Sunt Panait Stefan Andrei, am 21 de ani, locuiesc in Ilfov in comuna peris. In momentul de fata sunt student anul 3 ACS-UPB. Am urmat liceul Ion Neculce mate-info, de mic am stiut ca acesta este drumul pe care vreau sa il urmez. Sunt o persoana pasionata de informatica si asta este unul din motivele pentru care am ajuns sa urmez aceasta facultate. Sunt o persoana motivata si determinata acest lucru ma face sa dau randament in ceea ce intreprind. Inca de mic am fost si sunt pasionat de sport de la varsta de 7 ani am facut fotbal pana la varsta de 18 ani, acest lucru m-a invatat sa fiu un bun coechipier. In momentul de fata imi place sa am o viata activa. Din acest motiv merg regulat la sala. Datorita faptului ca la facultate am avut proiecte mai mult singur sunt o persoana autonoma. Ca experienta in lumea muncii am avut o experienta in afara tarii in Franta care nu a fost in domeniu.

Can Interview -questions

OOP principii : 4

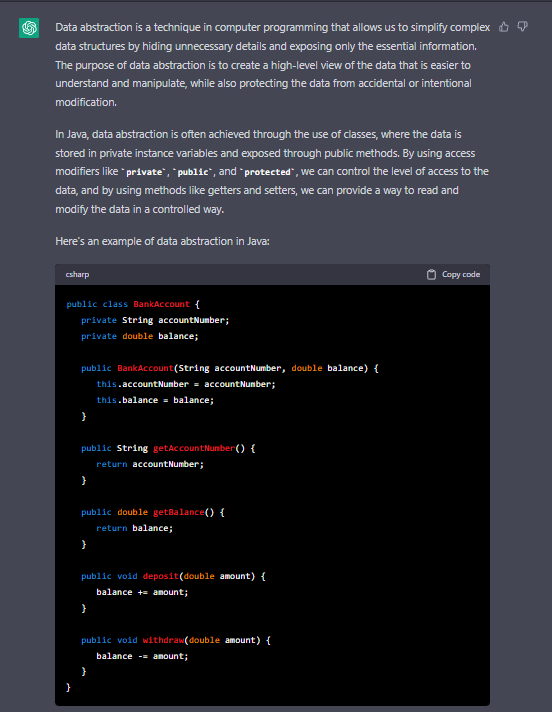
-Abstractizare

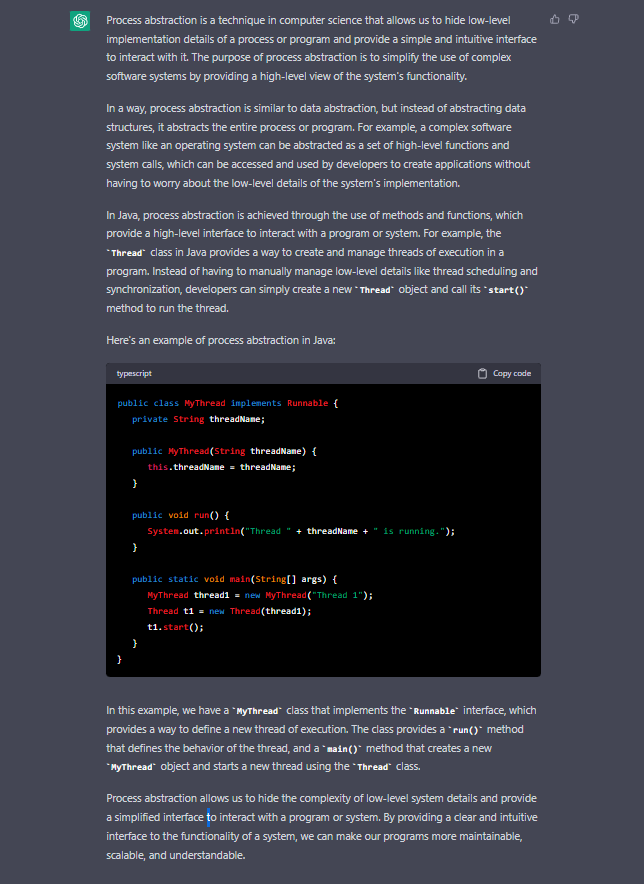
Concept ce ne lasa sa creeam clase abstracte si interfete care definesc un set de metode si proprietati fara a oferii o implementare.

Ascunde datele care nu sunt necesare si sublinieaza feature urile esentiale.

ABSTRACTIZARE - a arata doar feature-rile esentiale userului fara a intra in detalii.

Data abstraction



Process abstractrion 

-INCAPSULARE- mecanism prin care datele si metodele care lucreaza pe acele date sunt impachetate pentru a forma o unitate.

-putem ascunde membrii de date de clasa (private), folosite indirect prin metode.

(Pot avea getteri si setteri care initializeaza chiar si altceva.)

https://ro.myservername.com/unde-sa-gasesti-si-sa-l-prinzi-pe-nymble-in-pokemon-scarlet-violet

-Mostenire- mecanism de refolosire a codului specific limbajelor orientate obiect si preprezinta posibilitatea de a a defini o clasa care extinde o alta clasa deja existenta. Ideea de baza este de a prelua functionalitate si de a adauga una noua sau de a amodela pe cea existenta.

