МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Алгоритмы и структуры данных»
Тема: Иерархические списки

Студент гр. 7383 _____ Александров Р.А.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург
2018

Цель работы.

Познакомиться с иерархическими списками и использованием их в практических задачах на языке программирования C++.

Постановка задачи.

Бинарное коромысло устроено так, что у него есть два плеча: левое и правое. Каждое плечо представляет собой (невесомый) стержень определенной длины, с которого свисает либо гирька, либо еще одно бинарное коромысло, устроенное таким же образом.

Вариант 1. Подсчитать общий вес заданного бинарного коромысла, то есть суммарный вес его гирек.

Реализация задачи.

Для решения поставленной задачи в работе были использованы 3 класса: Main, BinKor, Action.

В классе Main определяются функции для считывания данных:

- void consoleRead() - из консоли;
- void fileRead() - из файла.

Пользователю предлагается либо ввести данные вручную, либо указать текстовый файл, в котором они находятся.

В классе BinKor определяются иерархический список для хранения бинарного коромысла и функции взаимодействия с ним:

- int levelBk(std::stringstream &sstream) постепенно заполняет список;
- int getWeight() возвращает суммарный вес гирек бинарного коромысла.

В классе Action находятся 2 функции:

- void start(std::string str) принимает на вход строку, вызывает подсчет общего веса коромысла;

- std::string getResultString(std::string str) принимает исходную строку, создает промежуточную, через валидацию исходной заполняет промежуточную и возвращает ее в функцию void start(std::string str) .

Тестирование программы.

Программа собрана и проверена в операционных системах Xubuntu 18.04 с использованием компилятора g++ и Windows с использованием MinGW. В других ОС и компиляторах тестирование не проводилось. Тесты находятся в приложении А.

Вывод.

В ходе лабораторной работы были получены основные навыки программирования иерархических списков на языке C++, изучены приемы хранения бинарного коромысла в иерархическом списке. Результатом стала программа, рассчитывающая суммарный вес гирек в коромысле.

ПРИЛОЖЕНИЕ А
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Таблица 1 – Тестирование программы

Input	Output
((8((2 6)(4 3)))(1((5 5)(1 2))))	16
((2((4 8)(7 1)))(1 3))	12
((6((4 4)(8 8)))(9 6))	18
((-6((4 4)(8 8)))(9 -6))	Unexpected char [((-6((4 4)(8 8)))(9 -6))]
(()))3((2 4)(7 8)))(5 2))	Error. Wrong string format

ПРИЛОЖЕНИЕ Б
КОД ПРОГРАММЫ

main.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include "main.h"
#include "action.h"
using namespace std;

void Main::fileRead() {
    string fileName, str;
    cout << "What`s the file name?" << endl;
    cin >> fileName;
    cout << "--------------" << endl;
    cout << "Reading from " << fileName << endl;
    cout << "--------------" << endl;

    ifstream inFile;
    inFile.open(fileName);
    if (!inFile) {
        cout << "Cannot find this file" << endl;
        cout << endl;
        return;
    }
    while (!inFile.eof()) {
        getline(inFile, str);
        action.start(str);
    }
    inFile.close();
}

void Main::consoleRead() {
    string str;
    cout << "Enter string:" << endl;
    getline(cin, str); // remove '\n'
    getline(cin, str);
    action.start(str);
}

void Main::menu() {
    cout << "1. Enter numbers from the txt" << endl;
    cout << "2. Enter numbers from the console" << endl;
    cout << "0. Exit" << endl;
}

int main() {
    cout << "Hello! This program calculates the weight of the binary
rocket" << endl;
```

```
    Main main;

    while (true) {
        main.menu();
        cin >> main.choice;
        switch (main.choice) {
            case 1:
                main.fileRead();
                break;
            case 2:
                main.consoleRead();
                break;
            case 0:
                exit(1);
        }
    }
}
```

main.h

```
#pragma once

#include "action.h"

class Main {
private:
    Action action;
public:
    unsigned int choice;

    void consoleRead();

    void fileRead();

    void menu();
};
```

binkor.cpp

```
#include <iostream>
#include "binkor.h"

// static count for sum weight
int BinKor::countWeight = 0;

BinKor::BinKor(stringstream &sstream, int pos) {
    char ch;
    switch (pos) {
        case 0:
            flag = true;
```

