

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №12-2  
по курсу: «Языки и методы программирования»  
«Обработка текстовых файлов»

Выполнил:  
Студент группы ИУ9-21Б

Проверил:  
Посевин Д. П.

Москва, 2024

## 1. Цель

Целью лабораторной работы является приобретение навыка разработки на языке C++ программ, осуществляющих анализ и преобразование текстовых файлов, записанных в различных форматах.

## 2. Персональный вариант

Найти все файлы с расширением «svg» в указанном каталоге и заменить прямоугольники с прямыми углами на прямоугольники с закруглёнными углами, и наоборот. Прямоугольник с прямыми углами в SVG-файле задаётся тегом rect, имеющим вид

```
<rect x="50"y="20"width="150"height="150"/>
```

Чтобы «закруглить» углы, достаточно добавить атрибуты rx и ry, задающие радиусы закругления. Например,

```
<rect x="50"y="20"width="150"height="150"rx="20"ry="20"/>
```

Работоспособность программы нужно проверить на наборе SVG-файлов, загруженных из интернета или созданных в векторном графическом редакторе.

## 3. Решение

### 3.1. Код

```
#include <bits/stdc++.h>
#include <sys/dir.h>

using namespace std;

using lineVec = vector<string>;

string getFileExtension(string path) { return
    path.substr(path.find_last_of('.') + 1); }

vector<string> getAllFilesInDirectory(string path, string
extension) {
    vector<string> result;
    DIR *wd = opendir(path.c_str());

    if (wd == NULL) {
        perror("opendir");
        return result;
    }

    for (struct dirent *ent; (ent = readdir(wd)) != NULL;) {
        if (ent->d_type == DT_REG) {
```

```

        string filename = path + "/" + ent->d_name;
        if (getFileExtension(filename) == extension) {
            result.push_back(filename);
        }
    }
}

return result;
}

lineVec getLinesFromFile(const string &path) {
    ifstream file(path);
    assert(file.good());

    lineVec result;

    for (string s; getline(file, s);)
        result.push_back(s);

    return result;
}

string strip(string l) {
    int i = 0;
    for (; i < l.size(); ++i)
        if (!isspace(l[i]))
            break;

    string r = "";
    for (; i < l.size(); ++i)
        r += l[i];

    return r;
}

bool hasSquareRect(string &l) {
    string s = strip(l);

    if (s[0] != '<' || s.back() != '>')
        return false;

    if (s.substr(1, 4) != "rect")
        return false;
}

```

```

if (s.find("rx") != string::npos && s.find("ry") !=
→   string::npos)
    return false;

return true;
}

bool hasRoundRect(string &l) {
    string s = strip(l);

    if (s[0] != '<' || s.back() != '>')
        return false;

    if (s.substr(1, 4) != "rect")
        return false;

    if (s.find("rx") != string::npos || s.find("ry") !=
→   string::npos)
        return true;

    return false;
}

void fixSquareRect(string &r) {
    if (r.find("rx") == string::npos)
        r = r.substr(0, r.size() - 2) + " rx=\"20\"/>";
    if (r.find("ry") == string::npos)
        r = r.substr(0, r.size() - 2) + " ry=\"20\"/>";
}

void fixRoundRect(string &r) {
    string res;
    auto rx = r.find("rx");
    auto ry = r.find("ry");

    set<int> exclude;

    for (int i = rx, c = 0; i < r.size(); ++i) {
        exclude.insert(i);
        if (r[i] == '"')
            c++;
        if (c == 2)
            break;
    }
}

```

```

        for (int i = ry, c = 0; i < r.size(); ++i) {
            exclude.insert(i);
            if (r[i] == '##')
                c++;
            if (c == 2)
                break;
        }

        for (int i = 0; i < r.size(); ++i)
            if (exclude.find(i) == exclude.end())
                res += r[i];

        r = res;
    }

    void fixSVGFile(const string &path) {
        auto lines = getLinesFromFile(path);
        for (auto &x : lines)
            if (hasSquareRect(x))
                fixSquareRect(x);
            else if (hasRoundRect(x))
                fixRoundRect(x);

        // const string newPath = path.substr(0,
        // → path.find_last_of('.')) + "-fixed.svg";
        auto newPath = path;
        ofstream output(newPath);

        for (auto &x : lines)
            output << x << '\n';

        output.close();
    }

    int main() {
        string path = "svgs";
        for (auto file : getAllFilesInDirectory(path, "svg"))
            fixSVGFile(file);
    }
}

```

Код 1: main.cc

## 3.2. Скриншоты

```
<svg viewBox='0 0 105 100' xmlns='http://www.w3.org/2000/svg'>
    <rect x="50" y="20" width="15" height="15" rx="20" ry="20"/>
    <rect x="50" y="20" width="15" height="15" rx="20"   />
    <rect x="50" y="20" width="15" height="15" ry="30"   />
    <rect x="50" y="20" width="15" height="15"   />
</svg>
```

Рис. 1: До запуска

```
<svg viewBox='0 0 105 100' xmlns='http://www.w3.org/2000/svg'>
    <rect x="50" y="20" width="15" height="15"   />
    <rect x="50" y="20" width="15" height="15" rx="20"   ry="20"/>
    <rect x="50" y="20" width="15" height="15" ry="30"   rx="20"/>
    <rect x="50" y="20" width="15" height="15"   rx="20" ry="20"/>
</svg>
```

Рис. 2: После запуска