-7

Специальность: 09.02.07 Информационные системы и программирование

Выполнил:

Студент группы Y2433

Тертень Д.В.

ОТЧЕТ

о лабораторной работе № 3

по теме: Типы данных определяемые пользователями

по дисциплине: Разработка программных модулей

Санкт-Петербург 2020

Проверил:

Антонов. М.Б.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

УНИВЕРСИТЕТ ИТМО

Факультет среднего профессионального образования

1. Цель и задачи

Цель: Создать программу обрабатывающие действия, описанные в файле.

Задачи:

* научиться работать с файлами;
* научиться относительно содержимого файла менять типы считываемых данных.

1. Задание

Общее ограничение на всю лабораторную работу: можно использовать только методы С++.

Дан файл имеющий следующий вид:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № операции | Операция | Тип данных (int/float/string) | Первый оператор | Второй оператор | Ожидаемый результат |
| 1 | + | int | 31 | 4 | int |
| 2 | - | float | 22,37 | 0,15 | float |
| 3 | \* | int | 0 | 2 | int |
| 4 | / | float | 0,69 | 29,8 | float |
| 5 | pow | int | 5 | 3 | int |

Требуется считать все операции из файла и вывести их результат в отдельный файл:

|  |  |  |
| --- | --- | --- |
| № операции | Результат | Тип |
| 1 | 35 | int |
| 2 | 22,22 | float |

Т.к. вывод в консоль не предусмотрен, в качестве результатов выполнения предоставлять: выходной файл.

1. КОд программы

Файл main.cpp:

/\*\*

\* @file main.cpp

\* @author Тертень Даниил

\* @brief Лабораторная работа №3

\*/

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

using namespace std;

/\*\*

\* @brief структура для хранения операций

\*/

struct Operation

{

int number = 0;

string action;

string type;

string first;

string second;

string result\_type;

};

/\*\*

\* @brief получение операции из файла

\* @param operation стрктура, в которую сохраняется операция

\* @param line входной файл

\*/

void getOperation(Operation &operation, ifstream &line)

{

line >> operation.number;

line >> operation.action;

line >> operation.type;

line >> operation.first;

line >> operation.second;

line >> operation.result\_type;

}

/\*\*

\* @brief произвести указанную операцию (сложение, вычитание и т.д.)

\* @param first первый операнд

\* @param second второй операнд

\* @param action операция

\* @return результат выполнения операции

\*/

template <typename A>

A doOperation(const A& first, const A& second, const string& action)

{

A result;

if (action == "+") {

result = first + second;

} else if (action == "-") {

result = first - second;

} else if (action == "\*") {

result = first \* second;

} else if (action == "/") {

if (second == 0) {

throw runtime\_error("division by zero");

}

result = first / second;

} else if (action == "pow") {

result = pow(first, second);

} else {

throw runtime\_error("unknown action");

}

return result;

}

/\*\*

\* @brief перегруженная функция doOperation для строк

\* @param first первый операнд

\* @param second второй операнд

\* @param action операция

\* @return результат выполнения операции

\*/

string doOperation(const string& first, const string& second, const string& action)

{

string result;

if (action == "+") {

result = first + second;

} else {

throw runtime\_error("unknown action");

}

return result;

}

/\*\*

\* @brief сохранение результатов операций в файл

\* @param result Результат операции

\* @param operation Операция

\* @param output Выходной файл

\*/

template <typename A>

void saveOperation(A result, Operation &operation, ofstream &output)

{

if (operation.result\_type == "int") {

result = (int) result;

}

output << operation.number << " " << result << " " << operation.result\_type << endl;

}

/\*\*

\* @brief перегруженная функция сохранения результатов операций в файл для строк

\* @param result Результат операции

\* @param operation Операция

\* @param output Выходной файл

\*/

void saveOperation(const string &result, Operation &operation, ofstream &output)

{

output << operation.number << " " << result << " " << operation.result\_type << endl;

}

/\*\*

\* @brief запуск функций для получения и сохранения результата для разных типов данных

\* @param operation операция

\* @param output выходной файл

\*/

void doAndSaveOperation(Operation &operation, ofstream &output)

{

try {

if (operation.type == "int") {

int first = stoi(operation.first);

int second = stoi(operation.second);

int result = doOperation(first, second, operation.action);

saveOperation(result, operation, output);

