



Обектно-ориентирано програмиране

Домашна работа №1

Пояснение:

- Реализирайте задачите спазвайки добрите ООП практики (валидация на данните, подходяща капсулация и тн.)
- **Решение, в които не са спазени ООП принципите ще бъдат оценени с 0 точки.**
- Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.
- **Не е разрешено** да ползвате библиотеки от STL и STL функции.

Изисквания за предаване:

- Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно и при установено плагиатство ще бъдат **анулирани**.
- Предаване на домашното в указания срок от всеки студент като .zip архив със следното име:

(номер_на_домашно)_SI_(курс)_(група)_(факултетен_номер)

- (номер_на_домашно) е цяло число, отговарящо на номера на домашното за което се отнася решението (например 1);
- (курс) е цяло число, отговарящо на курс (например 1);
- (група) е цяло число, отговарящо на групата Ви (например 1);
- (факултетен_номер) е цяло число, отговарящо на факултетния Ви номер (например 12345);

Пример за .zip архив за домашно: 1_SI_1_1_12345.zip

Архивът да съдържа само изходен код (.cpp и .h/.hpp файлове) с решение отговарящо на условията на задачите, като файловете изходен код за всяка задача трябва да са разположени в папка с име (номер_на_задача).

Качване на архива на посоченото място в Moodle





Обектно-ориентирано програмиране

Домашна работа №1

Задача 1 - Markdown tables parser (6 точки)

Да се имплементира програма за работа с markdown таблици от текстови файлове.

Вашата програма трябва да зарежда такава таблица от файл и да може да извършва следните операции:

- Отпечатване на данните в таблицата
- Промяна на име на колона
- Добавяне на ред.
- Промяна на стойност в колона по подаден номер на ред(започващо от 1) и име на колона.
- Промяна на стойност в колона по подадено име на колона и стойност. Ако има няколко срещания на стойността, която искаме да заменим, да се замени първото срещане.
- Селектиране - отпечатване на всички редове, които имат подадена стойност на подадена колона.
- Запазване на таблицата във файл.

Примерно съдържание на **table.md**:

Syntax	Description
-----	-----
Header	Title
Paragraph	Text

Забележка 1:

При четенето от файла между полетата може да има **произволен брой интервали**.

И големината на колоните може да е различна, но това не трябва да влияе на отпечатването.

Syntax	Description
-----	-----
Header	Title
Paragraph	Text

← са еквивалентни →

Syntax	Description
---	-----
Header	Title
Paragraph	Text

Забележка 2:

При прочитането от файла всяка колона може да бъде подравнена.

Подравняването става със символа ' : '

Syntax	Description	Test Text
:---	:----:	---:
Header	Title	Here's this
Paragraph	Text	And more

При отпечатване Syntax трябва да е подравнено вляво, Description трябва да е подравнено в средата и Test Text трябва да е подравнено вдясно. Ако няма срещане на :, то по подразбиране е подравнено вляво.

Ограничения:

Ще работите с таблици с най-много 10 колони и най-много 50 реда.

Всяка стойност ще е с най-много 20 символа.





Обектно-ориентирано програмиране

Домашна работа №1

Примерен интерфейс:

```
>table.md
Table loaded successfully!
> print
| Syntax      | Description |
| -----    | -----    |
| Header      | Title       |
| Paragraph   | Text        |

> change Description Desc
Operation successfully executed!
> print
| Syntax      | Desc        |
| -----    | -----    |
| Header      | Title       |
| Paragraph   | Text        |

> changeRow 2 Desc None
Operation successfully executed!
> print
| Syntax      | Desc        |
| -----    | -----    |
| Header      | Title       |
| Paragraph   | None        |

> addRow Test None
Operation successfully executed!
> print
| Syntax      | Desc        |
| -----    | -----    |
| Header      | Title       |
| Paragraph   | None        |
| Test       | None        |

> selectWhere Desc None
| Syntax      | Desc        |
| -----    | -----    |
| Paragraph   | None        |
| Test       | None        |

> change Desc None Empty
| Syntax      | Desc        |
| -----    | -----    |
| Header      | Title       |
| Paragraph   | Empty       |
| Test       | None        |

>save NewTable.md
File successfully saved!
```





Обектно-ориентирано програмиране

Домашна работа №1

Задача 2 – TextFilesManager (4 точки)

Да се имплементира система, която менажира текстови файлове. Всеки файл има:

- Име (до 64 символа)
- Съдържание (до 512 символа)
- Размер (в байтове)
- Време на създаване
- Време на последна модификация
- Права за достъп

Времената са във формат дата, часове, минути, секунди.

