



### KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	
	studia niestacjonarne:	
Nazwa przedmiotu	<b>Programowanie obiektowe 2</b>	
Nazwa przedmiotu w języku angielskim	<b>Object-oriented programming 2</b>	
Obowiązuje od roku akademickiego	<b>2023/24</b>	

### USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	<b>Informatyka</b>
Poziom kształcenia	<b>I stopień</b>
Profil studiów	<b>Ogólnoakademicki</b>
Forma i tryb prowadzenia studiów	<b>Studia stacjonarne i niestacjonarne</b>
Zakres	<b>Wszystkie specjalności</b>
Jednostka prowadząca przedmiot	<b>Katedra Systemów Informatycznych</b>
Koordinator przedmiotu	<b>Dr inż. Mariusz Bedla</b>
Zatwierdził	<b>Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk</b>

### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<b>przedmiot kierunkowy</b>	
Status przedmiotu	<b>obowiązkowy</b>	
Język prowadzenia zajęć	<b>polski</b>	
Usytuowanie w planie studiów - semestr	studia stacjonarne	<b>semestr III</b>
	studia niestacjonarne	<b>semestr III</b>
Wymagania wstępne	<b>Algorytmy i struktury danych Podstawy programowania 1 Programowanie obiektowe 1</b>	
Egzamin (TAK/NIE)	<b>TAK</b>	
Liczba punktów ECTS	<b>6</b>	

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	<b>30</b>		<b>30</b>	<b>15</b>	
	studia niestacjonarne:	<b>18</b>		<b>18</b>	<b>9</b>	

## EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Zna i rozumie paradygmat oraz zasady programowania obiektowego.	INF_W09
	W02	Zna i rozumie techniki oraz konstrukcje programistyczne związane z programowaniem obiektowym.	INF_W09
Umiejętności	U01	Potrafi implementować, testować i debugować programy obiektowe.	INF_U09
	U02	Potrafi ocenić przydatności różnych paradygmatów do rozwiązywania różnego typu problemów.	INF_U09
Kompetencje społeczne	K01	Jest gotów do wykorzystania posiadanej wiedzy w życiu zawodowym.	INF_K01, INF_K02

## TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	<ul style="list-style-type: none"><li>• Konstrukcje programistyczne w programowaniu obiektowym</li><li>• Przegląd elementów paradygmatu i zasad programowania obiektowego</li><li>• Porównanie paradygmatu obiektowego z innymi paradygmatami</li><li>• Ulotne przechowywanie obiektów</li><li>• Trwałe przechowywanie obiektów</li><li>• Współbieżny dostęp do obiektów</li><li>• Realizacja multiplatformowego interfejsu graficznego w aplikacjach obiektowych</li><li>• Komunikacja sieciowa w programowaniu obiektowym</li><li>• Realizacja typowych zadań programistycznych z wykorzystaniem programowania obiektowego</li></ul>
laboratorium	<ul style="list-style-type: none"><li>• Tworzenie programów wykorzystujących podstawowe konstrukcje programistyczne oraz paradygmat obiektowym</li><li>• Tworzenie programów zapewniających przechowywanie oraz współbieżny dostęp do obiektów</li><li>• Tworzenie programów posiadających multiplatformowego interfejsu graficzny</li><li>• Tworzenie programów wykorzystujących komunikację sieciową</li><li>• Tworzenie programów wykorzystujących również inne paradygmaty poza obiektowym</li><li>• Testowanie i debugowanie opracowanych programów</li></ul>
projekt	Tematyka zagadnień projektowych obejmuje stworzenie aplikacji obiektowej.

## METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X				
W02		X				
U01			X	X		X
U02				X		
K01				X		

## FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% punktów z pytań egzaminacyjnych.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z zadań laboratoryjnych oraz kolokwium.
projekt	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z zadań projektowych.

\*) zostawić tylko realizowane formy zajęć

## NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednostka
		studia stacjonarne					studia niestacjonarne					
		W	C	L	P	S	W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	30		30	15		18		18	9		h
2.	Inne (konsultacje, egzamin)	2		2	1		2		2	1		h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>80</b>					<b>50</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>3,2</b>					<b>2</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>70</b>					<b>100</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>2,8</b>					<b>4</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>45</b>					<b>27</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>1,8</b>					<b>1,08</b>					ECTS
9.	<b>Sumaryczne obciążenie pracą studenta</b>	<b>150</b>					<b>150</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>6</b>										ECTS

## LITERATURA

1. Java. Kompendium programisty. Wydanie XI, Herbert Schildt, Helion, 2020
2. Java. Przewodnik dla początkujących. Wydanie VIII, Herbert Schildt, Helion, 2020
3. Java. Podstawy. Wydanie XII, Cay S. Horstmann, Helion, 2022
4. Java. Techniki zaawansowane. Wydanie XI, Cay S. Horstmann, Helion, 2020
5. Java. Efektywne programowanie. Wydanie III, Joshua Bloch, Helion, 2018
6. Algorytmy. Wydanie IV, Robert Sedgewick, Kevin Wayne, Helion, 20212
7. JavaFX. Tworzenie graficznych interfejsów użytkownika, Urszula Piechota, Jacek Piechota, Helion, 2021
8. Inżynieria oprogramowania. Metody obiektowe w teorii i w praktyce. Ian Graham, WNT, Warszawa 2004
9. Słownik terminów z zakresu obiektowości. Kazimierz Subieta, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999

10. Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Helion, 2010
11. Wzorce projektowe. Rusz głową!, Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson, Helion, 2010
12. Inżynieria oprogramowania w ujęciu obiektywnym. UML, wzorce projektowe i Java, Bernd Bruegge, Allen H. Dutoit, Helion, 2011
13. UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji. Wydanie III, Craig Larman, Helion, 2011