МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

|  |  |  |
| --- | --- | --- |
| ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: |  |  |

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е. В. Павлов |
| (должность, уч. степень, звание) |  | (подпись, дата) |  | (инициалы, фамилия) |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

«ОПРЕДЕЛЕНИЕ СЛОЖНОСТИ ПРОГРАММНОГО КОДА

НА ОСНОВЕ ЛЕКСИЧЕСКОГО АНАЛИЗА»

ПО КУРСУ: «МЕТРОЛОГИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

|  |  |  |  |
| --- | --- | --- | --- |
| РАБОТУ ВЫПОЛНИЛ (-А) СТУДЕНТ (-КА): | 4931 | / | Е. Ю. Ильченко |
|  | (номер группы) |  | (инициалы, фамилия) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | / |  | / | 28.04.2021 |
|  |  | (подпись студента) |  | (дата отчета) |

1. Цель работы

Целью данной работы является оценка сложности программного кода на основе лексического анализа с использованием метрик Холстеда и Джилба.

1. Задание на лабораторную работу

Составить словарь операторов и операндов программного кода приложения (или его фрагмента) и произвести расчёт метрик Холстеда и Джилба.

Выполнить анализ полученных результатов.

Разрешается ограничить исходный код программы (из ЛР 1) и использовать для расчётов метрик Холстеда и Джилба фрагмент кода (или модуль) размером 120-160 строк (из которых не более 15% пустых), начиная с функции main.

|  |
| --- |
| Вариант задания:  6. Приложение для поиска туров (по стране) |

1. Словарь операторов и операндов

Анализируемый программный код представлен в *Приложении А*.

Таблица 1 — Словарь операторов

|  |  |  |  |
| --- | --- | --- | --- |
| N | Операторы | Номера строк | Количество повторений |
| 1 | . | 26; 31; 39; 43; 45; 52; 56; 57; 58; 67; 70; 78; 82; 86; 106; 110; 111; 118; 119; 121; 123; 125; 127; 129; 130; 132; 133; 136; 140; 141; 144; 145; 161; 165; 166; 171; 172; 186; 190; 193; 194 | 41 |
| 2 | () | 25; 26; 31; 39; 43; 45; 52; 56; 58; 67; 70; 82; 106; 110; 111; 118; 121; 123; 125; 127; 129; 132; 133; 136; 144; 145; 161; 171 ;172; 186; 190; 193; 194 | 33 |
| 3 | == | 72; 167; 174; 191 | 4 |
| 4 | = | 22, 22, 22; 31; 39; 56; 64; 70; 82; 93; 106; 118; 121; 123; 125; 127; 129; 141; 166; 183; 190 | 21 |
| 5 | , | 17; 21, 21, 21; 22, 22, 22, 22, 22; 78; 86; 130, 130, 130, 130, 130, 130; 140; 165 | 19 |
| 6 | case, default, switch | 33; 35; 41; 42; 50; 51; 65; 66; 94; 95; 96; 97; 98; 101; 102; 109; 108; 116; 117; 134; 135; 159; 160; 184; 185; 203; 204; 205; 206; 207; 210; 211; 212; 213; 214 | 35 |
| 8 | do, while | 27; 36; 100; 103; 209; 216 | 6 |
| 9 | if -else | 43; 47; 52; 60; 67; 72; 80; 89; 110; 113; 136; 142; 149; 155; 161; 167; 174; 179; 186; 191; 199 | 21 |
| 10 | using namespace | 10 | 1 |
| 11 |  | 15 | 1 |
| 12 | system | 18 | 1 |
| 13 | {} | 14 (217); 28 (216); 34 (215); 36 (100); 41 (99); 43 (46); 47 (49); 53 (59); 61 (63); 68 (88); 72 (79); 80 (87); 90 (92); 103 (209); 108 (208); 110 (112); 113 (115); 137 (154); 143(148); 150 (153); 156 (158); 162 (178); 168 (173); 175 (177); 180 (182); 187 (198); 192 (197); 200 (202) | 56 |
| 14 | << | 29; 30; 37; 38; 44; 48; 54; 55; 62; 69; 73; 76; 77; 81 84; 85; 91; 104; 105; 114; 120; 122; 124; 126; 128; 131; 138; 147; 152; 157; 163; 170; 176; 181; 188; 189;196; 201 | 38 |
| 15 | >> | 74; 139; 164 | 3 |
| 16 | string | 21 | 1 |
| 17 | unsigned short int | 22 | 1 |
| 18 | bool | 24 | 1 |
| 19 | int | 13 | 1 |
| 20 | void | 11 | 1 |
| 21 | Dlist | 19 | 1 |
| 22 | Check | check | 1 |
| *Итого:* | | | 287 |

