네이버 면접준비

1. 자바 및 언어

1.1.1자바란? :

1.1.2자바의 장점

1.1.3JAVA 바이트 코드와 다른 언어와의 차이는 무엇인가 ?

자바는 가상버신 바이트 코드로 컴파일 하며 실행시키려면 가상머신이 필요 하지만 c 나 cpp의 경우 각머신에 맞는 기계어로 컴파일한다.

1.1.4 JVM

자바 바이트 코드를 os에 맞게 해석해주는 가상 머신

1.1.5가비지컬렉션

불필요한 메모리를 알아서 정리해주는 것 자바에서는 명시적으로 불필요한 데이터를 표현하기위해 일반적으로 null을 선언

1.1.6 자바 thread : synchronized 키워드를 제공해 스레드간 동기화를 시켜 data의 thread-safe를 가능케 합니다.

기본적으로 자바에서 제공하는 collection들의 경우 thread safe하지 않지만 Syncronized 명령어를 사용해서 thread-safe하게 설계도 가능하다.

Syncronized 명령어에서 변수에서 사용할때와 메서드에 사용할때의 차이 ??

1.1.7 제네릭

제네릭은 클래스 내부에서 사용할 데이터 타입을 외부에서 지정하는 기법을 의미

1.1.8 자바 static

1.1.9 업캐스팅과 다운캐스팅

캐스팅은 타입을 변환하는 것을 의미 자바에서 상속관계에 있는 부모와 자식클래스간에는 서로간 형변환이 가능

1.2 C# 관련

1.2.1

1.3 파이썬 관련

1. 자료구조

2.1 리스트

2.1.1 ArrayList와 LinkkedList 차이

2.2 스택 큐

2.2.1 스택 : 선입 후출의 자료구조 리스트로 구현하는 것보다 배열로 사용하는 것이 더 좋다

2.2.2 큐 : 선입 선출의 자료구조로 리스트로 구현한느것이 배열로 사용하는 것이 더 좋다. 그 이유는 배열로 구현할경우 배열을 한칸씩 당겨줘야하는 연산이 필요하기 때문에 비효율적이다.

2.3 그래프

2.4 트리

노드와 간선으로 으루어져 있고 사이클을 이루이 않는 구조 특징으로는 하나의 루트노드로 이루어져 있으며 루트노드는 0개 이상의 자식노드를 가지고 있다. 트리에는 사이클이 존재할 수 없다.

2.4.1 이진트리 : 각 노드가 최대 두개의 자식을 갖는 트리

B+Tree AVL 트리 RedBlack Tree AVL 트리

2.4.2 트라이

2.5 셋

데이터를 비순차적으로 저장할 수 있는 순열 자료구조 비순차적이라 데이터 삽입 순서대로 저장되지 않아 일반적으로 순서를 기대할 순 없음 중복을 허용하지 않아 같은값을 삽입하면 마지막에 삽입한 값 하나만 저장됨

2.5.1 HashSet

HashCode를 사용해서 저장하기 전 먼저 hashCode를 얻어낸 다음 저장되어 있는 객체들의 해시 코드와 비교한 뒤 같은 해시 코드가 있다면 객체를 비교한뒤 동일한 객체가 있다면 중복 저장을 하지 않습니다.

시간 복잡도는 추가 contains next 의 O(1) O(1) O(h/n) 의 시간복잡도를 가짐

2.5.2 TreeSet

Tree 구조(Red-Black Tree)를 사용하여 Set을 사용하는 경우 장점은 자동 정렬을 해준다는 장점이 있습니다. 기본적으로 오름차순으로 정렬을 하고 추가 삭제에는 시간이 많이 걸림

시간 복잡도는 모드 Logn 의 시간복잡도를 가짐

2.5.3 LinkedHashSet

HashSet의 저장 순서를 보장해주는 Set 기본적인 구조는 LinkedHashMap 을 상속받아 사용하게 됨

시간복잡도는 모두 O(1) 을가짐

HashSet > TreeSet > LinkedHashSet 순서로 성능의 차이를 보임

기본적으로 Thread -safe 를 보장하지 않음 즉 동기화가 되지않음

2.6 맵

2.6.1 HashMap TreeMap LinkedHashMap

2.7 힙

2.8 우선순위 큐

1. 알고리즘

3.1 정렬 알고리즘

3.2 탐색알고리즘

3.3 최단경로

3.4 다이나믹 프로그래밍

DFS BFS 다익스트라 Floyd-WarShall DP 브르투포스 정렬 알고리즘 등에 대한것들

Stable한 솔트란?

1. DB

Index Transaction Entity 나중을 위한 column 등과 같은 기본 개념 및 추후 가능하다면 디비 튜닝까지??

