

## KARTA KURSU

Nazwa	<b>Techniki programowania obiektowego</b>
Nazwa w j. ang.	<b>Object Oriented Programming Techniques</b>

Koordynator	prof. dr hab. Piotr Czerski	Zespół dydaktyczny
		prof. dr hab. Piotr Czerski dr Rostislav Fojtik mgr inż. Agnieszka Smolarek
Punktacja ECTS*	st. stacjonarne: 3 st. niestacjonarne: 3	

### Opis kursu (cele kształcenia)

Celem kursu jest rozszerzenie wiedzy na temat wybranego języka programowania obiektowego. W ramach przedmiotu omawiane będą również wybrane wzorce projektowe. Kurs prowadzony jest w języku polskim.

### Warunki wstępne

Wiedza	Student zna podstawy analizy, projektowania i programowania obiektowego oraz podstawy języka <b>C++</b> , Java lub C#
Umiejętności	Potrafi zapisywać podstawowe algorytmy i struktury danych, projektuje i tworzy z wykorzystaniem podstaw metodologii obiektowej proste programy w języku <b>C++</b> , Java lub C#
Kursy	--

### Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01: Zna i umie dobierać odpowiednie struktury danych (kontenery) w celu ich optymalnego użycia w projektowanych aplikacjach i używanych algorytmach.	K_W04, K_W07
	W02: Zna i rozumie cel stosowania wzorców projektowych oraz orientuje się w ich implementacji w wybranym języku programowania.	K_W07
	W03: Posiada wiedzę umożliwiającą posługiwanie się zaawansowanymi elementami środowiska programistycznego służącymi do analizy i monitorowania przebiegu działania bardziej złożonych aplikacji.	K_W07

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	U01: potrafi wykorzystywać odpowiednie struktury danych (kontenery) w opracowywanych przez siebie programach komputerowych	K_U04
	U02: potrafi zidentyfikować typowe zadania programistyczne oraz dobrać i zaimplementować w wybranym języku programowania odpowiednie wzorce projektowe	K_U04
	U03: umie uzasadniać wybór konkretnych rozwiązań programistycznych w odniesieniu do realizowanych zadań uwzględniając ich złożoność, wydajność oraz możliwości rozwoju aplikacji.	K_U02, K_U04
	U04: potrafi posługiwać się wybranymi, zaawansowanymi elementami środowiska programistycznego, lokalizuje i naprawia błędy w opracowywanych przez siebie aplikacjach oraz umie je optymalizować pod względem wybranych kryteriów (wydajnościowych, funkcjonalnych itp.)	K_U04, K_U08

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	K01: potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych oraz dokumentacji technicznej) do poszerzania własnej wiedzy i zdobywania nowych umiejętności	K_K01

### Studia stacjonarne

Organizacja													
Forma zajęć	Wykład (W)	Ćwiczenia w grupach											
		A		K		L		S		P		E	
Liczba godzin	15					30							

### Studia niestacjonarne

Organizacja												
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin	10					20						

## Opis metod prowadzenia zajęć

Kurs składa się z wykładu i ćwiczeń prowadzonych w formie laboratoriów. W ramach laboratoriów studenci projektują i tworzą zadane programy w wybranym języku programowania obiektowego, które następnie są omawiane.

Studenci w ramach pracy indywidualnej realizują zadane projekty programistyczne, które są omawiane z prowadzącym zajęcia.

## Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01					X	X		X					
W02					X	X		X					
W03					X	X		X					
U01					X	X		X					
U02					X	X		X					
U03					X	X		X					
U04					X	X		X					
K01						X		X					

Kryteria oceny	Na ocenę dobrą lub bardzo dobrą zasługuje student, który potrafi wykorzystać w programach zaawansowane rozwiązania działające w oparciu o wybrane klasy-kontenery, wzorce projektowe oraz potrafi uzasadnić ich dobór do realizacji poszczególnych zadań.
----------------	---

Uwagi	W ramach kursu omawiane są zaawansowane elementy wybranego języka programowania. Szczególną uwagę zwraca się na znajomość i prawidłowy dobór wzorców projektowych oraz użycie kontenerów właściwych dla realizacji poszczególnych zadań programistycznych. Opanowanie zdobytej wiedzy odbywa się także w formie krótkich prac pisemnych lub testów sprawdzających podstawową znajomość wzorców projektowych i kontenerów.
-------	--

## Treści merytoryczne (wykaz tematów)

Szablony funkcji, Szablony klas,  
Klasy pojemnikowe – Wektor, Lista, Talia, Stos, Kolejka,  
Pojemnik kojarzący, zbiór, Mapa, słownik

Iteratory

Obsługa sytuacji wyjątkowych

Odwikłanie stosu

Rzucanie wyjątku z argumentem automatycznym

Wybrane wzorce projektowe:

- kreacyjne (budowniczy, fabryka abstrakcyjna, metoda wytwórcza, singleton)
- strukturalne (adapter, dekorator, fasada, kompozyt)
- behawioralne (obserwator, strategia)

## Wykaz literatury podstawowej

1. Bruce Eckel, „*Thinking in C++*”, HELION, 2002r.
2. Bruce Eckel, „*Thinking in Java*”, HELION, 2006r.
3. Bjarne Stroustrup, „*Język C++*”, WNT, 2008r. - wybrane rozdziały
4. Jerzy Grebosz, „*Pasja C++*”, Oficyna Kallimach, 2003

## Wykaz literatury uzupełniającej

1. Nicolai M. Josuttis, „*C++ Biblioteka standardowa. Podręcznik programisty*”, HELION, 2003r.
2. Kathy Sierra, Bert Bates, „*Rusz głową ! Java*”, HELION, 2010r.
3. Alan Shalloway, James R. Trott, „*Programowanie zorientowane obiektowo. Wzorce projektowe*”, HELION, 2005 r.
4. Scott Meyers, „*C++. 50 efektywnych sposobów na udoskonalenie Twoich programów*”, 2003, ISBN 83-7361-345-5

## Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) **studia stacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	5
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	10
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	10
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu/zaliczenia	10
Ogółem bilans czasu pracy		90
Liczba punktów ECTS w zależności od przyjętego przelicznika		3

## Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) **studia niestacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	5
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	10
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	10
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	15
	Przygotowanie do egzaminu/zaliczenia	20
Ogółem bilans czasu pracy		90
Liczba punktów ECTS w zależności od przyjętego przelicznika		3