Таблица 2 — Словарь операндов

|  |  |  |  |
| --- | --- | --- | --- |
|  | Операнды | Номера строк | Количество повторений |
| 1 | std | 10 | 1 |
| 2 | savetimeanddate | 11; 24 | 2 |
| 3 | main | 13 | 1 |
| 4 | 1251 | 15; 16 | 2 |
| 5 | LC\_ALL | 17 | 1 |
| 6 | "Russian" | 17 | 1 |
| 7 | "color F1 " | 18 | 1 |
| 8 | list | 19; 25; 42; 44; 51; 56; 57; 66; 77; 85; 109; 110; 117; 129; 131; 132; 135; 139; 140; 143; 144; 160; 164; 165; 170; 171; 185; 192; 193 | 29 |
| 9 | check | 20; 30; 38; 55; 69; 81; 105; 118; 120; 122; 124; 126; 128; 129 | 14 |
| 10 | resort | 21; 81; 85; 120; 129 | 5 |
| 11 | apart | 21; 222; 129 | 3 |
| 12 | food | 21; 126; 129 | 3 |
| 13 | code | 21; 73; 77; 117; 129; 138; 139; 140; 163; 164; 165 | 11 |
| 14 | h | 22; 124; 129; | 3 |
| 15 | price | 22; 128; 129 | 3 |
| 16 | x | 22; 30; 32; 215 | 4 |
| 17 | 0 | 22; 63; 92; 94; 182; 203; 210 | 7 |
| 18 | z | 22; 38; 40; 99 | 4 |
| 19 | y | 22; 105; 107; 208 | 4 |
| 20 | l | 22; 55; 56; 63; 69; 71; 77; 85; 92; 165; 166; 173; 182; 189; 190 | 15 |
| 21 | result | 23; 140; 141 | 3 |
| 22 | load | 25 | 1 |
| 23 | "\n\tГлавное меню:\n" | 28 | 1 |
| 24 | "1. Поиск тура;\n2. Панель мастера обновления ПО;\n0. Выход из программы;" | 29 | 1 |
| 25 | checkc | 30; 38; 105 | 3 |
| 26 | "cls" | 31; 39; 70; 74; 82; 106; 145; 150; 168; 194 | 10 |
| 27 | 1 | 34; 41; 71; 108; 139; 164; 166; 190 | 8 |
| 28 | "\n\tМеню:\n" | 36; 103 | 2 |
| 29 | "1. Вывести список всех предложений;\n2. Сортировать туры;\n3. Найти тур;\n0. Перейти в главное меню;" | 37 | 1 |
| 30 | free | 42; 51; 66; 109; 135; 160; 185 | 7 |
| 31 | "Купить тур можно в нашем агенстве по адресу г. Санкт-Петербург ул. Цветочная д. 2 \nПо всем вопросам звоните на горячую линию: 8(800)555-35-35;\n" | 43 | 1 |
| 32 | show | 44; 110 |  |
| 33 | "На данный момент туров по стране нет\n" | 47 | 1 |
| 34 | 2 | 50; 101; 116; 173 | 4 |
| 35 | "Что вы хотите сделать?\n\n" | 53 | 1 |
| 36 | "1. Сортировать по возрастанию цены;\n2. Сортировать по убыванию цены;\n0. Перейти в меню;" | 54 | 1 |
| 37 | checkf | 55; 68; 189 | 3 |
| 38 | sorting | 56 | 1 |
| 39 | save | 57; 132; 144; 171 | 4 |
| 40 | "Список предложений пуст!\n\n" | 61; 90; 180; 200 | 4 |
