

Частное образовательное учреждение высшего образования
“ЮЖНЫЙ УНИВЕРСИТЕТ (ИУБиП)”

Индивидуальное задание №1

по дисциплине: «Практикум по программированию»

Тема 4. Типы данных, определяемые пользователем
Вариант 15.

Работу выполнил:
студент группы ИД 401
Недвигин С. В.

Проверил:
преподаватель к.т.н. Сафонова С. А.

Задание: создать файл с полями: название альбома, название музыкального произведения, фамилия исполнителя, длительность произведения. Вывести список исполнителей указанного альбома. Определить исполнителя с наибольшим количеством записей операций.

Листинг кода.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <map>
#include <algorithm>

struct MusicPiece {
    std::string albumName;
    std::string title;
    std::string artistLastName;
    int duration; // Время в секундах
};

std::vector<MusicPiece> readMusicData(const std::string& filename) {
    std::vector<MusicPiece> musicData;
    std::ifstream file(filename);

    if (!file.is_open()) {
        std::cerr << "Unable to open file: " << filename << std::endl;
        return musicData;
    }

    std::string album, title, artist;
    int duration;

    while (file >> album >> title >> artist >> duration) {
        musicData.push_back({album, title, artist, duration});
    }

    file.close();
    return musicData;
}

void printArtistsOfAlbum(const std::vector<MusicPiece>& musicData, const
std::string& targetAlbum) {
```

```

std::cout << "Artists of album " << targetAlbum << ":" << std::endl;

for (const auto& piece : musicData) {
    if (piece.albumName == targetAlbum) {
        std::cout << piece.artistLastName << std::endl;
    }
}

}

std::string findMostProlificArtist(const std::vector<MusicPiece>& musicData) {
    std::map<std::string, int> artistCounts;

    for (const auto& piece : musicData) {
        artistCounts[piece.artistLastName]++;
    }

    auto mostProlific = std::max_element(artistCounts.begin(), artistCounts.end(),
        [](const auto& a, const auto& b) {
            return a.second < b.second;
        });

    return mostProlific->first;
}

int main() {
    std::string filename = "music.txt";
    std::cout << "Enter the target album: ";
    std::string targetAlbum;
    std::cin >> targetAlbum;

    std::vector<MusicPiece> musicData = readMusicData(filename);

    if (!musicData.empty()) {
        printArtistsOfAlbum(musicData, targetAlbum);

        std::string mostProlificArtist = findMostProlificArtist(musicData);
        std::cout << "Most prolific artist: " << mostProlificArtist << std::endl;
    }
}

```

```

    return 0;
}

```

Скриншот выполнения работы представлен на рисунке 1.

The screenshot shows a C++ IDE with a code editor on the left and a console window on the right. The code in the editor is as follows:

```

18. if (!file.is_open()) {
19.     std::cerr << "Unable to open file: " << filename << std::endl;
20.     return musicData;
21. }
22.
23. std::string album, title, artist;
24. int duration;
25.
26. while (file >> album >> title >> artist >> duration) {
27.     musicData.push_back(album, title, artist, duration);
28. }
29.
30. file.close();
31. return musicData;
32. }
33.
34. void printArtistsOfAlbum(const std::vector<MusicPiece>& musicData, const std::string& targetAlbum) {
35.     std::cout << "Artists of album " << targetAlbum << ": " << std::endl;
36.
37.     for (const auto& piece : musicData) {
38.         if (piece.albumName == targetAlbum) {
39.             std::cout << piece.artistLastName << std::endl;
40.         }
41.     }
42. }
43.
44. std::string findMostProlificArtist(const std::vector<MusicPiece>& musicData) {
45.     std::map<std::string, int> artistCounts;
46.
47.     for (const auto& piece : musicData) {
48.         artistCounts[piece.artistLastName]++;
49.     }
50.
51.     auto mostProlific = std::max_element(artistCounts.begin(), artistCounts.end(),
52.                                         [](const auto& a, const auto& b) {
53.                                             return a.second < b.second;
54.                                         });
55.
56.     return mostProlific->first;
57. }
58.
59. int main() {
60.     std::string filename = "music.txt";
61.     std::cout << "Enter the target album: ";
62.     std::string targetAlbum;
63.     std::cin >> targetAlbum;
64.
65.     std::vector<MusicPiece> musicData = readMusicData(filename);
66.
67.     if (!musicData.empty()) {
68.         printArtistsOfAlbum(musicData, targetAlbum);
69.
70.         std::string mostProlificArtist = findMostProlificArtist(musicData);
71.         std::cout << "Most prolific artist: " << mostProlificArtist << std::endl;
72.     }
73.
74.     return 0;
75. }

```

The console window on the right shows the following output:

```

Run 87ms on 18:42:40, 01/30 ✓
Run 9s on 18:46:28, 01/30 ✓
Run 122ms on 18:47:31, 01/30 ✓
Run 19s on 18:48:02, 01/30 ✓
Enter the target album: Back in black

```

Рисунок 1

Вывод: в ходе выполнения работы освоены программы простейшей структуры; получены практические навыки по написанию и отладке программ сортировки массивов структур.