Boya Zhou, Yuting Liang
Project 2 for Database Management System
Nov 1st, 2015

Introduction

An indexing mechanism. Our index has just three interfaces:
1 void Put(string key, Number data_value); or void Put(string key, string data_value); adds the index entry.
2 string Get(Number data_value); or string Get(string data_value); retrieves the key given the index and
3 void Remove(string key); deletes the index.

key in the problem statement is the same concept as key in project 1. Several database books also refer to it as rid.

data_value is the indexing attribute (or a list of attributes), typically expressed in the command `CREATE INDEX <indexname> ON <table_name> (<index_attr_list>)`.

In the first test, assume the provided table represented as a comma-separated file with the schema: Movies(Title, Year, Format, Genre, Director, Writer, Country, Studio, Price, Catalog No). The index in this case is defined as `CREATE INDEX movieInd ON Movie(year, format)`. The data value could be a composite string. For the movie Ghost for example, it could be `"1990|laserdisk"`. Using the index you created,
1 Find all DVD movies made in 1977.
2 Find all VHS movies made in 1990.
3 Find all DVD movies made in 2001.

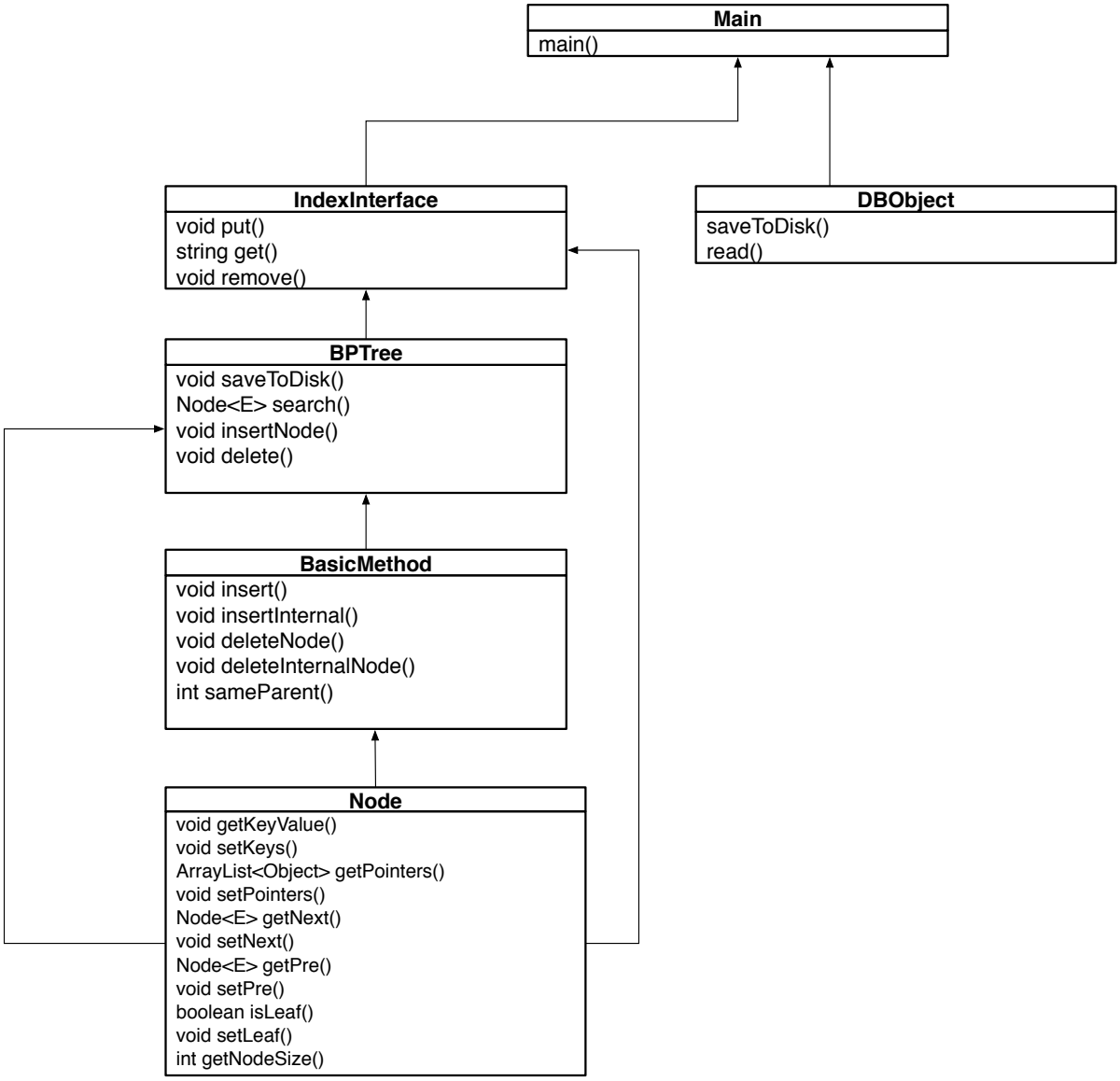
In the second test, assume the same table and index `yrInd ON Movie(year)`.
1 Find all movies made in 2000.
2 Find all movies made in 2005.
3 Find all movies made in 2010.

