

KARTA PRZEDMIOTU

Nazwa przedmiotu: **Programowanie obiektowe (MSCKAu>SM21PO22)**

Nazwa w języku polskim:

Nazwa w jęz. angielskim: **Object-oriented programming**

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

ZAL

Język wykładowy:

polski

Strona WWW:

<https://platforma.polsl.pl/rau3/course/view.php?id=590>

Skrócony opis:

Podstawowym celem przedmiotu jest zapoznanie studentów z wybranymi narzędziami do programowania w stylu orientowanym obiektowo oraz z wybranymi wzorcami projektowymi. Drugim celem jest pokazanie sposobu symulacji dyskretnych równań różnicowych i organizacji obliczeń w pętli ujemnego sprzężenia zwrotnego.

Zakłada się, że studenci rozpoczynający kurs znają podstawy matematyki dyskretnej oraz podstawy programowania, co najmniej w językach proceduralnych.

Opis:

ECTS: 3

Suma godzin: 75 (kontaktowo 40h / praca własna 35h)

Wykład: 15h

Laboratorium: 15h

Inne (omówienie kodów programów): 10h

Praca własna studenta obejmuje: przygotowanie do zajęć, dokończenie postawionych podczas laboratorium zadań, dodanie komentarzy do kodu, przygotowanie do testu.

Wykład

Wykład jest wykładem multimedialnym, bazującym na zawierających animacje slajdach. Wykładowca omawia wybrane zagadnienia wykładowe rozszerzając w razie potrzeby te zagadnienia, które najbardziej studentów interesują. Wybrane zagadnienia są dodatkowo ilustrowane demonstracjami fragmentów kodu, dokumentacji, działających programów bądź symulacji.

1. Systemy dokumentowania kodu programu. Doxygen: opcje konfiguracyjne, sposoby dokumentowania kodu, instrukcje strukturalne, formatowanie dokumentacji, odnośniki, kod HTML w dokumentacji.
2. Język C a C++. Typy różnicowane przy przeładowaniu funkcji, etapy dopasowania. Konstruktory i destruktor, lista inicjalizacyjna. Niepubliczny konstruktor, singletony.
3. Automatyczne i jawne konwersje typów, konstruktor jako konwerter, operator konwersji. Najczęściej przeładowywane operatory: operator przypisania, tablicowy, funkcyjny oraz odniesienia przez wskaźnik. Zręczne wskaźniki.
4. Wyjątki: specyfikacja wyjątków, wyjątki w konstruktorze i destruktorze, blok „try” na poziomie funkcji. Dziedziczenie i zawieranie klas: kolejność konstrukcji i destrukcji, sposoby dziedziczenia, zagnieżdżona definicja klasy, dziedziczenie wielokrotne i wirtualne. Funkcje wirtualne: rzutowanie dynamiczne, funkcje czysto wirtualne i klasy abstrakcyjne.
5. Szablony: sposoby uokretniania, specjalizacja, szablony z wieloma parametrami, szablony a przyjaźń. Przestrzenie nazw: deklarowanie i używanie, dyrektywa a instrukcja „using”, przestrzenie anonimowe.
6. Iteratory: typy, sposoby implementacji, iteratory predefiniowane. Pojemniki: typy pojemników, pojemniki standardowe, pojemnik „vector” – budowa, interfejs i wykorzystanie.
7. Pojemniki asocjacyjne: przykłady wykorzystania. Obiekty funkcyjne. Standardowe algorytmy. Klasa string: wykorzystanie i budowa wewnętrzna. Biblioteka do lokalizacji.
8. Standardowe wejście i wyjście. Strumienie. Flagi stanu formatowania. Pisanie własnych manipulatorów. Błędy i wyjątki w strumieniach. Wewnętrzna budowa strumieni, operacje nieformatowane. Strumienie plikowe i znakowe.
9. Wzorce projektowe w programowaniu orientowanym obiektowo.
10. Zmiany wprowadzane przez nowe wersje standardu.

Zajęcia laboratoryjne

1. Projektowanie i programowanie klasy do symulacji obiektu dyskretnego typu ARX.
2. Projektowanie interfejsu regulatora. Programowanie regulatorów
3. Wykorzystanie wzorca projektowego do implementacji elastycznego rozwiązania generacji wartości zadanej.
4. Wykorzystanie wzorca projektowego do implementacji szeregowych i równoległych połączeń obiektów.
5. Przechowywanie danych konfiguracyjnych w plikach. Tekstowy interfejs użytkownika.

Literatura:

1. D. Bismor: Programowanie systemów sterowania. Narzędzia i metody. WNT, Warszawa, 2010
2. A. Koenig, B. Moo: Potęga języka C++, Helion, 2004.
3. B. Eckel: Thinking in C++, Second Edition, vol. 1 & 2, Upper Saddle River, NJ, 2000 & 2004.
4. S. Meyers: Język C++ bardziej efektywny, WNT, 1998
5. M. Cline: C++ FAQ Lite, <http://www.parashift.com/c++-faq-lite>
6. D. van Heesch: Doxygen Manual, <http://www.stack.nl/~dimitri/doxygen/manual.html>
7. ISO/IEC JTC 1/SC22/WG21: Programming Languages -- C++, Working Draft, ISO, 2006

Efekty uczenia się:

Po ukończeniu przedmiotu student:

USOSweb: Szczegóły przedmiotu: MSCKAu>SM21PO22, w cyklu: <brak>, jednostka dawcy: <brak>, grupa przedm.: <brak>

- Zna składnię języka C++ według standardu ISO i zawartość standardowych bibliotek (napisanie kodu) K2A_W01, K2A_W02
- Zna sposób opisu podstawowych elementów automatyki dyskretnej za pomocą równań różnicowych (napisanie kodu) K2A_W01, K2A_W02
- Potrafi zaprojektować program w stylu orientowanym obiektowo, w oparciu o wzorce projektowe (napisanie kodu) K2A_U05, K2A_U08
- Potrafi napisać w stylu orientowanym obiektowo program implementujący techniki symulacji, identyfikacji, sterowania dyskretnego, a także przetestować poprawność jego działania (napisanie kodu) K2A_U05, K2A_U08
- Potrafi znaleźć błędy i zweryfikować działanie programu (napisanie kodu) K2A_W05

Metody i kryteria oceniania:

Z uwagi na fakt, że wszystkie treści wykładowe są odzwierciedlone w zadaniach laboratoryjnych, zaliczenie zarówno wykładu, jak i laboratorium następuje na podstawie zaliczenia poszczególnych fragmentów kodu, powstających podczas ćwiczeń laboratoryjnych.

Warunkiem zaliczenia każdego ćwiczenia laboratoryjnego jest przedstawienie kompletnego kodu z komentarzami zapisanymi w stylu umożliwiającym generację dokumentacji API. Kod musi być zgodny ze standardem ISO C++ (dać się skompilować).

Ćwiczenia są wykonywane w 5 blokach o długości 2:15 godziny zegarowej. Zaliczenie laboratorium następuje po uzyskaniu oceny z każdego z ćwiczeń pod warunkiem, że średnia wszystkich ocen jest równa co najmniej 3.0. W przypadku braku zaliczenia w ten sposób obliczonego istnieje możliwość jego uzyskania po wykonaniu ćwiczenia dodatkowego, zleconego przez prowadzącego.

Sylabus obowiązuje od roku akademickiego 2025/2026 i nie może być zmieniony podczas trwania semestru.

Punkty przedmiotu w cyklach:

<bez przypisanego programu>			
Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	3	2022/2023-L	