

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2023-2027

Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>programowanie obiektowe 2</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia inżynierskie I-go stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok II, semestr 4</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Wojciech Koziół</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Wojciech Koziół, mgr inż. Ewa Żesławska, mgr inż. Wojciech Gałka</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
4	15			30					4

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Dobra umiejętność programowania obiektowego w języku Java, w tym tworzenia prostych interfejsów graficznych

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Zapoznanie studentów z zagadnieniami dotyczącymi programowania w języku C#. Porównanie języków programowania Java i C#, przedstawienie podobieństw i różnic.
C ₂	Zapoznanie studentów z językiem C# i środowiskiem Visual Studio.
C ₃	Zapoznanie studentów z zagadnieniami dotyczącymi tworzenia graficznych interfejsów użytkownika i dostępu do relacyjnej bazy danych oraz nabycie umiejętności tworzenia takich aplikacji w języku C#.
C ₄	Nabycie przez studentów kompetencji w zakresie znajomości i wykorzystania kolejnego popularnego i ważnego języka programowania opartego o paradygmat obiektowy. Rozszerzenie warsztatu technologicznego.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu Student:	Odniesienie do efektów kierunkowych
EK_01	Zna podstawowe konstrukcje programistyczne i struktury danych występujące w języku C#, w tym również, zna dobrze zagadnienia z zakresu programowania obiektowego oraz wie, jak je wykorzystać i zaimplementować w języku C#.	K_Wo6
EK_02	Ma podstawową wiedzę na temat tworzenia graficznych interfejsów użytkownika oraz interfejsów do łączenia się z relacyjnymi bazami danych i użycia ich w języku C#.	K_Wo6
EK_03	Umie zastosować poznane konstrukcje i struktury danych, w tym również kolekcje w języku C#. Potrafi również wykorzystać odpowiednie biblioteki i interfejsy języka C# do tworzenia oprogramowania w paradygmacie obiektywnym, w tym również biblioteki do tworzenia graficznych interfejsów użytkownika oraz do łączenia się z relacyjnymi bazami danych.	K_Uo7
EK_04	Potrafi zaprojektować i utworzyć aplikację w języku C#, która umożliwia połączenie się z relacyjną bazą danych i wykorzystującą tę bazę. Utworzona aplikacja posiada odpowiedni do potrzeb interfejs użytkownika.	K_Uo7

3.3 Treści programowe**A. Problematyka wykładu**

Wprowadzenie do platformy .NET i języka C#. Porównanie technologii .NET i Java. Obsługa środowiska Microsoft Visual Studio .NET.
Podstawowe typy danych, wyrażenia, instrukcje, tablice jedno i wielowymiarowe, wskaźniki ,

struktury w języku C#.
Programowanie obiektowe w C#: tworzenie klas i obiektów, pola, właściwości, metody, hermetyzacja, dziedziczenie, abstrakcja, polimorfizm.
Obsługa wyjątków w języku C#.
Operacje na strumieniach w C#.
Kolekcje w języku C#, wstęp do technologii LINQ.
Wprowadzenie do Windows Forms .NET.
Wprowadzenie do technologii Entity Framework.

B. Problematyka ćwiczeń laboratoryjnych

Wprowadzenie do środowiska Visual Studio, tworzenie i uruchamianie prostych programów.
Typy danych, wyrażenia warunkowe i instrukcje sterujące, wskaźniki w języku C#.
Tworzenie klas, obiektów i struktur w języku C#.
Hermetyzacja, dziedziczenie i kompozycja w języku C#.
Polimorfizm i abstrakcja (klasy abstrakcyjne i interfejsy) w języku C#.
Obsługa wyjątków w języku C#.
Operacje na strumieniach w języku C#.
Obsługa kolekcji w języku C#, użycie technologii LINQ.
Tworzenie graficznego interfejsu użytkownika przy użyciu Windows Forms .NET.
Tworzenie prostych aplikacji używających bazy danych przy użyciu technologii Entity Framework.
Tworzenie projektu aplikacji na zaliczenie z interfejsem graficznym i bazą danych przy użyciu poznanych wcześniej technologii.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną.

Laboratoria: tworzenie programów komputerowych w oparciu o treści zadań zawartych w konspektach laboratoryjnych, projekt praktyczny w postaci oprogramowania realizowanego głównie jako praca domowa w dużej mierze samodzielna, ale konsultowana.

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	test zaliczeniowy z treści prezentowanej na wykładzie	wykład
EK_02	test zaliczeniowy z treści prezentowanej na wykładzie	wykład
EK_03	kolokwium i projekt	laboratoria
EK_04	projekt	laboratoria

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

LABORATORIUM:

Zaliczenie kolokwium sprawdzającego znajomość języka C# i jego składni na ocenę pozytywną. Weryfikowany jest: EK_03 oraz pośrednio EK_01.