| 41 | 3 | 65; 134 | 2 |
| 42 | "\n1. Найти предложение по индивидуальному коду;\n" << "2. Найти элемент по названию курорта;\n" | 68 | 1 |
| 43 | "Введите индивидуальный код предложения:\n" | 72; 137; 162 | 3 |
| 44 | "\nКупить тур можно в нашем агенстве по адресу г. Санкт-Петербург ул. Цветочная д. 2 \nПо всем вопросам звоните на горячую линию: 8(800)555-35-35;\n\n" | 75; 83 | 2 |
| 45 | "\nПо вашему запросу найдено:\n" | 76; 84 | 2 |
| 46 | findelem | 77; 85; 139; 164 | 4 |
| 47 | "Введите название курорта:\n" | 80 | 1 |
| 48 | checkname | 81; 120; 122; 126 | 4 |
| 49 | "1. Вывести список всех предложений;\n2. Добавить предложение;\n3. Изменить предложение;\n4. Удалить предложение;\n5. Удалить все предложения;\n0. Перейти в главное меню;" | 104 | 1 |
| 50 | "Список предложений пуст!Заполните его.\n" | 113 | 1 |
| 51 | forcode | 117 | 1 |
| 52 | fillandcheckdate | 118 | 1 |
| 53 | "Введите название курорта: " | 119 | 1 |
| 54 | "Введите тип и название апартаментов: " | 121 | 1 |
| 55 | "Введите кол-во человек: " | 123 | 1 |
| 56 | chec | 124 | 1 |
| 57 | "Введите информацию о питании: " | 125 | 1 |
| 58 | "Введите стоимость: " | 127 | 1 |
| 59 | checkprice | 128 | 1 |
| 60 | add | 129 | 1 |
| 61 | "Информация успешно добавлена.\n\n" | 130 | 1 |
| 62 | del | 131; 143; 170; 193 | 4 |
| 63 | editing | 140 | 1 |
| 64 | "\nПредложение успешно отредактировано!\n\n" | 146 | 1 |
| 65 | "\nТакого предложения нет!\n" | 151 | 1 |
| 66 | "Список предложений пуст!Заполните его.\n\n" | 156 | 1 |
| 67 | remove | 165 | 1 |
| 68 | "\nПредложение успешно удалёно!\n\n" | 169 | 1 |
| 69 | "\nТакого предложения нет!\n\n" | 175 | 1 |
| 70 | "Вы точно хотите удалить все предложения?\nПосле удаления все текущие данные будут потеряны!\n\n" | 187 | 1 |
| 71 | clear | 192 | 1 |
| 72 | "Данные удалены\n\n" | 195 | 1 |
| 73 | cout | 28; 29; 36; 37; 43; 47; 53; 54; 61; 68; 72; 75; 76; 80; 83; 84; 90; 103; 104; 113; 119; 121; 123; 125; 127; 130; 137; 146; 151; 156; 162; 169; 175; 180; 187; 188; 195; 200 | 38 |
| 74 | cin | 73; 138; 163 | 3 |
| 75 | SetConsoleCP | 15 | 1 |
| 76 | SetConsoleOutputCP | 16 | 1 |
| *Итого:* | | | 262 |

