Презентация проекта

Цели проекта

1

Повторение материала

Вспомнить работу с бинарными файлами, пользовательскими типами данных, текстовыми файлами

2

Работа в команде

Разбиение проекта на личные подзадачи для дальнейшего их решения

3

Работа с github

Непосредственная работа с github

Main

Main содержит все главные функции предложенных заданий.

TextToBinaryStudents — функция считывающая информацию из первого файла

TextToBinaryStudentsCards — функция считывающая информацию из второго файла

BinaryToAsciiText и PrintBinaryFile — функции использующиеся на протяжении всей работы, функции преобразования данных из бинарного файла в текст и преобразования текста в бинарный код соответственно

```
try
{
    int studentSize = CountStudentsInTextFile("students.txt");
    int cardSize = CountStudentsInTextFile("students-marks.txt");

    if (studentSize != cardSize)
    {
        std::cout << "The number of students and cards does not coincide!\n";
        return 1;
    }

    TextToBinaryStudents("students.txt", "students.bin");
    StudentsBinToTxt("students.bin", "students-output.txt");

    TextToBinaryStudentsCards("students-marks.txt", "students-mark.bin");
    StudentCardsBinToTxt("students-mark.bin", "students-mark-output.txt");

    Student* students = new Student[studentSize];
    BinToStudent(students, studentSize, "students.bin");

    StudentCard* studentsCards = new StudentCard[cardSize];
    BinToStudentCard(studentsCards, cardSize, "students-mark.bin");
```

Проверки ввода/вывода файлов

```cpp
void CheckInputFile(std::ifstream& fin)
{
    if(!fin.is_open())
    {
        throw std::runtime_error("Binary file couldn't be opened!");
    }
    if(fin.fail())
    {
        throw std::runtime_error("Input binary file error!");
    }
    if(fin.bad())
    {
        throw std::runtime_error("Critical error with binary file!");
    }
}
```

Ввод для бинарных файлов

```cpp
void CheckOutputFile(std::ofstream& fout)
{
    if(!fout.is_open())
    {
        throw std::runtime_error("Binary file couldn't be opened for writing!");
    }
    if(fout.fail())
    {
        throw std::runtime_error("Output binary file error!");
    }
    if(fout.bad())
    {
        throw std::runtime_error("Critical error with output binary file!");
    }
}
```

Вывод бинарных файлов

```cpp
void CheckTextFile(std::ifstream& fin)
{
    if(!fin.is_open())
    {
        throw std::runtime_error("Text file couldn't be opened!");
    }
    if(fin.fail())
    {
        throw std::runtime_error("Input text file error!");
    }
    if(fin.bad())
    {
        throw std::runtime_error("Critical error with text file!");
    }
}
```

Ввод для текстовых файлов

```cpp
void TaskB(Student* students, StudentCard* studentsCards, int size, const std::string& binaryFileName)
{
    std::ofstream binFile(binaryFileName, std::ios::binary); CheckOutputFile(binFile);
    for(int i{} ; i < size ; i++)
    {
        std::string line = std::to_string(studentsCards[i].group) +
            " " + std::to_string(studentsCards[i].id) + " " +
            students[i].surname + " " +
            studentsCards[i].firstSubject + " " + std::to_string(studentsCards[i].firstMark) + " " +
            studentsCards[i].secondSubject + " " + std::to_string(studentsCards[i].secondMark) + " " +
            studentsCards[i].thirdSubject + " " + std::to_string(studentsCards[i].thirdMark) + "\n";
        binFile.write(line.c_str(), line.size());
    }
}

void TaskBToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary); CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName); CheckOutputFile(txtFile);
    std::string line;
    while(std::getline(binFile,line))
    {
        txtFile << line << "\n";
    }
}
```

Задание B

Данные функции объединяют фамилию и оценки студентов и записывают их в бинарный и текстовый файлы

```cpp
void TaskC(Student* students, StudentCard* studentsCards, int size, const std::string& binaryFileName)
{
    std::ofstream binFile(binaryFileName, std::ios::binary); CheckOutputFile(binFile);
    for (int i{}; i < size; i++)
    {
        double average = (studentsCards[i].firstMark + studentsCards[i].secondMark + studentsCards[i].thirdMark) / 3.0;
        std::string line = std::to_string(studentsCards[i].group) +
            " " + std::to_string(studentsCards[i].id) + " " + students[i].surname + " " + std::to_string(average) + "\n";
        binFile.write(line.c_str(), line.size());
    }
}

void TaskCToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary);
    CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName);
    CheckOutputFile(txtFile);
    std::string line;
    while (std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}
```

