



## Karta przedmiotu

Nazwa i kod przedmiotu	Obiektowe języki programowania II, PG_00037343						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	październik 2022 r.		Rok akademicki realizacji przedmiotu		2023/2024		
Poziom kształcenia	I stopnia - inżynierskie		Grupa zajęć		Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		mieszane (blended-learning)		
Rok studiów	2		Język wykładowy		polski		
Semestr studiów	4		Liczba punktów ECTS		5.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Wydział Fizyki Technicznej i Matematyki Stosowanej -> Katedra Fizyki Teoretycznej i Informatyki Kwantowej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. arch. Jan Kozicki				
	Prowadzący zajęcia z przedmiotu		dr hab. inż. arch. Jan Kozicki				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	45.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 58.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	60		10.0		55.0	125
Cel przedmiotu	Nauczenie studenta programowania w wybranych językach obiektowych (C++ ISO/ANSI, C++14, C++17). Przedstawienie korzyści wynikających z posługiwania się technikami zorientowanymi obiektowo.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K6_W05] Posiada podstawową wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.		Posiada podstawową wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.		[SW1] Ocena wiedzy faktograficznej		
	[K6_K01] Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych. Potrafi inspirować i organizować proces uczenia się innych osób.		Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych. Potrafi inspirować i organizować proces uczenia się innych osób.		[SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce		
	[K6_U03] Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.		Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.		[SU1] Ocena realizacji zadania		
Treści przedmiotu	<ul style="list-style-type: none"><li>• Podstawowe elementy projektowania obiektowego</li><li>• Wielokrotne wykorzystywanie kodu</li><li>• Analiza obiektowa</li><li>• Abstrakcyjne typy danych</li><li>• Klasy i obiekty</li><li>• Zarządzanie pamięcią</li><li>• Mechanizmy dziedziczenia</li><li>• Obsługa wyjątków</li><li>• Metodyka projektowania obiektowego</li><li>• Zastosowanie technik obiektowych w różnych językach programowania</li></ul>						
Wymagania wstępne i dodatkowe	Znajomość systemów operacyjnych Unix/Linux i MS Windows. Wiedza z przedmiotów Proceduralne Języki Programowania I (FIZ1C301) i II (FIZ1C307) oraz Obiektowe Języki Programowania I (FIZ1C305).						

Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Krótkie sprawdziany ("wejściówki") na początku zajęć laboratoryjnych.	50.0%	20.0%
	Cotygodniowo zadawane krótkie zadania z bieżącego materiału.	50.0%	20.0%
	Pisemny sprawdzian wiedzy z wykładu.	50.0%	20.0%
	Sprawdzian z praktycznej umiejętności programowania (C++ ISO/ANSI).	50.0%	20.0%
	Projekt programistyczny - C++	50.0%	20.0%
Zalecana lista lektur	Podstawowa lista lektur	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley	
	Uzupełniająca lista lektur	1. B. Meyer Programowanie zorientowane obiektowo, Helion	
	Adresy eZasobów	Adresy na platformie eNauczanie: Obiektowe języki programowania II 2023/2024 sem.letni - Moodle ID: 35076 <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=35076">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=35076</a>	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p>1. Stwórz wektor liczb Fibonacciego i wypisz je stosując funkcję z ćwiczenia 2 (str.300). Aby stworzyć wektor napisz funkcję fibonacci(x,y,v,n), w której dwie początkowe liczby x i y są typu int, v to pusty std::vector, and n to liczba elementów. Każdy kolejny element w sekwencji to suma dwóch poprzednich, przykładowo, startując z 1 i 2 otrzymamy: 1, 2, 3, 5, 8, 13, 21, . . . Twoja funkcja powinna tworzyć taką sekwencję z dwóch początkowych liczb x,y.</p> <p>2. Zdefiniuj klasę Zlecenie zawierającą imię (zleceniodawcy), adres, datę i std::vector. Zakup to jest klasa zawierająca nazwę produktu, cenę oraz ilość. Zdefiniuj mechanizm czytania i pisania do pliku do klasy Zlecenie. Zdefiniuj funkcję drukującą zlecenia na ekranie. Stwórz plik zawierający co najmniej 10 zleceń, wczytaj go do std::vector, posortuj go według imienia zleceniodawcy i zapisz z powrotem do pliku. Zdefiniuj drugi plik zawierający co najmniej 10 zleceń z których około jedna trzecia to takie same zlecenia jak w pierwszym pliku, wczytaj go to std::list, posortuj go po adresie zleceniodawcy i zapisz z powrotem do pliku. Następnie połącz oba pliki używając polecenia std::merge()</p> <p>3. Napisz funkcję wyszukiwania binarnego dla std::vector (bez używania funkcji std::find() z biblioteki standardowej). Możesz stworzyć do tego dowolny interfejs jaki uznasz za stosowny. Przetestuj ją. Czy jesteś pewien, że Twoja funkcja działa prawidłowo? Teraz napisz funkcję wyszukiwania binarnego dla std::list. Przetestuj ją. Na ile te funkcje są podobne? Zastanów się czy można je zastąpić jedną funkcją z użyciem szablonów (templates). Czy dałoby się to zrobić gdybyś nie wiedział o istnieniu STL?</p> <p>4. Zmodyfikuj kalkulator z rozdziału 7 tak aby mógł on pobierać dane wejściowe z pliku i zapisywać wynik do pliku (lub skorzystaj z przekierowywania strumieni w systemie operacyjnym). Następnie skonstruuj w miarę dokładny test tego kalkulatora.</p> <p>5. Jakie są wady i zalety intruzywnych kontenerów w porównaniu ze standardowymi nie-intruzywnymi kontenerami C++? Zrób listę was i zalet.</p> <p>6. Zrób okienko (oparte na przykładzie My_window ze str.578) zawierające szachownicę 4x4 z kwadratowymi przyciskami. Przycisk po przyciśnięciu powinien wykonywać jakąś prostą akcję, np. wypisać współrzędne w okienku output lub zmienia swój kolor do czasu aż nie zostanie wciśnięty kolejny przycisk.</p> <p>7. wyjaśnić słowa kluczowe "this" i "constexpr"</p> <p>8. wyjaśnić na przykładzie kodu czym się różni polimorfizm statyczny od polimorfizmu dynamicznego, posłużyć się przy tym słowami kluczowymi "typename" oraz "virtual".</p>		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		