1. Оценка сложности программного кода

Таблица 3 — Значения метрик Холстеда

|  |  |  |
| --- | --- | --- |
| Наименование характеристики | Обозначение и формула  для вычисления | Значение |
| Число простых (уникальных) операторов |  | 22 |
| Число простых (уникальных) операндов |  | 76 |
| Общее число всех операторов |  | 287 |
| Общее число всех операндов |  | 262 |
| Словарь программы |  | 98 |
| Длина реализации программы |  | 549 |
| Расчётная длина программы |  | 573 |
| Разница между *N* и *Ñ* (в процентах) |  | 4,4 % |
| Объём программы (в битах) |  | 3631 |
| Сложность программы |  | 38 |
| Усилие |  | 137997 |
| Время реализации (в секундах) |  | 7667 |
| Время реализации (в минутах) | — | 128 |
| Время реализации (в часах) | — | 2,2 |

Таблица 4 — Значения метрики Джилба

|  |  |  |
| --- | --- | --- |
| Наименование характеристики | Обозначение и формула  для вычисления | Значение |
| Абсолютная сложность программы |  | 27 |
| Общее количество операторов в программе |  | 287 |
| Относительная сложность программы |  | 0,094077 |

Выводы по работе

В результате выполнения данной работы произведены расчёты показателей сложности для программного кода (см. Приложении А) с использованием метрик Холстеда и Джилба. Согласно заданию расчёт указанных метрических характеристик необходимо выполнить ручным методом, для этих целей был составлен словарь операторов и операндов анализируемого программного кода.

При анализе полученных значений метрик можно выделить следующее:

* Значение длины реализации больше значения расчетной длины на 4,4%, что может свидетельствовать об удачных приёмах кодирования, так как согласно метрике Холстеда, разница между двумя указанными характеристиками должна быть в пределах 10%.
* Время реализации *T* (около 3 часов) близко к реальному времени, которое было затрачено на кодирование анализируемого фрагмента кода;
* Необходимо отметить, что показатель сложности программы следует рассматриваться вкупе с аналогичными расчетами по другим проектам или в рамках текущего проекта для идентичных задач. В контексте настоящей работы данный показатель сложно интерпретировать с точки зрения оценки трудоёмкости кодирования или сложности проверки кода;
* Анализируемый программный код обладает невысокой относительной сложностью (*cl* = 0,094077), так как на 287 операторов приходится 6 циклические конструкции и 21 оператора условий, что позволяет говорить о низкой логической сложности данного программного кода.

Таким образом, можно заключить, что выполненная работа соответствует поставленной задаче и отвечает всем сформулированным в методических указаниях требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Павлов Е. В. Методические рекомендации к выполнению лабораторных работ: Метрология программного обеспечения / Евгений Васильевич Павлов. —   
   СПб ГУАП, 2021
2. Черников, Б. В. Управление качеством программного обеспечения: учебник /   
   Б. В. Черников. — М.: ИД «ФОРУМ»: ИНФРА-М, 2012. — 240 с.: ил.
3. Широков, А. И. Стандартизация, сертификация и оценка качества программного обеспечения: учебное пособие / А. И. Широков, Е. П. Потоцкий. —   
   М.: ИД «МИСиС», 2013. — 208 с.
4. Справочное руководство по C++ (машинный перевод) [Электронный ресурс]: Microsoft Docs. — Microsoft, 2021. — URL: <https://docs.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=vs-2017> (дата обращения: 21.03.2021).
5. Halstead Metrics [Электронный ресурс]: Static Metrics for C, C++, Ada and Java. — IBM Knowledge Center, 2021. — URL: <https://www.ibm.com/support/knowledgecenter/SSSHUF_8.0.0/com.ibm.rational.testrt.studio.doc/topics/csmhalstead.htm> (дата обращения: 21.03.2021).