Задание C

Данные функции считают средний балл студента и записывают их в бинарный и текстовый файлы

```cpp
void TaskD(Student* students, StudentCard* studentsCards, int size, const std::string& binaryFileName)
{
    std::ofstream binFile(binaryFileName, std::ios::binary);
    CheckOutputFile(binFile);
    for (int i{}; i < size; ++i)
    {
        double average{ (studentsCards[i].firstMark + studentsCards[i].secondMark + studentsCards[i].thirdMark) / 3.0 };
        if (average < 4)
        {
            std::string line = students[i].surname + " " + std::to_string(studentsCards[i].group) + " " + std::to_string(studentsCards[i].id) + "\n";
            binFile.write(line.c_str(), line.size());
        }
    }
}

void TaskDToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary);
    CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName);
    CheckOutputFile(txtFile);
    std::string line;
    while (std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}
```

Задание D

Данные функции ищут студентов со средним баллом меньше 4-ех и записывают их в бинарный и текстовый файлы

```cpp
void WriteBadStudentsToBinFile(const std::string& fileName, BadStudent* students, int size)
{
    std::ofstream binFile(fileName, std::ios::binary);

    for (int i {}; i < size; ++i)
    {
        std::string line = students[i].surname + " " +
            std::to_string(students[i].group) + " " + std::to_string(students[i].id) + "\n";
        binFile.write(line.c_str(), line.size());
    }
}

BadStudent* FillBadStudentList(const std::string& binaryFileName,const int outSize)
{
    std::ifstream binFile(binaryFileName, std::ios::binary);
    BadStudent* badStudents = new BadStudent[outSize];
    int i{};
    std::string line;
    while (std::getline(binFile, line) && i < outSize)
    {
        std::istringstream iss(line);
        iss >> badStudents[i].surname
            >> badStudents[i].group
            >> badStudents[i].id;
        ++i;
    }


    return badStudents;
```

```cpp
void SortBadStudent(BadStudent* badStudent, int outSize)
{
    for (int i{}; i < outSize - 1; ++i)
    {
        for (int j{}; j < outSize - i - 1; ++j)
        {
            bool swapNeeded = false;
            if (badStudent[j].group > badStudent[j + 1].group)
            {
                swapNeeded = true;
            }
            else if (badStudent[j].group == badStudent[j + 1].group &&
                badStudent[j].surname > badStudent[j + 1].surname)
            {
                swapNeeded = true;
            }

            if (swapNeeded)
            {
                BadStudent temp = badStudent[j];
                badStudent[j] = badStudent[j + 1];
                badStudent[j + 1] = temp;
            }
        }
    }
}

void TaskEToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary);
    CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName);
    CheckOutputFile(txtFile);
    std::string line;
    while (std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}
```

Задание E

Данные функции сортируют студентов с оценкой меньше 4-ех в алфавитном порядке и записывают их в бинарный и текстовый файлы

```cpp
void TaskDToTxtMod(std::ifstream& binFile, std::ofstream& txtFile)
{
    std::string line;
    while(std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}

void TaskEToTxtMod(std::ifstream& binFile, std::ofstream& txtFile)
{
    std::string line;
    while(std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}

void TaskF(const std::string& beforeSortBin, const std::string& afterSortBin, const std::string& outputTxtFile)
{
    std::ofstream txtFile(outputTxtFile);
    CheckOutputFile(txtFile);
    std::ifstream binFileBeforeSort(beforeSortBin,std::ios::binary );
    CheckOutputFile(txtFile);
    std::ifstream binFileAfterSort(afterSortBin,std::ios::binary);
    CheckOutputFile(txtFile);
    txtFile << "Неуспевающие (до сортировки):\n";
    TaskDToTxtMod(binFileBeforeSort, txtFile);
    txtFile << "\nНеуспевающие (после сортировки):\n";
    TaskEToTxtMod(binFileAfterSort, txtFile);
}
```