} else if (operation.type == "float") {

float first = stof(operation.first);

float second = stof(operation.second);

float result = doOperation(first, second, operation.action);

saveOperation(result, operation, output);

} else if (operation.type == "string") {

string result = doOperation(operation.first, operation.second, operation.action);

saveOperation(result, operation, output);

} else {

throw runtime\_error("unknown type");

}

} catch (exception &e) {

string error(e.what());

saveOperation(error, operation, output);

}

}

/\*\*

\* @brief Точка входа в программу, открытие файлов, запуск обработки каждой строки

\* @return 0

\*/

int main()

{

ifstream input;

ofstream output;

input.open("../input.txt", ios::in);

output.open("../output.txt", ios::out);

if (!input.is\_open()) {

cout << "File cannot be opened";

return 0;

}

while (!input.eof()) {

Operation operation;

getOperation(operation, input);

doAndSaveOperation(operation, output);

}

input.close();

output.close();

return 0;

}

В качестве результата работы программы представлен текст из выходного файла:

1 28 int

2 -28.3749 float

3 92 int

4 0 int

5 81 int

6 14 int

7 -11 int

8 209 int

9 2 int

10 2 int

11 6 int

12 -1.40981 float

13 16 int

14 0 int

15 1 int

16 HelloWorld string

17 -2 int

18 27 int

19 2 int

20 5 int

21 10 int

22 -5.42203 float

23 112 int

24 1 int

25 8 int

26 2.06987 float

27 10 int

28 9 int

29 0 int

30 1.42979e+11 float

31 8 int

32 -3.10291 float

33 3 int

34 1.02995 float

35 -2147483648 int

36 1.8468 float

37 16 int

38 61.1252 float

39 0 int

40 4 int

41 51 int

42 -10 int

43 168 int

44 1.44751 float

45 537824 int

46 31 int

47 -1 int

48 47 int

49 0 int

50 1 int

51 17 int

52 9 int

53 16 int

54 151.965 float

55 -2147483648 int

56 6 int

57 6 int

58 16.0057 float

59 0 int

60 0 int

61 34 int

62 16 int

63 544 int

64 0.249709 float

65 100 int

66 6 int

67 -25 int

68 0 int

69 0 int

70 -2.14748e+09 int

71 9 int

72 -3 int

73 80 int

74 0.540433 float

75 3125 int

76 30.6241 float

77 -3 int

78 27 int

79 2 int

80 4.52702e+06 int

81 19 int

82 7.13319 float

83 15 int

84 0.505008 float

85 -2147483648 int

86 17.7185 float

87 -16 int

88 56 int

89 0 int

90 -2.14748e+09 int

91 30 int

92 -17 int

93 3 int

94 0.144889 float

95 8 int

96 8 int

97 2 int

98 29.7416 float

99 0 int

100 0 int

101 30 int

102 -3.79674 float

103 16 int

104 0 int

105 410338673 int

106 39 int

107 2 int

108 212 int

109 0 int

110 5883.71 float

