

Задание на проект по Структури от данни и програмиране - практикум

Проектът е цялостна задача, която трябва да решите с помощта на познанията по C++, получени през последните три семестъра. Правилата за завършване на курса по практикума “Структури от данни и програмиране” с проект са следните:

1. Трябва да предадете:
 - документация на проекта;
 - изходен код на решението;
 - изпълним файл с решението;
 - няколко примера, подбрани от Вас, които демонстрират работата на задачата.
2. В решението на задачата задължително трябва да използвате на парадигмата на ООП, т.е. да решите задачата с класове.
3. Документацията на проекта трябва да съдържа:
 - o кратък анализ на задачата и Вашия подход за решение (на какви стъпки сте разделили решението, какъв метод или алгоритъм сте избрали, как сте се справили с конкретен проблем)
 - o кратко описание на класовете, създадени от Вас за решение на задачата, (описание на член-данните и член-функциите им и начина на използване на класовете)
 - o идеи за бъдещи подобрения
4. По време на защитата трябва да разкажете в рамките на 10 минути Вашето решение и да демонстрирате работата на програмата с подготвени от Вас данни.
5. Възможно е да ви дадем малка задача за допълнение или промяна на функционалността на проекта ви, която вие трябва да реализирате на място за максимум 1 час.
6. Невъзможност да реализирате малката задача за промяна на проекта на място означава, че не познавате добре проекта си и сте ползвали чужда помощ за реализацията му. Последното ще се отрази негативно на оценката ви.
7. Критерии за оценка на проекта (% от оценката):
 - o правилно решение на задачата- 45%

- o добре подбрана структура на класовете решаващи задачата- 20%
- o подреден и читаем код - 10%
- o подходяща и ясна документация - 10%
- o представяне на проекта - 15%
- o възможност за бърза реализация на малка промяна в проекта (познаване на кода) – при невъзможност, оценката става слаба.

8. Тези от вас, които желаят може да направят и презентация на реализирания проект.

IML

Иван си е измислил език от тагове, но няма как да пише на него. Да се напише parser за езика IML (Ivan Markup Language). Езикът съдържа 8 tag-a: шест от тях имат и съответен затварящ и два нямат и са единични. Потребителят въвежда две имена на файлове. Първото име е за входен, а второто да изходен файл, като вашата програма трябва да parse-не съдържанието на входния файл в изходния. Ако във входния файл, участват и тагове извън езика да се изведе съобщение за грешка. Иван иска да се позволи влагане или вгнездяване на tag-ове. Т.е. във всеки произволен момент от четенето на файла, трябва да има повече отварящи, отколкото затварящи се tag-ове или броят им да бъде равен. Ако това не е така програмата извежда съобщение за грешка на стандартния изход за грешки.

1) `<trim>TEXT</trim>`

Ако някъде в TEXT има няколко whitespace-a един до друг -> оставя само първия.

2) `<toupper>TEXT</toupper>`

Извежда TEXT с главни букви.

3) `<tolower>TEXT</tolower>`

Извежда TEXT с малки букви.

4) `<hex>NUMBER</hex>`

Извежда в шестнайсетична бройна система записаното в десетична такава число NUMBER.

Забележка: буквите в шестнайсетичния запис зависят от tag-овете tolower и toupper, като по подразбиране са главни.

5) `<oct>NUMBER</oct>`

Извежда в осмична бройна система записаното в десетична такава число NUMBER.

6) `<pre > NUMBER</pre>`

Извежда NUMBER като число с два знака след десетичната точка.

7) <nl>

Извежда нов ред.

Забележка: това е единственият начин да се изведе нов ред.

Т.е. новите редове в обикновения текст се игнорират

8) <tb>

Извежда 5 последователни space-a.

Ако някъде в tag-a toupper/tolower се среща tag tolower/toupper, то вътрешният tag override-ва външния. Същото важи и за tag-овете hex и oct.