Zaliczenie projektu na ocenę pozytywną. Weryfikowane są: EK_03, EK_04 oraz pośrednio EK_01, EK_02.

Warunki zaliczenia dla efektów EK_03 w ramach kolokwium:

Prowadzący laboratorium sam ustala zakres zadań dla kolokwium i kryteria jego oceny.

Warunki zaliczenia dla efektu EK_04 oraz EK_03 w ramach projektu:

Dostateczny:

Student potrafi tworzyć oprogramowanie z graficznym interfejsem użytkownika umożliwiające łączenie się z relacyjnymi bazami danych. Tworzone oprogramowanie jest realizowane w ramach większego projektu wykonywanego poza zajęciami. Student łączy się z utworzoną przez siebie bazą danych (co najmniej dwie tabele połączone relacją) poprzez utworzony przez siebie prosty interfejs graficzny. Student tworzy również bardzo prostą dokumentację do projektu.

Dobry:

Student potrafi tworzyć oprogramowanie z graficznym interfejsem użytkownika umożliwiające łączenie się z relacyjnymi bazami danych. Tworzone oprogramowanie jest realizowane w ramach większego projektu wykonywanego poza zajęciami. Student łączy się z bardziej złożoną relacyjną bazą danych niż na ocenę dostateczny (co najmniej trzy tabele połączone relacjami). Student sam projektuje i tworzy bazę danych. Obsługa bardziej złożonej bazy danych wymusza utworzenie bardziej złożonego interfejsu użytkownika. Utworzony interfejs użytkownika jest dobrze przemyślany. Student tworzy również prostą dokumentację projektu, w tym również instrukcję obsługi aplikacji.

Bardzo dobry:

Student potrafi tworzyć oprogramowanie z graficznym interfejsem użytkownika umożliwiające łączenie się z relacyjnymi bazami danych. Tworzone oprogramowanie jest realizowane w ramach większego projektu wykonywanego poza zajęciami. Student łączy się z bardziej złożoną relacyjną bazą danych (jej schemat zawiera przynajmniej cztery tabele połączone relacjami), którą sam zaprojektował i utworzył. Stworzony przez studenta graficzny interfejs aplikacji jest bardzo dobrze przemyślany - cechuje się przejrzystością i intuicyjnością obsługi. Student tworzy do projektu bardziej szczegółową dokumentację i szczegółową instrukcję obsługi utworzonej przez siebie aplikacji.

Ocena końcowa jest średnią ocen uzyskanych dla poszczególnych efektów lub obliczana jest według indywidualnego algorytmu ustalonego przez prowadzącego zajęcia laboratoryjne. Na ocenę mogą wpływać również obserwacje studentów prowadzone przez nauczyciela prowadzącego podczas zajęć.

Ocena końcowa jest średnią ważoną kolokwium z wagą 1 i projektu z wagą 3, przy czym wymagane jest aby i kolokwium i projekt zaliczone były na ocenę pozytywną.

WYKŁAD:

1. Pozytywny wynik testu z zakresu materiału prezentowanego na wykładzie, przy czym:

- Student otrzymuje ocenę pozytywną z testu jeśli uzyska co najmniej 60% pozytywnych odpowiedzi z zadań dotyczących efektu EK_01 i co najmniej 60% pozytywnych odpowiedzi z zadań dotyczących efektu EK_02.

Uwaga: Student przystępujący do testu poprawkowego jest zobowiązany do poprawy tylko tych efektów, których nie zaliczył w terminie podstawowym.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	45
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, udział w teście)	3
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, przygotowanie projektu zaliczeniowego)	55
SUMA GODZIN	103
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1. J. Matulewski: C# : lekcje programowania : praktyczna nauka programowania dla platform .NET i .NET Core, Gliwice, Helion 2021 2. Troelsen ; tł. [z ang.] Maciej Baranowski: Język C# 2010 i platforma .NET 4 , Warszawa, Wydawnictwo Naukowe PWN, 2011 3. M. Lis: C# : praktyczny kurs, Gliwice, Helion, 2021 4. Ł. Sosna: Visual Studio 2022 : C# i .NET : programowanie kontrolek, Gliwice, Helion, 2023 5. https://www.entityframeworktutorial.net/
Literatura uzupełniająca: 1. I.Griffiths ; tł. Piotr Rajca: C# 5.0. Programowanie : tworzenie aplikacji Windows 8, internetowych oraz biurowych w .NET 4.5 Framework, Gliwice, Helion, 2013 2. K. Rychlicki-Kicior: C# : tworzenie aplikacji graficznych w.NET 3.0, Gliwice, Helion, 2007