111 2 int

112 -20 int

113 6 int

114 0.646622 float

115 -2147483648 int

116 43.1903 float

117 -17 int

118 0.127587 float

119 1 int

120 4.25935 float

121 23 int

122 -11 int

123 12 int

124 0.011334 float

125 0 int

126 6 int

127 -15 int

128 119 int

129 0 int

130 277.494 float

131 36 int

132 -4 int

133 8 int

134 1 int

135 -2147483648 int

136 16.7562 float

137 -2 int

138 8 int

139 0 int

140 0 int

141 13 int

142 -4.34939 float

143 126 int

144 6 int

145 -2147483648 int

146 11 int

147 11 int

148 108 int

149 0 int

150 7.14297e+34 float

151 18 int

152 8.07855 float

153 2 int

154 0 int

155 -2147483648 int

156 8.27536 float

157 -12 int

158 448 int

159 division by zero int

160 4643.07 float

161 28 int

162 26.4311 float

163 126 int

164 5 int

165 1280000000 int

166 65.2343 float

167 2 int

168 40.0931 float

169 0 int

170 -2.14748e+09 int

171 22 int

172 4.73651 float

173 3 int

174 0.0194596 float

175 6561 int

176 44.0342 float

177 12 int

178 164 int

179 18 int

180 1.36935e+37 float

181 9 int

182 -6 int

183 15 int

184 11.6744 float

185 1089 int

186 8.18034 float

187 -10 int

188 23 int

189 0 int

190 2.72279e+38 float

191 23 int

192 -12 int

193 72 int

194 0.889818 float

195 -2147483648 int

196 18.5794 float

197 -2 int

198 214 int

199 0 int

200 -2.14748e+09 int

201 WeeNeedMoreGold string

202 14 int

203 0 int

204 0 int

205 -2147483648 int

206 27 int

207 -3 int

208 150 int

209 0 int

210 0.140952 float

211 21 int

212 2 int

213 55 int

214 0 int

215 1073741824 int

216 40 int

217 1 int

218 0 int

219 0 int

220 3.68486 float

221 61 int

222 15 int

223 48 int

224 2 int

225 -2147483648 int

226 22 int

227 19 int

228 1 int

229 0 int

230 926970 int

231 24 int

232 -4 int

233 2 int

234 1.89733 float

235 27 int

236 17 int

237 -12 int

238 3.43005 float

239 division by zero int

240 -2.14748e+09 int

241 WeeNeedMoreWood string

242 1 int

243 232 int

244 0.693425 float

245 169 int

246 17 int

247 -13 int

248 126.308 float

249 1 int

250 10634.6 float

1. Диаграммы активности

Диаграммы активности представлены на рисунках 1, 2, 3.



Рисунок 1 – диаграмма функции main

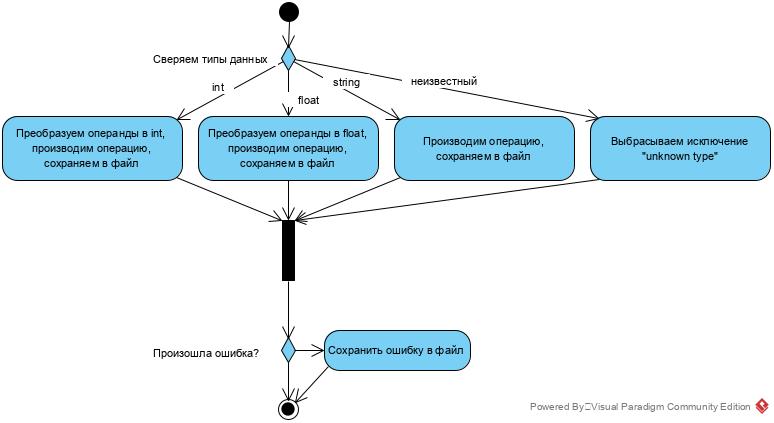


Рисунок 2 – диаграмма функции doAndSaveOperation

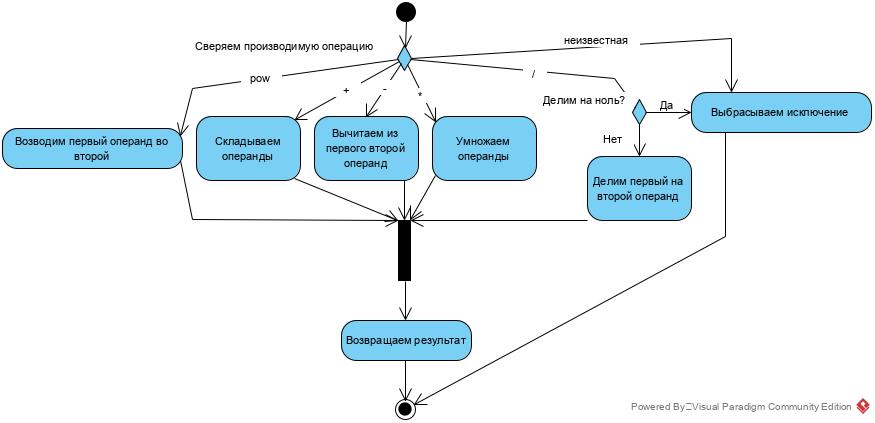


Рисунок 3 – диаграмма функции doOperation

# ВЫВОД

В процессе выполнения лабораторной работы были получены навыки работы с файлами, шаблонами, определением типов данных в зависимости от внешних данных.