1. 네트워크

TCP/IP UDP 차이 3-way-handshaking 등

1. 프레임워크

6.1Spring을 주로 다룰 예정 DI/DL JPA 등등 BEAN

6.2. net

6.3 django

1. 웹

7.1 Http란?

인터넷에서 데이터를 주고 받을 수 있는 프로토콜

특징은 비연결성을 가지고 있다 비연결성은 클라이언트가 서버와 한번 연결을 맺은 후 클라이언트 요청에 대해 서버가 응답을 마치면 연결을 끊어버림

HTTP는 인터넷 상에서 불특정 다수의 통신환경을 기반으로 설계됨 만약 서버에서 다수의 클라이언트와 연결을 계속 유지해야한다면 많은 리소스가 필요하지만 연결을 유지하지않기 때문에 리소스가 줄어들어 더 많은 연결을 할 수 있음

하지만 단점으로 서버는 클라이언트를 기억하고 있지 않아 동일한 클라이언트 모든 요청에 대해 매번 새로운 연결을 시도/해제의 과정을 거쳐야 하므로 연결 / 해제에 많은 오버헤드가 발생한다는 단점 이에 대한 해겨랙으로 keepalive 속성을 사용할 수 있지만 이것을 사용하면 상대방의 안부를 묻기위해 패킷을 주기적으로 보내는 것이지만 프로세스 수가 기하급수적으로 늘어나 메모리를 많이 사용하게됨

무상태성의 특징 때문에 서버는 클라이언트를 식별할 수가 없어 이를 해결하기 위해 쿠키와 세션의 개념이 도입됨

쿠키 :

세션 :

7.2 URL 과 URI 의 차이는 ?

RESTFul 한 API란 무엇일까

7.3 http 동작 과정

1. 사용자가 웹 브라우저에 URL 주소 입력

2. DNS 서버에 웹 서버의 호스트 이름을 IP 주소로 변경 요청

3. 웹 서버와 TCP 연결 시도

3 -way HandShake : 클라이언트 – 서버간 신뢰성있는 연결을 하기 위해 3번의 패킷 교환 과정

SYN 클라이언트가 서버로 임의로 생성한 시퀀스 번호를 전달

SYN ACK 서버는 클라이언트에서 전달한 시퀀스를 +1 시켜서 전달한다.

ACK : 클라이언트가 서버에서 전달해준 시퀀스를 +1시켜서 다시 전달함(서버 클라이언트간 패킷 교환이 정상적으로 이루어졌다는 신호)

4. 서버에게 GET POST PUT 등과 같은 명령 전송

5. 서버가 클라이언트에게 데이터(웹 문서) 회신

6.서버 – 클라이언트간 연결 해제

웹브라우저 동작과정

REST한 방식이뭔지

GET PUT POST DELETE TRACE OPTIONS 상세 분류

Http 상태코드

Https란 ?

1. 프로젝트
2. 컴퓨터 CS

9.1 프레임워크와 라이브러리 차이는?

9.1.1 프레임워크 : 소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 형태의 클래스들을 제공하는 것으로 애플리케이션의 틀과 구조를 결정할 뿐 아니라 개발된 개발제의 코드를 제어하여 구체적이고 확장 가능한 기반코드를 가지고 있고 설계자가 의도하는 여러 디자인 패턴의 집합으로 구성되어 있다.

9.1.2 라이브러리 : 자주 쓰일 만한 기능들을 모아 놓은 유틸(클래스)들의 모음집

프레임워크에는 라이브러리에 뼈대가 되는 클래스들과 그 클래스들의 관계로 만들어진 일종의 설계의 기본 틀이 추가된다 프레임워크는 제작자가 기초로 해서 만들라는 기본 틀이라는 것을 제공하고 이를 사용해 확장해 가며 입맛대로 개발하면 되는 것

* 1. 프로세스와 스레드의 차이
     1. 프로세스 : 메모리에 올라와 실행되고 있는 프로그램의 인스턴스
        1. 프로세스 메모리 구조 :

Code 영역 : 프로그램을 실행시키는 실행 파일 내의 명령어들이 올라가는공간

Data 영역 : 전역변수 static 변수의 할당

Heap 영역 : 동적할당을 위한 메모리 영역

Stack 영역 : 지역변수 , 함수 호출시 전달되는 인자(파라미터)를 위한 메모리 영역

* + - 1. 프로세스 상태변화

프로세스 상태에는 ready , blocked running

9.2.1.3프로세스 스케줄링

CPU 할당 순서 및 방법을 결정하는 일 우선순위 알고리즘과 라운드 로빈 알고리즘을 혼합해서 스케줄링하는 것이 보편정

선점 스케줄링:

