworzenia aplikacji technicznych.

#### Umiejętności

Student umie stosować narzędzia służące do programowania z wykorzystaniem elementów programowania obiektowego. Potrafi programować w języku C#.

#### Kompetencje społeczne

Świadomość szybkości rozwoju technologii informatycznych i związanej z tym konieczności aktualizacji wiedzy. Potrafi samodzielnie wyszukiwać informacje w literaturze i Internecie, także w językach obcych.

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Umiejętności nabyte w ramach zajęć laboratoryjnych weryfikowane są na podstawie kolokwium zaliczeniowego lub wykonanej przez siebie aplikacji projektowej z uwzględnieniem aktywności indywidualnej na zajęciach. Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć (szczególnie za: omówienie dodatkowych aspektów zagadnienia; efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu; umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium; uwagi związane z udoskonaleniem materiałów dydaktycznych; staranność)

### Treści programowe

Implementacja w środowisko Visual Studio C# Express Edition przedstawionych na wykładzie zagadnień w praktycznych programach obiektowego wykorzystujących następujące elementy programowania: deklaracje typów obiektowych, pola i metody, pola typu readonly, elementy statyczne, konstruktory i destruktor, właściwości, przeciążenie metod, przeciążenia operatorów, hermetyzacja, dziedziczenie, polimorfizm i jego zastosowanie, klasy i metody abstrakcyjne, wykorzystanie kolekcji różnych typów, wykorzystanie delegat oraz interfejsów, programowanie aplikacji wykorzystującej wiele wątków, implementacja wybranych wzorców projektowych.

### Metody dydaktyczne

Praca indywidualna w laboratorium komputerowym, polegająca na implementacji zadanych funkcji i programów komputerowych, dyskusja, demonstracje.

### Literatura

#### Podstawowa

1. Troelsen, Andrew; Japikse, Phiplip. Język C# 6.0 i platforma .NET 4.6. Red. . Warszawa: Wydawnictwo Naukowe PWN, 2017, 1465 s. ISBN 978-83-01-19832-9



2. Chłosta, Paweł. Aplikacje Windows Forms .Net w C#. Red. . Warszawa: Wydawnictwo Naukowe PWN, 2006, 318 s. ISBN 83-01-14830-6
3. Rodenburg, Jort. Koduj jak profesjonalista C#. Red. . Warszawa: Promise, 2022, 466 s. ISBN 978-83-7541-459-2
4. Trey Nash, Accelerated C# 2010, Apress, 2010
5. C# language specification: <https://github.com/dotnet/csharp-lang/tree/main/spec>
6. Internet

Uzupełniająca

1. Sharp, John. Microsoft Visual C# 2017 Krok po kroku. Red. . Warszawa: Promise, 2019, 937 s. ISBN 978-83-7541-362-5
2. Stasiak, Andrzej; Dąbrowski, Włodzimierz; Wolski, Michał. Modelowanie systemów informatycznych w języku UML 2.1. Red. . Warszawa: Wydawnictwo Naukowe PWN, 2009, 196 s. ISBN 978-83-01-21108-0

**Bilans nakładu pracy przeciętnego studenta**

	Godzin	ECTS
Łączny nakład pracy	50	2,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,5
Praca własna studenta (studia literaturowe, przygotowanie do kolokwium zaliczeniowego lub wykonanie aplikacji projektowej) <sup>1</sup>	20	0,5

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