Задание F

Функции записывают данные об учениках до сортировки и после и записывают их в бинарный и текстовый файлы

```cpp
void TaskG(Student* students, StudentCard* cards, int size, int groupNum, const std::string& fileName)
{
    int* groupIndices = new int[size];
    int count{};

    for (int i{}; i < size; ++i)
    {
        if (cards[i].group == groupNum)
        {
            groupIndices[count++] = i;
        }
    }

    for (int i{}; i < count - 1; ++i)
    {
        for (int j{}; j < count - i - 1; ++j)
        {
            if (students[groupIndices[j]].surname > students[groupIndices[j + 1]].surname)
            {
                std::swap(groupIndices[j], groupIndices[j + 1]);
            }
        }
    }

    std::ofstream fout(fileName, std::ios::binary);
    CheckOutputFile(fout);

    for (int i{}; i < count; ++i)
    {
        int idx = groupIndices[i];
        std::string line = std::to_string(cards[idx].group) + " " +
                           std::to_string(cards[idx].id) + " " +
                           students[idx].surname + " " +
                           cards[idx].firstSubject + ":" + std::to_string(cards[idx].firstMark) + " " +
                           cards[idx].secondSubject + ":" + std::to_string(cards[idx].secondMark) + " " +
                           cards[idx].thirdSubject + ":" + std::to_string(cards[idx].thirdMark) + "\n";

        fout.write(line.c_str(), line.size());
    }

    delete[] groupIndices;
}
```

```cpp
void TaskGToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary);
    CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName);
    CheckOutputFile(txtFile);

    std::string line;
    while (std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}
```

Задание G

Функции формируют ведомость оценок упорядоченной по алфавиту и записывают их в бинарный и текстовый файлы

```cpp
void TaskH(Student* students, StudentCard* cards, int size, int groupNum, const std::string& fileName)
{
    StudentAvg* studentAvgs = new StudentAvg[size];
    int count{};

    for (int i{}; i < size; ++i)
    {
        if (cards[i].group == groupNum)
        {
            double avg{(cards[i].firstMark + cards[i].secondMark + cards[i].thirdMark) / 3.0};
            studentAvgs[count].index = i;
            studentAvgs[count].average = avg;
            count++;
        }
    }

    for (int i{}; i < count - 1; ++i)
    {
        for (int j{}; j < count - i - 1; ++j)
        {
            if (studentAvgs[j].average < studentAvgs[j + 1].average)
            {
                std::swap(studentAvgs[j], studentAvgs[j + 1]);
            }
        }
    }

    std::ofstream fout(fileName, std::ios::binary);
    CheckOutputFile(fout);

    for (int i{}; i < count; ++i)
    {
        int idx = studentAvgs[i].index;
        std::string line = std::to_string(cards[idx].group) + " " +
                           std::to_string(cards[idx].id) + " " +
                           students[idx].surname + " " +
                           cards[idx].firstSubject + ":" + std::to_string(cards[idx].firstMark) + " " +
                           cards[idx].secondSubject + ":" + std::to_string(cards[idx].secondMark) + " " +
                           cards[idx].thirdSubject + ":" + std::to_string(cards[idx].thirdMark) + " " +
                           "Avg:" + std::to_string(studentAvgs[i].average) + "\n";

        fout.write(line.c_str(), line.size());
    }

    delete[] studentAvgs;
}

void TaskHToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary);
    CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName);
    CheckOutputFile(txtFile);
    std::string line;
    while(std::getline(binFile, line))
    {
        txtFile << line << "\n";
    }
}
```

Задание H

Функции формируют ведомость оценок упорядоченной по убыванию среднего балла и записывают их в бинарный и текстовый файлы

```cpp
void TaskI(Student* students, StudentCard* studentsCards, int size, const std::string& binaryFileName)
{
    std::ofstream binFile(binaryFileName, std::ios::binary); CheckOutputFile(binFile);
    for (int i{}; i < size; ++i)
    {
        double average{(studentsCards[i].firstMark + studentsCards[i].secondMark + studentsCards[i].thirdMark)/ 3.0};
        if (average >= 8)
        {
            std::string line = students[i].surname  + " " + std::to_string(studentsCards[i].group) +
                " " + std::to_string(studentsCards[i].id) + "\n";
            binFile.write(line.c_str(), line.size());
        }
    }
}

void TaskIToTxt(const std::string& binFileName, const std::string& txtFileName)
{
    std::ifstream binFile(binFileName, std::ios::binary); CheckInputFile(binFile);
    std::ofstream txtFile(txtFileName); CheckOutputFile(txtFile);
    std::string line; while (std::getline(binFile, line)) txtFile << line << "\n";
}
```

Задание I

Функции формируют список отличников и записывают их в бинарный и текстовый файлы

Спасибо за внимание

Над кодом работали: И. Сергей, В. Александр, а также лидер команды — Е. Михаил