یادگیری جاوا

در این دوره آموزشی هدف آن است که از طریق انجام پروژه های گوناگون مباحث متعدد جاوا هم از حیث مفهوم و هم از حیث کاربرد یاد گرفته شود. پروژه در چندین فاز پیش می رود که در هر فاز، سرفصلهای آموزشی آن مشخص شده است.

توجه: نیازی نیست توابع به طور کامل و با جزییات پیاده سازی شوند. در این فاز صرفا هدف یادگیری مفاهیم هست نه کارکرد کامل سیستم. مثلا میتوانید در توابع display به چاپ یک پیام متناسب بسنده کنید و از پیاده سازی جزییات صرفنظر کنید.

پروژه: پیاده سازی یک سیستم مدیریت کتابخانه (Library Management System)

فاز ۲: استفاده از مفاهیم شی گرایی وتحلیل انواع ساختمان داده و الگوریتم

نسخه ۲٫۱- افزودن انواع دیگر به کتابخانه

در این نسخه میخواهیم میخواهیم این کتابخانه به جزء کتاب نگهداری و مدیریت مجله، پایان نامه و مرجع را نیز انجام دهد (برای هر یک از این انواع حداقل ۳ فیلد متناسب در نظر بگیرید). تمامی این انواع تابع display دارند که برای کاربر اطلاعات آن را نمایش می‌دهد. همچنین قابلیت جستجو روی تمامی این انواع و فیلدهایشان ممکن باشد. برای طراحی و پیاده سازی این ویژگی‌ها لازم است از مفاهیم Object Oriented Programming (OOP) به درستی استفاده شوند.

برچسب‌ها: (Inheritance, Polymorphism, Aggregation, Association, Abstraction, Interface, Encapsulation)

نسخه ۲.۲- تحلیل ساختمان داده و الگوریتم

برای نگهداری و پردازش آیتم‌های مختلف کتابخانه با فرض اینکه در فاز بعدی روال تغییر وضعیت کتاب ها (امانت یا موجود) را هم میخواهید اضافه کنید، انواع Collectionهای جاوا را بررسی کنید. با توجه به اینکه روی آیتم ها نیاز به عملیات درج، حذف، جستجو، مرتب سازی و فهرست کردن کتابها داریم بهترین ساختمان داده و الگوریتم های جستجو و مرتب سازی را بیان و پیاده کنید.

برچسب‌ها: (Complexity Time ،Data Structure ،Searching & Sorting Algorithms)

نسخه ۲.۳ - طراحی سیستم امانت‌گیری و بازگشت کتاب:

در نظر گرفتن متدهایی برای امانت گرفتن و بازگشت کتاب‌ها از جمله بررسی وضعیت موجودی کتاب‌ها و مدیریت آن‌ها، افزودن ویژگی‌های اضافی مانند ثبت تاریخ بازگشت کتاب‌ها، نمایش وضعیت کتاب‌ها، و نمایش کتاب‌های امانت‌رفته. (در این نسخه کماکان فرض میشود که سیستم تک کاربری هست و تعامل با کاربر از طریق رابط کاربری متنی برای امانت‌گیری، بازگشت کتاب‌ها و مشاهده وضعیت منابع) می‌باشد.

روابط بین کلاسها و جزییات فیلدها و متدهای آنان را در نمودار Class Diagram رسم کنید (میتواند روی کاغذ هم باشد.)

برچسب‌ها: (Class Diagram،)

نسخه ۲٫۴- بررسی اصول SOLID

پیاده سازی خود را با اصول SOLID تحلیل کنید و برآورده کردن قواعد آن و یا نقض آنها را شفاف کنید.

برچسب‌ها: (SOLID)

آموزش جاوا (سطح مقدماتی ۲)

مفاهیم OOP & Interfaces

برای یادگیری مفاهیم پایه شی گرایی در جاوا سایت javatpoint مناسب است. مباحث Inheritance، Polymorphism، Association و Aggregation را از این سایت یاد بگیرید. یادگیری مفاهیم Abstraction و Encapsulation هم کلیدی است. در جاوا از Interfaceها استفاده زیادی می‌شود که میتوانید از این لینک بیشتر آشنا شوید. تفاوت Interface و Abstract Class در این صفحه بیان شده است.

برای مرور نکات این مباحث در آموزش ویدئویی این دوره در یودمی در کدهای درس ۰۷۳ تا ۰۷۹ مناسب است. (این دوره برای سایر مباحث هم دوره مختصر و مفید و خوبی هست. میتونید استفاده کنید)

