

Elektronika i Telekomunikacja (wrzesień 2025)

Metodyki programowania

1. **Algorytmy i ich analiza.**
 - 1.1. Złożoność czasowa i pamięciowa algorytmów.
 - 1.2. Szacowanie złożoności algorytmicznej.
 - 1.3. Algorytmy sortowania.
 - 1.4. Algorytmy wyszukiwania w strukturach danych.
2. **Typy i struktury danych.**
 - 2.1. Podstawowe typy danych.
 - 2.2. Struktury danych: tablice, listy, drzewa, stos, mapa.
 - 2.3. Reprezentacja łańcuchów tekstowych (string).
3. **Operatory i instrukcje sterujące.**
 - 3.1. Operatory arytmetyczne, logiczne, bitowe, warunkowe.
 - 3.2. Hierarchia i łączność operatorów.
 - 3.3. Instrukcje warunkowe (if, switch-case).
 - 3.4. Pętle (for, while, do-while).
4. **Wskaźniki i referencje.**
 - 4.1. Składnia i semantyka.
 - 4.2. lvalue reference.
 - 4.3. Wskaźniki na funkcje.
 - 4.4. Mechanizm wywołań zwrotnych (callback).
5. **Funkcje i programowanie funkcyjne.**
 - 5.1. Rola i struktura funkcji.
 - 5.2. Przekazywanie argumentów i zwracanie wyników.
 - 5.3. Rekurencja.
 - 5.4. Programowanie funkcyjne: funkcje wyższego rzędu, czyste funkcje, unikanie efektów ubocznych.
6. **Obsługa błędów i wyjątków.**
 - 6.1. Rodzaje błędów i wyjątków.
 - 6.2. Obsługa wyjątków w C++/Python.
7. **Programowanie obiektowe.**
 - 7.1. Założenia paradygmatu obiektowego.
 - 7.2. Metody konstrukcyjne i dostępne.
 - 7.3. Wzorce projektowe (klasyfikacja, przykłady).
8. **Dobre praktyki programistyczne.**
 - 8.1. Zasady SOLID.
 - 8.2. Refaktoryzacja kodu.
 - 8.3. Debbuger i profiler.
9. **Szablony.**
 - 9.1. Szablony funkcji.
 - 9.2. Szablony klas.
 - 9.3. Metodyka projektowania konstrukcji szablonych.
10. **Proces budowania kodu.**
 - 10.1. Preprocessing.
 - 10.2. Kompilacja.
 - 10.3. Konsolidacja.
 - 10.4. Struktura pliku obiektowego (format ELF).
 - 10.5. Optymalizacja.