Правата на достъп са представени като стринг от точно 9 символа.

Те са разделени в 3 групи по 3 символа.

- първата е за потребителя – собственик на файла
- втората – за групата собственик
- третата – за всички останали.

Всяка от тези групи има 3 вида права, отбелязани с по 1 символ

- за четене (r),
- за писане (w)
- за изпълнение (x).
- Ако някоя група няма някое право, това се отбелязва с -.

Тоест, при стринг "rwxr-xr-" означава, че потребителят собственик има права да пише, чете и изпълнява файла, групата собственик – да чете и изпълнява и всички останали – само да четат файла.

При изпълнението на всяка операция върху файл **се приема като параметър ролята на потребителя, който я изпълнява.**

Системата от файлове трябва да поддържа определен максимален брой файлове (**броят се подава в конструктора и за този максимален брой трябва да бъде заделено точно количество памет**) и следните функционалности:

- Да се създаде нов празен файл в системата (като се подава и времето на създаване)
- Да се редактира (презаписва съдържанието на) вече създаден файл по име (като се подава и времето на модификацията)
- Да се добавя текст към вече създаден файл по име (като се подава и времето на модификацията)
- Да се изтрива от системата файл по име
- Да се променят правата на даден файл, като се подава символ за групата, за която ще се променят правата (и за потребител собственик, g за група собственик, o за други) и символ за типа права – ако го няма, да се добави, ако го има, да се премахне
- Да се принтира цялата информация за даден файл
- Да се принтира съдържанието за даден файл.
- Да се принтират всички файлове.
- Да се сортират всички файлове по:
 - Име
 - Време на създаване
 - Време на последна модификация
 - Размер





Обектно-ориентирано програмиране

Домашна работа №1

Примерен интерфейс:

```
TextFileManager fs(5);
fs.addFile("test.txt", 14, 0, 0, 20, 3, 2023, "rwxr—r—");

fs.editFile("test.txt", "Hello, world!", 14, 10, 0, 20, 3, 2023, u); // Hello world!
                                                                    // Last modified on: 14:10:00 20.03.2023

fs.editFile("test.txt", "Hello, world!", 14, 11, 0, 20, 3, 2023, o); // Error! You do not have the rights to edit!

fs.addInFile("test.txt", "Hello, my friend!", 14, 13, 0, 20, 3, 2023, u); // Hello world!
                                                                    // Hello, my friend!
                                                                    // Last modified on: 14:13:00 20.03.2023

fs.editFile("test.txt", "Hello", 14, 15, 0, 20, 3, 2023, u);          // Hello

fs.addFile("attest.txt", 14, 30, 0, 20, 3, 2023, "rwxr—r—");
fs.deleteFile("attest.txt");

fs.addFile("attest.txt", 14, 35, 0, 20, 3, 2023, "rwxr—r—");

fs.changeRights("attest.txt", u, w);          //r-xr—r—

fs.changeRights("attest.txt", u, w);          //rwxr—r—

fs.changeRights("attest.txt", u, r);          //-wxr—r—
fs.printFile("attest.txt", u);                // Error! You do not have the rights to read!

fs.printFile("test.txt", o);                  // Hello

fs.printFile("attest.txt", o);                //

fs.printFileInfo("test.txt");
    // Name: test.txt
    // Size: 5
    // Created on: 14:00:00 20.03.2023
    // Last modified on: 14:15:00 20.03.2023
    // Rights: rwxr—r—
fs.sort(SortOptions::name);
fs.print();
    //attest.txt
    //test.txt

fs.sort(SortOptions::modificationTime);
fs.print();//test.txt
    //attest.txt
```