```
            content.elem.deep = nullptr;
            content.elem.weight = 0;
            if (sstream.peek() == '(')
                sstream >> ch;
            levelBk(sstream);
            sstream >> ch;
            break;
        case 1:
            flag = false;
            content.atom.atomicEl = new BinKor(sstream, 0);
            content.atom.next = new BinKor(sstream, 2);
            break;
        case 2:
            flag = false;
            content.atom.atomicEl = new BinKor(sstream, 0);
            content.atom.next = nullptr;
            break;
    }
}

int BinKor::levelBk(stringstream &sstream) {
    char ch;
    if (sstream.peek() == '(')
        sstream >> ch;
    sstream >> content.elem.level;
    if (sstream.peek() == '(') {
        content.elem.deep = new BinKor(sstream);
    } else {
        sstream >> content.elem.weight;
    }
    sstream >> ch;
}

int BinKor::getWeight() {
    if (!flag) {
        if (content.atom.next) {
            (content.atom.next)->getWeight();
            (content.atom.atomicEl)->getWeight();
        } else
            (content.atom.atomicEl)->getWeight();
    } else {
        countWeight += content.elem.weight;
        if (content.elem.deep) {
            (content.elem.deep)->getWeight();
```

```
        } else {
            return (countWeight);
        }
    }
}

BinKor::~BinKor() {
    if (!flag) {
        delete content.atom.atomicEl;
        if (content.atom.next)
            delete content.atom.next;
    } else if (content.elem.deep)
        delete content.elem.deep;

    countWeight = 0;
}
```

binkor.h

```
#pragma once

#include <sstream>

using namespace std;

class BinKor {
    struct Atom {
        BinKor *atomicEl;
        BinKor *next;
    };

    struct Elem {
        int level;
        int weight;
        BinKor *deep;
    };

    union Content {
        Atom atom;
        Elem elem;
    };
private:
    bool flag; //true: element, false: atom
    union Content content;
    static int countWeight;

public:
    explicit BinKor(stringstream &sstream, int pos = 1);
```

```
    int levelBk(stringstream &sstream);

    int getWeight();

    ~BinKor();
};
```

action.cpp

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cctype>
#include <string>
#include "binkor.h"
#include "main.h"
#include "action.h"

using namespace std;

void Action::start(std::string str) {
    stringstream sstream;
    string resultStr;

    if (str.empty()) {
        cout << "The entering string is empty!" << endl;
        return;
    }
    resultStr = getResultString(str);
    if (resultStr.empty())
        return;

    sstream << resultStr;
    BinKor list(sstream);

    int answer = list.getWeight();
    cout << "Full weight of the binary rocket = " << answer << endl;
```

```
    cout << endl;
}

string Action::getResultString(string str) {
    string midStr;
    for (char c : str) {
        if (c == '(') {
            if (isspace(midStr.back()))
                midStr.erase(midStr.size() - 1, 1);
            midStr.push_back(c);
        } else if (c == ' ') {
            if (isdigit(midStr.back()))
                midStr.push_back(c);
        } else if (c == ')') {
            if (isspace(midStr.back()))
                midStr.erase(midStr.size() - 1, 1);
            midStr.push_back(c);
        } else if (isdigit(c)) {
            midStr.push_back(c);
        } else {
            cout << "Unexpected char [ " << str << " ]" << endl;
            cout << endl;
            return "";
        }
    }
    for (int i = 0; i < midStr.size(); i++) {
        switch (midStr[i]) {
            case '(':
                i++;
                if (midStr[i] == '(') {
                    i++;
                    if (isdigit(midStr[i]))
                        continue;
                }
```

```
                cout << "Error. Wrong string format" << endl;
                return "";
            case ')':
                i++;
                if (midStr[i] == '(') {
                    i++;
                    if (isdigit(midStr[i]))
                        continue;
                } else if (str[i] == ')')
                    continue;
                cout << "Error. Wrong string format" << endl;
                return "";
        }
        return midStr;
    }
}
```

action.h

```
#pragma once

#include <string>

class Action {
public:
    void start(std::string str);

    std::string getResultString(std::string str);
};
```

Makefile

```
CXX=g++
RM=rm -f
LDFLAGS=-g -Wall

SRCS=main.cpp action.cpp binkor.cpp
OBJS=$(subst .cpp,.o,$(SRCS))

all: main

main: $(OBJS)
     $(CXX) $(LDFLAGS) -o main $(OBJS)
```

```
main.o: main.cpp main.h

action.o: action.cpp action.h

binkor.o: binkor.cpp binkor.h

clean:
	$(RM) $(OBJS)

distclean: clean
	$(RM) main
```