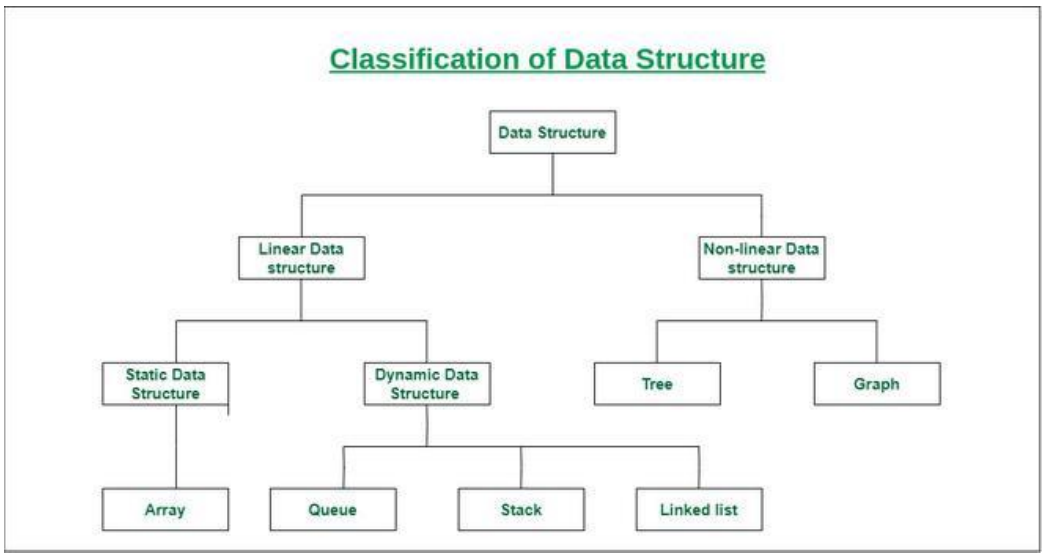
بعد از این آموزش باید بتوانید به این سوالات پاسخ دهید:

- منظور از Encapsulation چیست و چرا در شیء گرایی مهم هست؟
- مفهوم Polymorphism چیست و مثالی از آن؟
- تفاوت Overloading با Overriding ؟
- تفاوت Composition با Inheritance؟
- تفاوت Interface با Abstract Class؟
- اگر nested static class داشته باشیم چه محدودیتی دارد؟

Data Structures & Java Collections

جاوا ابزارهایی برای کار با ساختمان داده‌ها دارد. در این مرحله باید مفاهیم زیر را یاد بگیرید:

- آرایه‌ها (Arrays)
- لیست‌های پیوندی (Linked Lists)
- پشته‌ها و صف‌ها (Stacks & Queues)
- درخت‌ها و گراف‌ها (Trees & Graphs)
- جدول‌های هش (Hash Tables)



میتوانید از این بخش javatpoint و یا tutorialspoint فرا بگیرید سعی کنید بر تمامی جزییات این ساختمان داده ها از جمله پیچیدگی زمانی اعمال اصلی و نحوه ذخیره سازی و کاربردشان مسلط شوید خصوصا مثل hashset، treeset.
همچنین در این فیلم یوتیوب نیز Java Collections با کد توضیح داده شده اند.

وجه دیگر ساختمان داده آن است که با الگوهای حل مساله مثل بازگشتی، تقسیم و حل و حریصانه و همچنین الگوریتم‌های معروف و پرتکرار مانند جستجو و مرتب‌سازی (Searching & Sorting Algorithms) مسلط باشید تا برنامه‌ها را به طور صحیح و بهینه، پیاده سازی کنید. به طور مثال الگوریتم‌های جستجوی Linear, binary, modified binary search و الگوریتم‌های مرتب سازی Insertion, selection, bubble, cyclic, merge, quick, count, radix, and heap sort و الگوهای حل مساله مثل بازگشتی، تقسیم و حل و حریصانه را بدانید. از این بخش tutorialspoint میتوانید این دو مبحث را یاد بگیرید. در این صفحه از geeksforgeeks هم، مثال‌های ساده‌ای برای تحلیل پیچیدگی زمانی بیان شده است.

بعد از این آموزش باید بتوانید به این سوالات پاسخ دهید:

- **تفاوت بین پشته و صف را توضیح دهید. چه زمانی از هر کدام استفاده می کنید؟**
- **پیاده سازی جستجوی باینری در یک آرایه مرتب شده**
- **یک تابع بازگشتی برای محاسبه دنباله فیبوناچی بنویسید.**
- **نحوه عملکرد یک جدول هش و پیچیدگی زمانی آن برای عملیات های مختلف را شرح دهید.**
- **یک مثال از روش حل مساله divide & conquer و یک نمونه از greedy عنوان کنید.**

مفاهیم SOLID

یادگیری مفاهیم شیء گرایی برای طراحی صحیح یک سیستم OOP کفایت نمی‌کند. قواعد و اصولی تحت عنوان SOLID مطرح هست که بیان می‌کند در طراحی کلاس‌های لازم برای پیاده سازی یک برنامه شیء گرا به چه نکاتی لازم است توجه شود. این لینک منبع مناسبی برای فراگیری این مفهوم هست. همچنین در این لینک مثال‌های خوبی از نقض هر یک از این اصول و چگونگی اصلاح آن ذکر شده است.

بعد از این آموزش باید بتوانید به این سوالات پاسخ دهید:

- **قواعد SOLID را بیان کنید و برای هر یک مثال بزنید.**