

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2023/2024

Wydział Mechaniczny

Kierunek studiów: Informatyka Stosowana

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: S

Stopień studiów: I

Specjalności: Bez specjalności- blok A,Bez specjalności- blok B

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Programowanie obiektowe
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	
KOD PRZEDMIOTU	WM INFST oIS B4 23/24
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	2.00
SEMESTRY	3

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	PROJEKT	SEMINARIUM
3	15	0	0	30	0	0

3 CELE PRZEDMIOTU

Cel 1 Przekazanie wiedzy i umiejętności z zakresu technik programowania zgodnych z paradygmatem obiektowym

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Absolwent zna i rozumie technologie programistyczne, metody projektowania i tworzenia aplikacji oraz języki programowania zgodnie z obiektywnym paradygmatem programowania.

EK2 Umiejętności Absolwent potrafi zastosować metody programowania obiektowego do rozwiązania postawionego prostego problemu programistycznego, w tym dobrać język programowania, niezbędne biblioteki oraz sposób komunikacji z użytkownikiem.

EK3 Umiejętności Absolwent potrafi zaprojektować i zbudować prostą aplikację zgodnie z zasadami programowania obiektowego, wykorzystując informacje pozyskane z literatury, dokumentacji oraz serwisów internetowych; potrafi ocenić przydatność poszczególnych technik programowania obiektowego do rozwiązania postawionego problemu programistycznego.

EK4 Kompetencje społeczne Absolwent jest gotów do ciągłego dokształcania się podnoszenia swoich kompetencji zawodowych i społecznych w zakresie i poszukiwania aktualnych rozwiązań w zakresie technik programowania obiektowego w warunkach szybko postępującego rozwoju informatyki.

6 TREŚCI PROGRAMOWE

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Wprowadzenie do obiektowej analizy problemu informatycznego. Obiektywny paradygmat programowania, w tym założenia SOLID. Podstawowe pojęcia i terminy: abstrakcja, enkapsulacja, dziedziczenie, polimorfizm. Zaawansowane techniki obiektowe: przeciążanie operatorów, klasy generyczne, operatory RTTI. Przedstawienie środowiska programowania oraz wymaganych narzędzi i bibliotek. Metodyka definiowania klas, atrybutów i metod. Włączanie bibliotek, używanie przestrzeni nazw. Tworzenie obiektów. Składniki klas o specjalnym znaczeniu: konstruktory i destruktory, składniki statyczne, składniki stałe, referencje do obiektów innych klas, klasy zagnieżdżone. Praktyka użycia enkapsulacji, definiowanie widoczności składników klasy. Obiektywne struktury danych, w tym klasy kontenerowe (kolekcje).	5
W2	Technika dziedziczenia. Charakterystyka i podstawowe rodzaje, w tym dziedziczenie wielobazowe i wielopokoleniowe. Korzyści oraz ograniczenia praktycznego zastosowania tej techniki. Dziedziczenie w kontekście widoczności składników klasy, omówienie składników, które nie są dziedziczone. Hierarchia dziedziczenia klas w bibliotekach SDK. Definiowanie klas i metod wirtualnych. Polimorfizm metody jako technika wywoływania funkcji wirtualnych. Definicja klas abstrakcyjnych oraz tworzenie hierarchii klas z ich udziałem. Mechanizm RTTI jako źródło informacji o klasie w warunkach stosowania polimorfizmu. Projektowanie graficznego interfejsu użytkownika GUI z wykorzystaniem obiektowych bibliotek komponentów wizualnych. Obiektywne techniki obsługi zdarzeń z wykorzystaniem konstrukcji jak wyrażenia lambda.	5

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W3	Definicje oraz klasyfikacja sytuacji wyjątkowych. Wykrycie oraz reakcja na wystąpienie wyjątku z wykorzystaniem dostępnych klas. Praktyczna realizacja wymagania obsłuż lub określ. Tworzenie klas wyjątków, programowe zgłaszanie wyjątków. Interfejs jako element programowania obiektowego. Definiowanie składników interfejsu. Technika implementacji w klasie oraz użycie w postaci typu referencyjnego jako przykłady praktycznych zastosowań interfejsów. Interfejsy jako alternatywa dla dziedziczenia. Zaawansowane techniki zastosowań interfejsów: interfejsy funkcjonalne, domyślne implementacje, wstrzykiwanie zależności, odwrócone sterowanie. Technika tworzenia klas generycznych z wykorzystaniem uogólnionych typów danych jako rodzaj polimorfizmu klasy. Tworzenie klas i metod generycznych z uwzględnieniem korzyści oraz ograniczeń. Zawężenie zakresu typów uogólnionych w klasach generycznych, w tym zastosowanie typu nieokreślonego z ograniczeniem od góry lub od dołu. Przeciążanie operatorów. Wielowątkowość, wykorzystanie klas implementujących wątki drugoplanowe. Wstęp do wzorców projektowych.	5

