# Основни термини в програмирането

* Програмиране: Процесът на писане на инструкции, които компютърът може да изпълнява, за да извърши конкретна задача или да реши определен проблем.
* Програмен Език: Формализиран език, който съдържа набор от инструкции, които могат да бъдат използвани за създаване на програми, които контролират поведението на машина, особено компютър.
* Алгоритъм: Набор от стъпки или правила, следвани за решаване на проблем или извършване на задача.
* Променлива: Именувана област в паметта, която съхранява стойности, които могат да се променят по време на изпълнение на програмата.
* Функция: Блок код, който е организиран за извършване на конкретна задача; функциите могат да приемат входни данни и да връщат резултат.
* Обектно-ориентирано програмиране (ООП): Парадигма, която използва "обекти" – данни, структурирани в полета, и код, в формата на процедури, често наречени методи.
* Клас: Шаблон или чертеж, който се използва за създаването на обекти. Определя какви данни обектът съхранява и какви операции (методи) може да изпълнява.
* Наследяване: Основен концепт в ООП, който позволява на един клас да наследи свойства и методи от друг клас.
* Интерфейс: Определя набор от методи, които даден клас трябва да имплементира, без да задава как точно тези методи трябва да бъдат изпълнени.
* API (Application Programming Interface): Набор от правила и спецификации, които позволяват на различни софтуерни приложения да комуникират помежду си.
* Итерация: Повторение или цикъл в програмирането, където блок от инструкции се изпълнява многократно.
* Рекурсия: Техника в програмирането, където функция се извиква сама в своето тяло.
* Масив (Array): Структура от данни, която съхранява фиксиран брой елементи от един и същ тип в наредена последователност. Елементите могат да бъдат достъпени директно чрез техния индекс.
* Стек (Stack): Линейна структура от данни, която следва принципа "последен вътре, първи навън" (LIFO). Позволява добавяне и премахване на елементи само от единия край.
* Опашка (Queue): Линейна структура от данни, която следва принципа "първи вътре, първи навън" (FIFO). Елементите се добавят от единия край и се премахват от другия.
* Свързан списък (Linked List): Структура от данни, състояща се от поредица от елементи, наречени възли, където всеки възел съдържа данни и референция (връзка) към следващия възел в последователността.
* Хеш таблица (Hash Table): Структура от данни, която съхранява асоциации между ключове и стойности, позволявайки бърз достъп до елементите чрез техните ключове.
* Регулярен израз (Regular Expression): Стринг, който описва или съответства на набор от стрингове, следвайки определени синтактични правила. Използва се за търсене, замяна или проверка на текст.
* База данни (Database): Организирана колекция от данни, обикновено електронно съхранена в компютърна система. Позволява лесен достъп, управление и актуализация на данните.
* SQL (Structured Query Language): Стандартизиран език за управление на бази данни, използван за извличане, актуализиране и управление на данни в релационни бази данни.
* Инкапсулация (Encapsulation): ООП концепция, която ограничава директния достъп до някои компоненти на обект и крие детайлите на реализацията на обекта от потребителя.
* Полиморфизъм (Polymorphism): Възможността на един интерфейс да бъде изпълнен по различни начини; в ООП, полиморфизмът позволява на методи с едно и също име да имат различни реализации за различни класове.
* Git: Разпределена система за управление на версии, която се използва за проследяване на промените във файловете и координиране на работата между няколко човека по един проект.
* Изключение (Exception): Събитие, което се случва по време на изпълнението на програма, прекъсващо нормалния поток на инструкциите. Изключенията често се обработват с помощта на специален код, за да не доведат до крах на програмата.
* Фреймуърк (Framework): Предварително написан код, който програмистите могат да използват за разработване на приложения. Фреймуъркът диктува структурата и потока на приложението и често предоставя набор от инструменти и библиотеки за улесняване на разработката.
* Библиотека (Library): Колекция от предварително написан код (функции, класове, обекти), който програмистите могат да използват в своите програми, за да спестят време, като използват вече разработен функционал.
* Интеграционно тестване (Integration Testing): Тестов процес, който проверява взаимодействията и интерфейсите между компоненти или системи.
* Юнит тест (Unit Test): Тип тест, който проверява отделни компоненти на софтуера за грешки, като обикновено тества функции или методи.
* IDE (Integrated Development Environment): Програма, която предоставя обширни удобства на програмистите за разработка на софтуер. IDE обикновено включва редактор за код, компилатор и/или интерпретатор.
* Компилатор (Compiler): Програма, която превръща кода, написан в един програмен език, в друг език, обикновено машинен код, който компютърът може да изпълнява.
* Интерпретатор (Interpreter): Програма, която изпълнява инструкции написани в програмен или скриптов език, като ги превежда в машинен код едновременно с тяхното изпълнение, вместо да ги компилира предварително.
* Асинхронно програмиране (Asynchronous Programming): Модел на програмиране, който позволява изпълнението на задачи да се извършва паралелно с основния поток на програмата, без да го блокира.
* Синхронизация (Synchronization): Процес на координиране на едновременно изпълнение на процеси или нишки, за да се осигури