-Polimorfism descrie un patern in care clasele au functionalitate diferita in timp ce au aceeasi superclasa sau interfata .

Type 2: Runtime polymorphism

It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. This type of polymorphism is achieved by Method Overriding. Method overriding, on the other hand, occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.

Method Overloading: When there are multiple functions with the same name but different parameters then these functions are said to be overloaded. Functions can be overloaded by changes in the number of arguments or/and a change in the type of arguments.

Modificatori de acces :

-private reduce la a fi vizibila decat in clasa respectiva

-default accesibil decat in pachetul respectiv, dar nu in afara pachetului

-protected accesibil doar in acel pachet sadar si daca este mostenit in afara pachetului.

-public poate fi accesat de oriunde un membreu al clasei.

DESIGN PATTERNS: solutii reutilizabile a probleme comune in diferite arii soft

1: creational patterns : ex FACTORY si SINGLETON

2: STRUCTURAL PATTERNS: adapter, decorator, facade

3: BEHAVIORAL patterns: observer, strategy, command

FACTORY: da un mod de a creea obiecte fara a specifica exact clasa din care fac parte.

Observer: Defineste o relatie de dependenta intre obiecte astfel incat cand un obiect isi schimba starea, toti dependentii lui sunt notificati si actualizati automat.( un obiect numit subiect notifica observerii).

import java.util.ArrayList;

import java.util.List;

}

Strategy DP : Lasa userul sa defineasca o familie de algoritmi sau comportamente incapsulate ca un obiect si le face interschimbabile la runtime.

EX:

HTTP (HYPERTEXT TRANSFER PROTOCOL)

Functioneaza request reponse. Clientru face un request si serveru raspunde.

Transferul de date se face prin TCP.

Deci practic cand fac google.com ii spun sa deschida o comunicatie tcp catre google.com. Dupa ce conexiunea TCP e facuta clientu (browser) face un get request, iar serverul raspunde cu ce trebuie afisat (site). Dupa ce serverul a trimis raspunsul acesta inchide conexiunea.

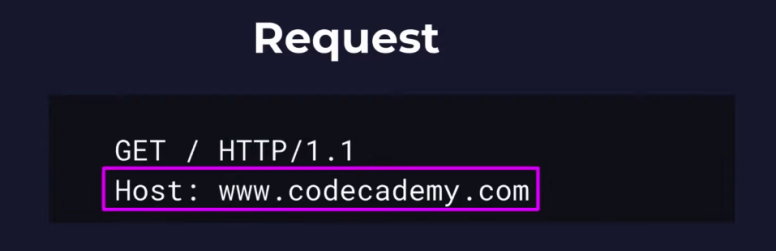
Get: intoarce ceva de la server

POST: da date serverului

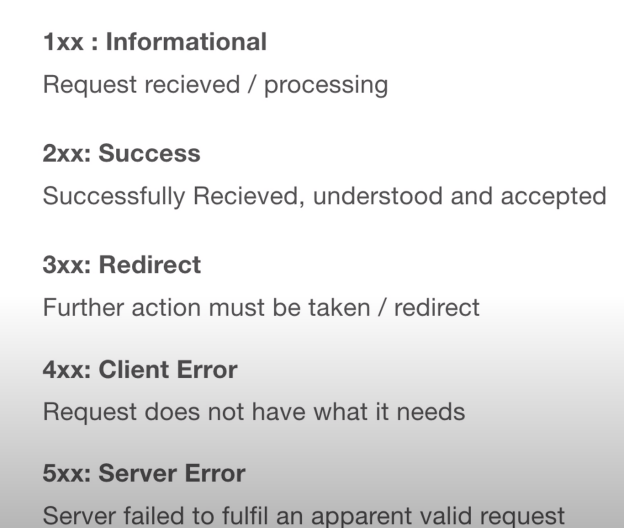
PUT: updateaza date deja pe server

Delete: sterge ceva de pe server.

Patch: updateaza partial date deja pe server.



Status codes



CRUD = CREATE READ UPDATE DELETE

API - aplication programming interface - interfata pe care un program o foloseste sa vorbeasca cu alt program

REST API - representational state transfer - comunicare si restful service. Beneficii - simple si standardizate, sacalbil si fara stare, perfurmanta buna suparta cache.

Request: operatie, endpoint, param/body, header.

Response: de obicei in json.

Complexitati:

Vector - insert O(1) dar daca trebuie realocat O(n)

-deleting worst case O(n) dar daca e la final O(1)

Linked List: insert beginning O(1) arbitrary O(n)

-delete O(1) first si last rest n

-cautare O(n)

Quicksort O nlogn avarage si cel mai rau n^2

Binary search log n;

What is DOM - Document Object Model - Tree-like representation of the HTML document.

Each HTML element is represented as a JS object, which allows JS to access HTML elements.

-DOM objects contain methods and proprieties to manipulate them: change text and HTML attributes,insert new HTM elements, etc.

Promise = future value bcs js takes time to respond.

