KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Języki obiektowe [S1MiKC2>JO]

Przedmiot

Kierunek studiów
Mikroelektronika i komunikacja cyfrowa

Rok/Semestr
2/3

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
24

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Marek Michalski
marek.michalski@put.poznan.pl

dr inż. Michał Sybis
michal.sybis@put.poznan.pl

Wykładowcy

Wymagania wstępne

Podstawowa znajomość zagadnień z zakresu logiki matematycznej i kombinatoryki. Zdolność do tworzenia prostych algorytmów. Dodatkowo, powinien posiadać wiedzę i podstawowe umiejętności programowania oraz obsługi komputera typu PC. Umiejętność pozyskiwania informacji z literatury i innych źródeł w języku polskim lub angielskim, a także ich integracji, interpretacji i wyciągania wniosków. Świadomość ograniczeń własnej wiedzy i umiejętności oraz potrzeba ciągłego samodoskonalenia i zdobywania nowych kompetencji.

Cel przedmiotu

Celem przedmiotu jest zapoznanie studentów z podstawami inżynierii oprogramowania, w tym z zasadami projektowania obiektowego. Zajęcia wprowadzają kluczowe zagadnienia związane z praktyką programowania obiektowego w języku C++ oraz Java, obejmując zarówno podstawowe, jak i bardziej zaawansowane konstrukcje języka.

Przedmiotowe efekty uczenia się

Wiedza:
1. Ma zaawansowaną wiedzę w zakresie zasad tworzenia programów komputerowych i wykorzystania poznanych struktur języka C++/Java.

Umiejętności:
1. Potrafi opracować proste programy korzystając z języka C++ w celu przeprowadzenia analizy i rozwiązywania problemów właściwych dla kierunku studiów.
2. Potrafi zapisać, przedstawić i przetworzyć zebrane dane w formie liczbowej i graficznej korzystając z poznanych języków.
3. Potrafi zaimplementować i wykorzystać znane modele matematyczne do rozwiązywania problemów korzystając z języka C++/Java.

Kompetencje społeczne:
1. Rozumie konieczność poszerzania wiedzy na temat korzystania z programowania obiektowego.
2. Ma świadomość możliwości i ograniczeń współczesnej informatyki przy jednoczesnym otwarciu na możliwość zastosowań w nowych dziedzinach życia codziennego, gospodarki, techniki i nauki.
3. Ma umiejętność formułowania własnych opinii na temat aktualnie stosowanych i dostępnych technologii i rozwiązań w projektowaniu nowoczesnych systemów informatycznych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w trakcie wykładów jest weryfikowana podczas zaliczenia, które ma formę pisemną i/lub ustną. Zaliczenie składa się 6 - 11 pytań (testowych i/lub otwartych), które mogą być różnie punktowane. Próg zaliczeniowy to 50% możliwych do zdobycia punktów.
Umiejętności nabyte podczas laboratorium są weryfikowane poprzez realizację ćwiczeń praktycznych (polegających na implementacji określonych funkcjonalności zgodnie z przygotowaną instrukcją) oraz 1-2 kolokwiów i/lub projektu zaliczeniowego. Realizacja każdego z ćwiczeń jest oceniana, a liczba punktów przypisana do danego zadania zależy od jego stopnia skomplikowania. Wpływ na ocenę końcową ma również ocena pracy i zaangażowania studenta podczas zajęć, a także realizacja ewentualnych dodatkowych zadań domowych. Ocena końcowa jest ustalana na podstawie liczby zdobytych punktów, z progiem zaliczeniowym wynoszącym 50% maksymalnej możliwej liczby punktów za wszystkie ćwiczenia. Ćwiczenia i kolokwia i projekt mają różną wagę w ocenie końcowej, co odzwierciedla ich różną złożoność i znaczenie w procesie weryfikacji umiejętności.
Skala ocen: < 50% - 2,0 (ndst); 50% do 59% - 3,0 (dst); 60% do 69% - 3,5 (dst+); 70% do 79% - 4,0 (db); 80% do 89% - 4,5 (db+); 90% do 100% - 5,0 (bdb).

Treści programowe

Tematyka wykładu koncentruje się na programowaniu obiektowym, obejmując kluczowe zagadnienia, takie jak klasy, obiekty, dziedziczenie, polimorfizm oraz enkapsulacja danych, które stanowią fundament nowoczesnego projektowania oprogramowania. Omówione zostaną również praktyczne zastosowania standardowej biblioteki szablonów (STL), w tym struktury danych, algorytmy oraz iteratory, które znacząco ułatwiają pracę z językiem C++ i Java. Poruszone zostaną kwestie bezpiecznego programowania, niebezpieczestw wynikających z nieoptymalnego i nieprawidłowego kodu lub słabości wybranych rozwiazań.

Tematyka zajęć

W ramach wykładów i laboratoriów C++ omawiane są następujące zagadnienia:
1. Klasy i obiekty (tworzenie, atrybutu, konstruktory, destruktory).
2. Zarządzanie pamięcią i wskaźniki (operatory new, delete, smart pointers).
3. Dziedziczenie i polimorfizm (dziedziczenie, dziedziczenie wielokrotne, funkcje wirtualne).
4. Obsługa wyjątków i projektowanie wielowątkowe (mechanizm try-catch, obsługa wielu wątków, synchronizacja wątków).
5. Zaawansowane techniki programowania obiektowego (szablony, przeciążanie operatorów).
6. Biblioteki (w tym STL).
W ramach wykładów i laboratoriów Java realizowane są następujące tematy:
1. Podstawowe elementy i składnia języka Java: typy danych, operatory arytmetyczne i kolejność działań, operacje wejścia-wyjścia, pętle, podejmowanie decyzji (instrukcje warunkowe), funkcje,

operacje z wykorzystaniem tablic.
2. Obiektowość, w tym polimirfizm, dziedziczenie, różne jego rodzaje (klasy abstrakcyjne, interfejsy), pakiety definicji klas, zasięgi definicji.
3. Przetwarzanie danych - odczyt i zapis plików w różnych formatach, graficzna prezentacja wyników, wybrane biblioteki, komunikacja z bazą danych, komunikacja sieciowa.
4. Implementacja wybranych zagadnień rachunku prawdopodobieństwa - generowanie liczb pseudolosowych, wyznaczanie momentów, elementy analizy statystycznej (zgodność z rozkładem, losowość).
5. Obsługa i generowanie wyjątków, programowanie wielowątkowe.
6. Realizacja wybranych usług sieciowych (np. HTTP)
7. Rozwiazania typowe dla Javy, np. servlety, JSP

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, ilustrowana przykładami realizowanymi na komputerze lub przedstawianymi na tablicy, ew. forma warsztatowa.
2. Ćwiczenia laboratoryjne: wykonanie zadań podanych przez prowadzącego z wykorzystaniem komputerów - ćwiczenia praktyczne, ew. wspomagane prezentacją multimedialną

Literatura

Podstawowa:
Jerzy Grębosz, Opus Magnum C++11 : programowanie w języku C++. T. 1/T.2/T.3, Gliwice : Wydawnictwo Helion, 2020.
Bruce Eckel, Thinking in Java

Uzupełniająca:
Kayshav Dattatri, Język C++. Efektywne programowanie obiektowe, Helion, 2005

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	84	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	54	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwiów/egzaminu, wykonanie projektu)	30	1,00