mainPage
-main: Scene -results: Scene -page: Scene -width: int -height: int -search: Button -querry: TextField -logo: Image -background: Background -engine: BM25 -resultsList: Vector<SimpleEntry<String, Double>> -recent: VBox -mainLayout: VBox linkColor: String
+start(window: Stage): void +stop(): void +refreshRecents(window: Stage): void +refreshResults(q: String, window: Stage): void +setPageContents(htmlContent: String, window: Stage): void

Term
-id: int -stem: String
+Term(s: String, i: int) +getId(): int +getStem(): String

TFIDFCOS
-resultLimit: int -documentCount: int -avgDocLength: double -indexCorpis: IndexReader -stemmer: Stemmer -results: ArrayList<SimpleEntry<String, ArrayList<double>>> -cosineSim: ArrayList<SimpleEntry<String, Double>>
+TFIDFCOS() +stem(word: String): String +start(q: Term []): void +close(): void +printResults(): void +getScore(w: Term, v: ArrayList<String>): double +getCosSum(d: ArrayList<Double>, q: ArrayList<Doulbe>): double +getVec(v: ArrayList<String>, list: Term []): ArrayList<Double>

TFIDF
-resultLimit: int -documentCount: int -avgDocLength: double -indexCorpis: IndexReader -stemmer: Stemmer -results: Vector<SimpleEntry<String, Double>>
+TFIDF() +stem(word: String): String +start(q: Term []): void +close(): void +printResults(): void +getScore(d: IndexedDoc, q: Term []): double +okapiTF(w: Term, d: IndexedDoc): double

Stemmer
-b: char [] -i: int -i_end: int -j: int -k: int -INC: int
+Stemmer() +add(ch: char): void +add(w: char [] , wLen: int): void +toString(): String +getResultLength(): int +getResultBuffer(): char [] -cons(i: int): boolean -m(): int -vowelinstem(): boolean -doublec(j: int): boolean -cvc(i: int): boolean -ends(s: String): boolean -setto(s: String): void -r(s: String): void -step1(): void -step2(): void -step2(): void -step4(): void -step5(): void -step6(): void +stem(): void

MetaData
-description: String -keywords: String
+MetaData(k: String, d: String) +getKeywords(): String +getDescription(): String +setKeywords(k: String): void +setDescription(d: String): void

IndexedDoc
-contents: HashMap<Term, Integer>
+IndexedDoc(c: HashMap<Term, Integet>) +getContents(): HashMap<Term, Integer> +getTotalTerms(): int +getLength(): int +termCount(t: Term): int +containsTerm(t: Term): boolean +getTerms(): Term []

IndexWriter
+addIndex(u: String, n: String): void

IndexReader
-fileDirectory: String -index: HashMap<String, IndexedDoc> -schema: String -pattern: Pattern -stemmer: Stemmer
+IndexReader() +start(): void +close(): void +getMap(): HashMap<String, IndexDoc> +getDocLenth(fileName: String): int +getDocFreq(t: Term): int +getTermFreq(fileName: String, t: Term): int +getAvgDocLength(): double +getDocCount(): int +getTermNo(file: String): int +getTermTotal(): int +getTotalTermFrequency(t: Term): int -processWord(word: String): String +getTerms(file: String): Term [] +getTermByDoc(d: IndexedDoc): Term [] -addFile(file: String): void -readFromIndex(): boolean +addIndex(u: String): void

BM25
-resultLimit: int -documentCount: int -avgDocLength: double -indexCorpus: IndexReader -stemmer: Stemmer -results: Vector<SimpleEntry<String, Double>>
+BM25() +stem(word: String): String +start(q: Term []): void +close(): void +printResults(): void +getScore(d: IndexedDoc, q: Term []): double +getP2(tf: int, d: IndexedDoc): double +getP3(tf: int): double

Crawler
-good: boolean -seed: String -pageLimit: int -crawledPages: Hashtable<String, String>
+Crawler(seedUrl: String, PageLimit: int) +isGood(): boolean +start(url: String, depth: int): Hashtable<String, String> -downloadPage(u: String): String -getLinks(url: String): Elements -getImages(url: String): Elements -getConnection(url: String): Document -getMetaData(url: String): MetaData -getHtml(url: String): String -stripFileName(path: String): String -isUrlValid(url: String): boolean -processLink(link: String, base: String): String -encodeUrl(string: String): String

CREATED WITH YUML