Технологии и методы программирования.

Рубежный контроль №2

Выполнил: Журавлёв Максим ИУ8-24

Оглавление

[Тип данных: Список 2](#_Toc515240143)

[1. Язык C++ : 2](#_Toc515240144)

[1) Односвязный список: 2](#_Toc515240145)

[2 ) Применение 2](#_Toc515240146)

[2. Язык Java: 4](#_Toc515240147)

[1) ArrayList 4](#_Toc515240148)

[2)Применение 4](#_Toc515240149)

[1. Язык Python: 5](#_Toc515240150)

[1) list 5](#_Toc515240151)

[2) Применение 5](#_Toc515240152)

[Тип данных: Дерево 6](#_Toc515240153)

[1. Язык C++ : 6](#_Toc515240154)

[1)бинарное дерево 6](#_Toc515240155)

[2)Применение 6](#_Toc515240156)

[2. Язык Java : 7](#_Toc515240157)

[1)Бинарное дерево 7](#_Toc515240158)

[2)Применение 8](#_Toc515240159)

[3. Язык Python : 10](#_Toc515240160)

[1) Бинарное дерево 10](#_Toc515240161)

[2)Применение 10](#_Toc515240162)

[Тип данных: Отображение 13](#_Toc515240163)

[1. Язык C++ : 13](#_Toc515240164)

[1)Map 13](#_Toc515240165)

[2)Применение 13](#_Toc515240166)

[1. Язык Java : 14](#_Toc515240167)

[1) map 14](#_Toc515240168)

[2) Применение 14](#_Toc515240169)

[1. Язык Python : 15](#_Toc515240170)

[1)Словарь 15](#_Toc515240171)

[2)Применение 15](#_Toc515240172)

# Тип данных: Список

## 1. Язык C++ :

### 1) Односвязный список:

struct list  
{  
  int field; // поле данных  
  struct list \*ptr; // указатель на следующий элемент  
};

### 2 ) Применение

#include <conio.h>

#include <iostream.h>

struct element

{

int x;

element \*Next;

};

class List

{

element \*Head;

public:

  List() {Head = NULL;}

~List();

void Add(int x);

void Show();

};

List::~List()

{

    while (Head != NULL)

     {

        element \*temp = Head->Next;

        delete Head;

        Head = temp;

     }

}

void List::Add(int x)               {

element \*temp = new element;

temp->x = x;

temp->Next = Head;

Head = temp;

}

void List::Show()

{

element \*temp=Head;

while (temp != NULL)

{

  cout << temp->x << " ";

  temp = temp->Next;

}

}

int main()

{

clrscr();

  int N;

  int x;

  List lst;

   cout << "N = ";

   cin >> N;

        for (int i=0; i<N; i++)

        {

          cout << i+1 << ". x = ";

          cin >> x;

          lst.Add(x);

        }

    lst.Show();

cin.ignore().get();

return 0;

}

## 2. Язык Java:

### 1) ArrayList

public ArrayList(){

new int[16];

}

### 2)Применение

import java.util.ArrayList;

public class Program{

public static void main(String[] args) {

ArrayList<String> people = new ArrayList<String>();

// добавление в список ряд элементов

people.add("Tom");

people.add("Alice");

people.add("Kate");

people.add("Sam");

people.add(1, "Bob"); // добавляется элемент по индексу 1

System.out.println(people.get(1));// получение 2-го объекта

people.set(1, "Robert"); // установка нового значения для 2-го объекта

System.out.printf("ArrayList has %d elements \n", people.size());

for(String person : people){

System.out.println(person);

}

// проверка наличие элемента

if(people.contains("Tom")){

System.out.println("ArrayList contains Tom");

}

// удаление конкретного элемента

people.remove("Robert");

// удаление по индексу

people.remove(0);

Object[] peopleArray = people.toArray();

for(Object person : peopleArray){

System.out.println(person);

}

}

}

## 1. Язык Python:

### 1) list

typedef struct

{

PyObject\_VAR\_HEAD

PyObject \*\*ob\_item; //список указателей на элементы списка

Py\_ssize\_t allocated; //количество выделенной памяти.

} PyListObject;

### 2) Применение

>>> l = []

>>> l.append(1)

>>> l.append(2)

>>> l.append(3)

>>> l [1, 2, 3]

>>> for e in l:

... print e

# Тип данных: Дерево

## 1. Язык C++ :

### 1)бинарное дерево

struct Node                             //Звено дерева

{

   int x;                               //То, что записываем в дерево

   Node \*l,\*r;                          //Это указатели на новые звенья

};

### 2)Применение

#include <iostream>

using namespace std;

struct Node

{

   int x;

   Node \*l,\*r;                      //указатели на новые звенья

};

void show(Node \*&Tree)              //Функция обхода

{

if (Tree != NULL)

{

   show(Tree->l);

        cout << Tree->x;

   show(Tree->r);

}

}

void add\_node(int x,Node \*&MyTree)  //Фукция добавления звена в дерево

{

if (NULL == MyTree)

{

MyTree = new Node;