LABORATORIUM KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K1	Zapoznanie się ze środowiskiem programowania: edytor kodu, debugger, system pomocy, struktura projektu, pliki źródłowe, pośrednie i wynikowe. Opanowanie podstaw budowy, kompilacji i uruchamiania aplikacji w trybie tekstowym i z poziomu SDK. Budowa aplikacji wymagającej zdefiniowania klasy i utworzenia kilku instancji. System tworzenia dokumentacji projektowej i jego praktyczne wykorzystanie. Określanie stanu obiektu podczas jego tworzenia (konstruktor) oraz za pomocą zdefiniowanych metod publicznych. Formatowanie danych wyjściowych. Budowa aplikacji z wykorzystaniem związków klas typu agregacja (zawieranie) i asocjacja (zastosowanie kolekcji). Budowa aplikacji z zastosowaniem związku klas typu uogólnienie (technika dziedziczenia). Zaliczenie weryfikujące nabyte dotąd umiejętności.	12

LABORATORIUM KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K2	Zapoznanie się z podstawami budowy aplikacji z graficznym interfejsem użytkownika. Wybór układu (layout), rozmieszczanie komponentów, modyfikowanie ich właściwości i obsługa zdarzeń. Budowa aplikacji z GUI oraz praktycznym wykorzystaniem techniki definiowania klas abstrakcyjnych i polimorfizmu metody. Budowa aplikacji wykorzystującej system obsługi wyjątków. Implementacja odpowiedniej, adekwatnej reakcji na wystąpienie sytuacji wyjątkowej oraz programowe zgłaszanie wyjątku. Budowa aplikacji z wykorzystaniem technik definiowania i implementacji interfejsów. Wykorzystanie technik, jak wstrzykiwanie zależności i odwrócone sterowanie. Budowa aplikacji z wykorzystaniem obiektowych technik zapisu i odczytu danych z plików dyskowych oraz obsługi sytuacji wyjątkowych. Zapoznanie się z obiektowymi narzędziami do obsługi systemu plików. Budowa aplikacji implementującej wybrane podstawowe wzorce projektowe z grupy konstrukcyjnych. Tworzenie wątków drugoplanowych, przesyłanie danych, komunikacja między wątkami, synchronizacja. Budowa aplikacji realizującej komunikację sieciową z wykorzystaniem gniazd (sockets). Opracowanie protokołu przesyłania danych, implementacja serwera i klienta. Wykonanie tematu zaliczeniowego.	18

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykład

N2 Ćwiczenia laboratoryjne

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	45
Konsultacje przedmiotowe	4
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	11
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	0
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	60
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	2.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Test z wykładu

F2 Kody źródłowe z ćwiczeń laboratoryjnych

F3 Testy na laboratoriach komputerowych

F4 Dwa kolokwia praktyczne na zajęciach laboratoryjnych

OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Zaliczenie testu z wykładów

W2 Obecność na min. 75% zajęć laboratoryjnych, w tym max. 1 nieobecność nieusprawiedliwiona

W3 Zaliczenie ćwiczeń laboratoryjnych (kody źródłowe i testy)

OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA

B1 Dwa kolokwia praktyczne

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Student uzyskał nie więcej niż 50% punktów z testu zaliczeniowego z wykładów.
NA OCENĘ 3.0	Student uzyskał więcej niż 50% punktów z testu zaliczeniowego z wykładów do 60% włącznie.
NA OCENĘ 3.5	Student uzyskał więcej niż 60% punktów z testu zaliczeniowego z wykładów do 70% włącznie.
NA OCENĘ 4.0	Student uzyskał więcej niż 70% punktów z testu zaliczeniowego z wykładów do 80% włącznie.
NA OCENĘ 4.5	Student uzyskał więcej niż 80% punktów z testu zaliczeniowego z wykładów do 90% włącznie.
NA OCENĘ 5.0	Student uzyskał więcej niż 90% punktów z testu zaliczeniowego z wykładów.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Student spełnia mniej niż 6 wymagań na ocenę bardzo dobrą.
NA OCENĘ 3.0	Student spełnia 6 z 10 wymagań na ocenę bardzo dobrą.
NA OCENĘ 3.5	Student spełnia 7 z 10 wymagań na ocenę bardzo dobrą.
NA OCENĘ 4.0	Student spełnia 8 z 10 wymagań na ocenę bardzo dobrą.
NA OCENĘ 4.5	Student spełnia 9 z 10 wymagań na ocenę bardzo dobrą.
NA OCENĘ 5.0	Student potrafi: (1) poprawnie zdefiniować klasę wg wymagań, (2) zbudować GUI, (3) określić związki pomiędzy klasami, (4) użyć hermetyzacji, (5) użyć dziedziczenia, (6) użyć polimorfizmu, (7) zdefiniować i zastosować interfejsy, (8) zdefiniować i zastosować klasy generyczne, (9) zdefiniować obsługę wyjątków, (10) zbudować aplikację sieciową w oparciu o sockety
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	Student uzyskał poniżej 50% punktów z zaliczeń laboratoriów komputerowych.
NA OCENĘ 3.0	Student uzyskał więcej niż 50% punktów z zaliczeń laboratoriów komputerowych, do 60% włącznie.
NA OCENĘ 3.5	Student uzyskał więcej niż 60% punktów z zaliczeń laboratoriów komputerowych, do 70% włącznie.
NA OCENĘ 4.0	Student uzyskał więcej niż 70% punktów z zaliczeń laboratoriów komputerowych, do 80% włącznie.
NA OCENĘ 4.5	Student uzyskał więcej niż 80% punktów z zaliczeń laboratoriów komputerowych, do 90% włącznie.
NA OCENĘ 5.0	Student uzyskał więcej niż 90% punktów z zaliczeń laboratoriów komputerowych.
EFEKT KSZTAŁCENIA 4	

NA OCENĘ 2.0	Student nie potrafi wyszukiwać informacji dotyczących najbardziej aktualnych rozwiązań w zakresie technik programowania obiektowego.
NA OCENĘ 3.0	Student potrafi wyszukać pojedyncze informacje dotyczące pojedynczej techniki programowania obiektowego w lokalnej pomocy środowiska programowania.
NA OCENĘ 3.5	Student potrafi wyszukać pojedyncze informacje dotyczące dowolnych technik programowania obiektowego w lokalnej pomocy środowiska programowania.
NA OCENĘ 4.0	Student potrafi wyszukiwać dowolne informacje dotyczące technik programowania w lokalnej pomocy oraz dokumentacji on-line.
NA OCENĘ 4.5	Student potrafi wyszukiwać informacje na dowolny temat przy wykorzystaniu różnych źródeł lokalnych i zdalnych, jak pomoc lokalna, oficjalne serwisy internetowe, dokumentacja on-line, fora internetowe, itp.
NA OCENĘ 5.0	Student spełnia wymagania na ocenę 4.5 oraz z wielu znalezionych rozwiązań potrafi wybrać najlepsze z punktu widzenia rozwiązywanego problemu programistycznego.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	K1_W19 K1_W23 K1_W26	Cel 1	W1 W2 W3	N1 N2	F1 P1
EK2	K1_U18 K1_U26	Cel 1	K1 K2	N2	F2 F3 F4 P1
EK3	K1_U01 K1_U20	Cel 1	K1 K2	N2	F2 F3 F4 P1
EK4	K1_K01 K1_K08	Cel 1	W1 W2 W3 K1 K2	N1 N2	F1 F2 F3 F4 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

[1] | **Herbert Schildt** — *Java. Kompendium programisty*, Gliwice, 2020, Helion

LITERATURA UZUPEŁNIAJĄCA

- [1] | **Bruce Eckel** — *Thinking in Java, Edycja polska*, Gliwice, 2021, Helion
- [2] | **M. SOMASHEKARA** — *Object Oriented Programming with Java*, England, 2017, PRENTICE HALL

LITERATURA DODATKOWA

- [1] | **Graham Lee** — *Modern Programming: Object Oriented Programming and Best Practices: Deconstruct object-oriented programming and use it with other programming paradigms to build applications*, USA, 2019, Packt Publishing

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH**OSOBA ODPOWIEDZIALNA ZA KARTĘ**

dr hab. inż. Grzegorz, Mariusz Filo (kontakt: filo@mech.pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr hab. inż. Grzegorz Filo (kontakt: grzegorz.filo@pk.edu.pl)

2 dr inż. Paweł Lempa (kontakt: pawel.lempa@pk.edu.pl)

3 pracownicy Katedry Informatyki Stosowanej (kontakt:)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....

.....

.....