ПРИЛОЖЕНИЕ А

Анализируемый программный код

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219 | #include <iostream>  #include <windows.h>  #include "Check.h"  #include "Class.h"  #include <string>  #include <fstream>  #include <iostream>  #include <time.h>  #include <cstdlib>  using namespace std;  void savetimeanddate();  int main()  {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  setlocale(LC\_ALL, "Russian");  system("color F1 ");  Dlist list;  Check check;  string resort, apart, food, code;  unsigned short int h, price, x = 0, z = 0, y = 0, l;  bool result;  savetimeanddate();  list.load();  do  {  cout << "\n\tГлавное меню:\n";  cout << "1. Поиск тура;\n2. Панель мастера обновления ПО;\n0. Выход из программы;";  x = check.checkc();  system("cls");  switch (x)  {  case 1:  do {  cout << "\n\tМеню:\n";  cout << "1. Вывести список всех предложений;\n2. Сортировать туры;\n3. Найти тур;\n0. Перейти в главное меню;";  z = check.checkc();  system("cls");  switch (z) {  case 1:  if (list.free()) {  cout << "Купить тур можно в нашем агенстве по адресу г. Санкт-Петербург ул. Цветочная д. 2 \nПо всем вопросам звоните на горячую линию: 8(800)555-35-35;\n";  list.show();  }  else {  cout << "На данный момент туров по стране нет\n";  }  break;  case 2:  if (list.free())  {  cout << "Что вы хотите сделать?\n\n";  cout << "1. Сортировать по возрастанию цены;\n2. Сортировать по убыванию цены;\n0. Перейти в меню;";  l = check.checkf();  list.sorting(l);  list.save();  }  else  {  cout << "Список предложений пуст!\n\n";  }  l = 0;  break;  case 3:  if (list.free())  {  cout << "\n1. Найти предложение по индивидуальному коду;\n" << "2. Найти элемент по названию курорта;\n";  l = check.checkf();  system("cls");  if (l == 1) {  cout << "Введите индивидуальный код предложения:\n";  cin >> code;  system("cls");  cout << "\nКупить тур можно в нашем агенстве по адресу г. Санкт-Петербург ул. Цветочная д. 2 \nПо всем вопросам звоните на горячую линию: 8(800)555-35-35;\n\n";  cout << "\nПо вашему запросу найдено:\n";  list.findelem(l, code);  }  else {  cout << "Введите название курорта:\n";  resort = check.checkname();  system("cls");  cout << "\nКупить тур можно в нашем агенстве по адресу г. Санкт-Петербург ул. Цветочная д. 2 \nПо всем вопросам звоните на горячую линию: 8(800)555-35-35;\n\n";  cout << "\nПо вашему запросу найдено:\n";  list.findelem(l, resort);  }  }  else  {  cout << "Список предложений пуст!\n\n";  }  l = 0;  break;  case 0:  break;  default:  break;  }  } while (z);  break;  case 2:  do {  cout << "\n\tМеню:\n";  cout << "1. Вывести список всех предложений;\n2. Добавить предложение;\n3. Изменить предложение;\n4. Удалить предложение;\n5. Удалить все предложения;\n0. Перейти в главное меню;";  y = check.checkc();  system("cls");  switch (y) {  case 1:  if (list.free()) {  list.show();  }  else {  cout << "Список предложений пуст!Заполните его.\n";  }  break;  case 2:  code = list.forcode();  check.fillandcheckdate(date);  cout << "Введите название курорта: ";  resort = check.checkname();  cout << "Введите тип и название апартаментов: ";  apart = check.checkname();  cout << "Введите кол-во человек: ";  h = check.chec();  cout << "Введите информацию о питании: ";  food = check.checkname();  cout << "Введите стоимость: ";  price = check.checkprice();  list.add(code, date, resort, apart, h, food, price);  cout << "Информация успешно добавлена.\n\n";  list.del();  list.save();  break;  case 3:  if (list.free())  {  cout << "Введите индивидуальный код предложения:\n";  cin >> code;  list.findelem(1, code);  result = list.editing(code);  if (result)  {  list.del();  list.save();  system("cls");  cout << "\nПредложение успешно отредактировано!\n\n";  }  else  {  system("cls");  cout << "\nТакого предложения нет!\n";  }  }  else  {  cout << "Список предложений пуст!Заполните его.\n\n";  }  break;  case 4:  if (list.free())  {  cout << "Введите индивидуальный код предложения:\n";  cin >> code;  list.findelem(1, code);  l = list.remove(code);  if (l == 1)  {  system("cls");  cout << "\nПредложение успешно удалёно!\n\n";  list.del();  list.save();  }  else if (l == 2)  {  cout << "\nТакого предложения нет!\n\n";  }  }  else  {  cout << "Список предложений пуст!\n\n";  }  l = 0;  break;  case 5:  if (list.free())  {  cout << "Вы точно хотите удалить все предложения?\nПосле удаления все текущие данные будут потеряны!\n\n";  cout << "1. Да;\n2. Нет;\n";  l = check.checkf();  if (l == 1)  {  list.clear();  list.del();  system("cls");  cout << "Данные удалены\n\n";  }  }  else  {  cout << "Список предложений пуст!\n\n";  }  break;  case 0:  break;  default:  break;  }  } while (y);  break;  case 0:  break;  default:  break;  }  } while (x);  } |
|  |  |