MyTree->x = x;

MyTree->l = MyTree->r = NULL;

}

                   if (x < MyTree->x)

                      {

                          if (MyTree->l != NULL) add\_node(x,MyTree->l);

                          else

                          {

                              MyTree->l = new Node;

                              MyTree->l->l = MyTree->l->r = NULL;

                              MyTree->l->x = x;

                          }

                      }

                    if (x > MyTree->x)

                      {

                          if (MyTree->r != NULL) add\_node(x, MyTree->r);

                          else

                          {

                              MyTree->r = new Node;

                              MyTree->r->l = MyTree->r->r = NULL;

                              MyTree->r->x = x;

                          }

                      }

}

int main()

{

   Node \*Tree = NULL;

  for (int i=5; i>0; i--) add\_node(i,Tree);

  show(Tree);

  cin.get();

   return 0;

}

## 2. Язык Java :

### 1)Бинарное дерево

public abstract class HashTree<E extends Comparable<E>> {

private Node root = null; //корень

private Node[] nodes; //значение

public HashTree(int capacity) { //выделение памяти

this.nodes = new Node[capacity]; }

public abstract int getElementHash(E element);

…

}

### 2)Применение

public class BinarySearchTree {

    Node root;

    static class Node {

        int key;

        int value;

        Node l;

        Node r;

        Node p;

        public Node(int key, int value, Node p) {

            this.key = key;

            this.value = value;

            this.p = p;

        }

    }

    Node search(Node t, int key) {

        if (t == null || t.key == key)

            return t;

        if (key < t.key)

            return search(t.l, key);

        else

            return search(t.r, key);

    }

    public Node search(int key) {

        return search(root, key);

    }

    Node insert(Node t, Node p, int key, int value) {

        if (t == null) {

            t = new Node(key, value, p);

        } else {

            if (key < t.key)

                t.l = insert(t.l, t, key, value);

            else

                t.r = insert(t.r, t, key, value);

        }

        return t;

    }

    public void insert(int key, int value) {

        root = insert(root, null, key, value);

    }

    void replace(Node a, Node b) {

        if (a.p == null)

            root = b;

        else if (a == a.p.l)

            a.p.l = b;

        else

            a.p.r = b;

        if (b != null)

            b.p = a.p;

    }

    void remove(Node t, int key) {

        if (t == null)

            return;

        if (key < t.key)

            remove(t.l, key);

        else if (key > t.key)

            remove(t.r, key);

        else if (t.l != null && t.r != null) {

            Node m = t.r;

            while (m.l != null)

                m = m.l;

            t.key = m.key;

            t.value = m.value;

            replace(m, m.r);

        } else if (t.l != null) {

            replace(t, t.l);

        } else if (t.r != null) {

            replace(t, t.r);

        } else {

            replace(t, null);

        }

    }

    public void remove(int key) {

        remove(root, key);

    }

    void print(Node t) {

        if (t != null) {

            print(t.l);

            System.out.print(t.key + ":" + t.value + " ");

            print(t.r);

        }

    }

    public void print() {

        print(root);

        System.out.println();

    }

    // Usage example

    public static void main(String[] args) {

        BinarySearchTree tree = new BinarySearchTree();

        tree.insert(3, 1);

        tree.insert(2, 2);

        tree.insert(4, 5);

        tree.print();

        tree.remove(2);

        tree.remove(3);

        tree.print();

        tree.remove(4);

    }

}

## 3. Язык Python :

### 1) Бинарное дерево

class Tree:

def \_\_init\_\_(self, left, right):

self.left = left

self.right = right

### 2)Применение

import uuid

def sanitize\_id(id):

return id.strip().replace(" ", "")

(\_ADD, \_DELETE, \_INSERT) = range(3)

(\_ROOT, \_DEPTH, \_WIDTH) = range(3)

class Node:

def \_\_init\_\_(self, name, identifier=None, expanded=True):

self.\_\_identifier = (str(uuid.uuid1()) if identifier is None else

sanitize\_id(str(identifier)))

self.name = name

self.expanded = expanded

self.\_\_bpointer = None

self.\_\_fpointer = []

@property

def identifier(self):

return self.\_\_identifier

@property

def bpointer(self):

return self.\_\_bpointer

@bpointer.setter

def bpointer(self, value):

if value is not None:

self.\_\_bpointer = sanitize\_id(value)

@property

def fpointer(self):

return self.\_\_fpointer

def update\_fpointer(self, identifier, mode=\_ADD):

if mode is \_ADD:

self.\_\_fpointer.append(sanitize\_id(identifier))

elif mode is \_DELETE:

self.\_\_fpointer.remove(sanitize\_id(identifier))

elif mode is \_INSERT:

self.\_\_fpointer = [sanitize\_id(identifier)]

class Tree:

def \_\_init\_\_(self):

self.nodes = []

def get\_index(self, position):

for index, node in enumerate(self.nodes):

if node.identifier == position:

break

return index

def create\_node(self, name, identifier=None, parent=None):

node = Node(name, identifier)

self.nodes.append(node)

self.\_\_update\_fpointer(parent, node.identifier, \_ADD)

node.bpointer = parent

return node

def show(self, position, level=\_ROOT):

queue = self[position].fpointer

if level == \_ROOT:

print("{0} [{1}]".format(self[position].name, self[position].identifier))

else:

print("\t"\*level, "{0} [{1}]".format(self[position].name, self[position].identifier))

if self[position].expanded:

level += 1

for element in queue:

self.show(element, level) # recursive call

def expand\_tree(self, position, mode=\_DEPTH):