비선점 스케줄링

* + 1. 스레드 : 프로세스 내에서 할당받은 자원을 이용해 동작하는 실행단위
       1. 멀티스레드
       2. DeadLock

9.3. OOP 개념

현실 세계를 프로그래밍으로 옮겨와 현실 세계의 사물들을 객체로 보고 그 객체로부터 개발하고자 하는 특징과 기능을 뽑아와 프로그래밍 하는 기법을 의미 OOP로 코드를 작성하면 재사용성과 변형 가능성을 높일 수 있음

9.3.1 객체지향언어의 특징

캡슐화 : 데이터와 데이터를 처리하는 함수를 하나로 묶는 것 캡슐화된 객체의 세부 내용이 외부의 은폐되어 변경이 발생할 때 오류의 파급효과가 적다. 캡슐화된 객체들은 재사용이 용이하다.

정보은닉 : 캡슐화에서 가장 중요한 개념으로 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것

추상화 : 불필요한 부분을 생략하고 객체의 속성중 가장 중요한 것에만 중점을 두어 개략화하는 것 즉 모델화

상속성 : 이미 정의된 상위 클래스의 모든 속성과 연산을 하위클래스가 물려받는 것

다형성 메시지에 의해 개체 가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체가 가지고 있는 고유한 방법으로 응답할 수 있는 능력을 의미 객체들은 동일한 메소드명을 사용해서 같은 의미의 응답을 함.

9.3.2 SOLID 원칙

Single Respongsiblity Principle( 단일책임 원칙) : 소프트웨어의 설계 부품(클래스 함수)는 단 하나의 책임만을 가져야 한다.

설계가 잘된 프로그램은 기본적으로 새로운 요구사항과 프로그램 변경에 영향을 받는 부분이 적다 즉 응집도는 높고 결합도는 낮은 프로그램이다. 만약 한 클래스가 수행 기능이 많다면 클래스 내부의 함수 끼리 강한 결합을 발생할 가능성이 높아진다 이에 따라 유지보수 비용이 증가하게 되므로 책임을 분리시켜야한다.

Open-Closed Principle (개방 폐쇄 원칙) : 기존의 코드를 변경하지않고(close) 기능을 수정하거나 추가 할 수 있도록 설계 해야한다.

즉 인터페이스의 설계가 필요하다는 것을 의미

Liscov Substitution Principle (리스코프 치환원칙 ) : 자식클래스는 부모클래스에서 가능한 행위를 수행 할 수 있어야한다. 부모클래스와 자식 클래스 사이의 행위에는 일관성이 있어야한다는 원칙이며 이는 객체 지향 프로그래밍에서 부모 클래스의 인스턴스를 사용해도 문제가 없어야한다는 것을 의미

상속 관계에서 일반화 관계 IS A 관계가 성립되어야 한다.

Dependecncy Inversion Principle(의존 역전 법칙) : 의존 관계를 맺을 때 변화하기 쉬운 것 보단 변화하기 어려운 것에 의존해야 한다는 원칙이다. 여기서 변화하기 어려운 것이란 즉 추상적인 것을 의미 변화하기 쉬운 것은 구체화된 클래스를 의미하고 어려운 것은 추상클래스나 인터페이스를 의미한다.

Interface Segregation Principle 인터페이스 분리원칙 한 클래스는 자신이 사용하지 않는 인터페이스는 구현하지 말아야 한다. 하나의 일반적인 인터페이스 보다는 여러 개의 구체적인 인터페이스가 낫다. 즉 사용하지 않는 기능에는 영향을 받지 말아야한다는 것을 의미

9.3.3 다형성 상속

9.4 디자인패턴

9.4 Parameter 와 Argument 차이

Parameter = 함수를 선언할 때 사용된 변수

Argument : 함수가 호출 되었을 때 함수의 파라미터로 전달된 실제 값

9.5 Call By Value Call By reference 차이

Call by value = 인자로 받은 값을 복사하여 처리하는 방식 넘어온 값을 증가시켜도 원래의 값이 보존된다.

값을 복사하여 넘기기 때문에 메모리 사용량이 늘어난다.

Call by reference : 인자로 받은 값의 주소를 참조하여 직접 값에 영향을 주는 방식

값을 복사하지 않고 직접 참조하기 때문에 속도가 빠르다.

원래의 값에 영향을 주는 리스크가 존재

10 . 인공지능

11. 빅데이터

12.. 블록체인

13그동안 나왔던 질문들 복기

컴파일 언어와 인터프리터언어의 차이는 ?

14.추가적으로 궁금한것들 아래 달아주세염 :