Technically the above three could be counted as five interfaces because of Number vs. string. The Number inputs only apply if you are working with B+ Trees. You have a choice of index type: B+ tree or hash. Additional assumptions will be required to complete this assignment and limit the scope of the exercise. That is OK, but you must enumerate those assumptions.

Assumption

1.we use the B Plus Tree to finish this project
2.we assume this is a duplicate index, rather than not duplicate index or using one pointer to point to all the same index-record
3.In the leaf nodes, the pointers are used to store primary key directly rather than point tuple in the database
4. the fanout can be any number
5.we make the movies' name as the primary key of the database, the key in interface as well.
6.we change the comma in movie name in movies.txt to "/"

UML Diagram



Design Decisions

In the Node, we just give some fundamental attribute to describe both the leaf node and internal node.

In BasicMethod, we still do not reach the tree level, but we begin to treat internal nodes and leaf nodes differently, also, we give whether two nodes have the same parent.

In BPTree, we reach the tree level, in search method we return the target node, in insert and delete method, we use methods in BasicMethods, when insert, there is possibility to split leaf nodes and internal nodes and height of the tree maybe increase, and when delete, there is possibility elements in leaf nodes will redistribute, if fail, the leaf nodes will merge and also the parent nodes, and the height of the tree maybe decrease. They are more detailed in the program.

In interface, we use the interface given on the website of cs542.

In DBObject, we make the movies.txt into a DB Object, like read it into the memory, and it can also to be saved into CS542.db.

In main, we can test the BPTree, and results are show as follows.

Result

After insert every data_value in the index, we can use index to find the primary key of the database, the results are as follows:

First we find the key using index year and show all the result.

```
---------------Search--------------------
----------------------------------------
get 1977 is Jack the Giant Killer | A Man About the House | No
Children/ No Pets | Roper's Niece |

get 1990 is Ghost | Pacific Heights |

get 2000 is Gladiator | Malena |
```

Second we find the key using composite index year|format and show all the result.

```
---------------Search(composite index)--------------------
------------------------------------------------------------
get 1977|DVD is Not Found!

get 1990|VHS is Pacific Heights |
```

```
get 2000|DVD is Crouching Tiger/ Hidden Dragon | Gladiator | Malena |
```

Third we find the key using year 1990 and we find two results, after we remove one of them, the Pacific Heights, we use the index 1990 to search again, and we can just find Ghost here.

```
---------------Remove--------------------
-----------------------------------------
get original content is Ghost | Pacific Heights |
get new content is Ghost |
```