# Python generator. Loosly based on an algorithm from 'Essential LISP' by

# John R. Anderson, Albert T. Corbett, and Brian J. Reiser, page 239-241

yield position

queue = self[position].fpointer

while queue:

yield queue[0]

expansion = self[queue[0]].fpointer

if mode is \_DEPTH:

queue = expansion + queue[1:] # depth-first

elif mode is \_WIDTH:

queue = queue[1:] + expansion # width-first

def is\_branch(self, position):

return self[position].fpointer

def \_\_update\_fpointer(self, position, identifier, mode):

if position is None:

return

else:

self[position].update\_fpointer(identifier, mode)

def \_\_update\_bpointer(self, position, identifier):

self[position].bpointer = identifier

def \_\_getitem\_\_(self, key):

return self.nodes[self.get\_index(key)]

def \_\_setitem\_\_(self, key, item):

self.nodes[self.get\_index(key)] = item

def \_\_len\_\_(self):

return len(self.nodes)

def \_\_contains\_\_(self, identifier):

return [node.identifier for node in self.nodes if node.identifier is identifier]

if \_\_name\_\_ == "\_\_main\_\_":

tree = Tree()

tree.create\_node("Harry", "harry") # root node

tree.create\_node("Jane", "jane", parent = "harry")

tree.create\_node("Bill", "bill", parent = "harry")

tree.create\_node("Joe", "joe", parent = "jane")

tree.create\_node("Diane", "diane", parent = "jane")

tree.create\_node("George", "george", parent = "diane")

tree.create\_node("Mary", "mary", parent = "diane")

tree.create\_node("Jill", "jill", parent = "george")

tree.create\_node("Carol", "carol", parent = "jill")

tree.create\_node("Grace", "grace", parent = "bill")

tree.create\_node("Mark", "mark", parent = "jane")

print("="\*80)

tree.show("harry")

print("="\*80)

for node in tree.expand\_tree("harry", mode=\_WIDTH):

print(node)

print("="\*80)

# Тип данных: Отображение

## 1. Язык C++ :

### 1)Map

template<

    class Key,  
    class T,  
    class Compare = [std::less](http://ru.cppreference.com/w/cpp/utility/functional/less)<Key>,  
    class Allocator = [std::allocator](http://ru.cppreference.com/w/cpp/memory/allocator)<[std::pair](http://ru.cppreference.com/w/cpp/utility/pair)<const Key, T> >

> class map;

### 2)Применение

#include <iostream>

#include <string>

#include <map>

#include <fstream>

using namespace std;

int main()

{

map <string,int> words;

ifstream in;

in.open("in.txt");

string word;

while (in>>word)

{

words[word]++;

}

ofstream out;

out.open("out.txt");

int count = 0;

map <string,int>::iterator cur;

out<<"Words count:"<<endl;

for (cur=words.begin();cur!=words.end();cur++)

{

out<<(\*cur).first<<": "<<(\*cur).second<<endl;count+=(\*cur).second;

}

out<<"Words percenc:"<<endl;

for (cur=words.begin();cur!=words.end();cur++)

{

out<<(\*cur).first<<": "<<(float)((float)(\*cur).second/(float)count)\*100<<"%"<<endl;

}

return 0;

}

## 1. Язык Java :

### 1) map

public class MapExamples {

    private [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) name;

    private double sum;

    public MapExamples([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) name, double sum) {

        this.name = name;

        this.sum = sum;

    }

@Override

    public int hashCode() {

        return 1;

    }

### 2) Применение

public class MapExamples {

    private [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) name;

    private double sum;

    public MapExamples([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) name, double sum) {

        this.name = name;

        this.sum = sum;

    }

    @Override

    public int hashCode() {

        return 1;

    }

    public static void main([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky)[] args) {

        MapExamples example1 = new MapExamples("Some name", 34);

        MapExamples example2 = new MapExamples("Another name", 12.5);

[System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).out.println(example1.hashCode());

[System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).out.println(example2.hashCode());

    }

}

## 1. Язык Python :

### 1)Словарь

typedef struct {

Py\_ssize\_t me\_hash;

PyObject \*me\_key;

PyObject \*me\_value;

} PyDictEntry;

### 2)Применение

>>> d1 = {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5} >>> d2 = {'three': 3, 'two': 2, 'five': 5, 'four': 4, 'one': 1} >>> d1 == d2 True

>>> d1.keys() ['four', 'three', 'five', 'two', 'one']

>>> d2.keys() ['four', 'one', 'five', 'three', 'two']