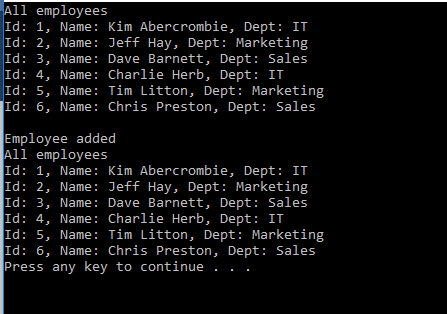
Student: Brian Johnston

Class COP 2362

Assignment 5-3

Print screen:



Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using BinaryTree;

namespace QueryBinaryTree

{

class Program

{

static void doWork()

{

//Construct and populate an instance of the BinaryTree class

Tree<Employee> empTree = new Tree<Employee>(new Employee { Id = 1, FirstName = "Kim", LastName = "Aberrombie", Department = "IT" });

empTree.Insert(new Employee { Id = 2, FirstName = "Jeff", LastName = "Hay", Department = "Marketing" });

empTree.Insert(new Employee { Id = 4, FirstName = "Charlie", LastName = "Herb", Department = "IT" });

empTree.Insert(new Employee { Id = 6, FirstName = "Chris", LastName = "Preston", Department = "Sales" });

empTree.Insert(new Employee { Id = 3, FirstName = "Dave", LastName = "Barnett", Department = "Sales" });

empTree.Insert(new Employee { Id = 5, FirstName = "Tim", LastName = "Litton", Department = "Marketing" });

//Generates an enumerable collection of employeees named allEmployees and then iterates through the collection to diplay the details of each employee

Console.WriteLine("All employees");

var allEmployes = from e in empTree.ToList<Employee>() select e;

foreach (var emp in allEmployes)

{

Console.WriteLine(emp);

}

empTree.Insert(new Employee { Id = 7, FirstName = "David", LastName = "Simpson", Department = "IT" });

Console.WriteLine("\nEmployee added");

Console.WriteLine("All employees");

foreach (var emp in allEmployes)

{

Console.WriteLine(emp);

}

/\*

//Invokes the select method to list the departments found in the binary tree

Console.WriteLine("List of departments");

//var depts = empTree.Select(d => d.Department).Distinct();

var depts = (from d in empTree select d.Department).Distinct();

foreach (var dept in depts)

{

Console.WriteLine("Department: {0}", dept);

}

//Use the Where method to filter the employees and retun only those in the IT department

Console.WriteLine("\nEmployees in the IT department");

//var ITEmployees = empTree.Where(e => String.Equals(e.Department, "IT")).Select(emp => emp);

var ITEmployees = from e in empTree where String.Equals(e.Department, "IT") select e;

foreach (var emp in ITEmployees)

{

Console.WriteLine(emp);

}

//Use the GroupBy method to group the employees found in the binary tree by department

Console.WriteLine("\nAll employees grouped by department");

//var employeesByDept = empTree.GroupBy(e => e.Department);

var employeesByDept = from e in empTree group e by e.Department;

foreach (var dept in employeesByDept)

{

Console.WriteLine("Department: {0}", dept.Key);

foreach (var emp in dept)

{

Console.WriteLine("\t{0} {1}", emp.FirstName, emp.LastName);

}

}

\*/

}

static void Main()

{

try

{

doWork();

}

catch (Exception ex)

{

Console.WriteLine("Exception: {0}", ex.Message);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QueryBinaryTree

{

class Employee : IComparable<Employee>

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Department { get; set; }

public int Id { get; set; }

//Convert object to string representation

public override string ToString()

{

return String.Format("Id: {0}, Name: {1} {2}, Dept: {3}", this.Id, this.FirstName, this.LastName, this.Department);

}

//Method compares employee objects based on the value of the Id field

int IComparable<Employee>.CompareTo(Employee other)

{

if (other == null)

{

return 1;

}

if (this.Id > other.Id)

{

return 1;

}

if (this.Id < other.Id)

{

return -1;

}

return 0;

}

}

}