

# Приложение за работа с електронни таблици

## Представяне на данните

Данните на една таблица ще записваме в текстов файл по следния начин:

1. Всеки ред във файла представя отделен ред в таблицата.
2. Всеки ред във файла съдържа данни разделени със запетаи. Тези данни се интерпретират като стойностите в клетките на реда.
3. Всеки ред в таблицата може да съдържа различен брой клетки. Затова и всеки ред във файла може да съдържа различен брой елементи разделени със запетаи.
4. Празен ред във файла представя празен ред в таблицата. (т.е. ред, в който всички клетки са празни).
5. Между две запетаи във файла може да няма никакви данни. По този начин се представя празна клетка.
6. Между данните и запетаите може да има произволен брой празни символи (whitespace).

Така за една таблица може да има различни представяния. Например таблицата:

10	20	30	40
10		1000	
	10		

може да се представи по следните начини (възможни са и други представяния):

10, 20, 30, 40	10, 20 , 30 , 40
10,,1000,	10, , 1000,
,,,	, , ,
,10	, 10

## Типове данни в таблицата

Всяка клетка в таблицата има тип, като в една таблица може да има едновременно клетки от различни типове. Вашето приложение трябва да може да поддържа следните типове:

**Цяло число** – поредица от цифри, без никакви други символи между тях. В началото на числото може да има знак '+' или '-'. Например:

123  
-123  
+123

**Дробно число** – поредица от цифри, следвана от символ за точка и след нея друга поредица от цифри. В началото на числото може да има знак '+' или '-'. Например:

123.456  
-123.456  
+123.456

**Символен низ (стринг)** – поредица от произволни символи оградени в кавички. Подобно на низовете в C++, ако искате да включите символа за кавичка в даден низ, трябва да го представите като '\', а ако искате да включите наклонена черта, трябва да я представите като '\\. Например:

"Hello world!"  
"C:\\temp\\"  
"\\"This is a quotation\\""

**Дата** – поредица от три числа, разделени с точка, представляваща датата във формат dd.mm.yyyy или yyyy.mm.dd. Например:

02.12.2018  
15.3.1999  
2018.05.01

## Нужна функционалност

След като вашето приложение отвори даден файл, то трябва да може да извършва посочените по-долу операции:

load	Зарежда таблица от файл.
save	Записва таблицата във файл.
sort	Подрежда редовете на таблицата според стойностите в подадена колона
print	Извежда съдържанието на таблицата на екрана.
edit	Редактира съдържанието на дадена клетка. За целта потребителят въвежда текст, който ще бъде новото съдържание на клетката. Забележете, че по този начин може да се промени типът на дадена клетка, например от число, тя може да стане символен низ.

Ако при зареждането на данните, приложението ви открие грешка, то трябва да изведе подходящо съобщение за грешка и да прекрати своето изпълнение. Съобщението трябва да подсказва на потребителя какво не е наред във входните данни. Например:

- Ако липсва запетая трябва да се изведе на кой ред и след кой символи липсва запетаята;
- Ако съдържанието на дадена клетка е от неизвестен тип, трябва да се изведе на кой ред и коя колона е клетката и какво точно е некоректното съдържание. Например нека предположим, че на ред 2, колона 5, потребителят е въвел 123.123.123. Приложението ви може да изведе например следното съобщение: *"Error: row 2, col 5, 123.123.123 is unknown data type"*.

## Извеждане на таблицата на екрана

При извеждане на заредената таблица (командата print), данните в колоните трябва да се подравнят. Между отделните колони трябва да се поставят символи за отвесна черта (|). По-долу е даден пример за входен файл и възможно негово извеждане:

Входен файл	Извеждане
10, "Hello world!",123.56	10   Hello world!   123.56
"\"Quoted\""	"Quoted"
1, 2, 3, 4	1   2   3   4

## Редактиране на клетки

Командата Edit трябва да позволява (с подходящи параметри) на потребителя да променя стойностите на отделните клетки. Това става като се укажат реда и колоната на клетката, която искаме да променим, а също и каква стойност да запише в нея. Потребителят може да въведе произволен тип данни, който се поддържа от вашата програма (например цяло число, дробно число, низ и т.н.).

Ако потребителят въведе неправилни данни, приложението ви не трябва да променя нищо в таблицата, а само да изведе на екрана съобщение, че са въведени неправилни данни. В този случай приложението ви НЕ трябва да прекратява своето изпълнение.

## Сортиране

При поискване на подреждане на редовете трябва да се подаде номер на колона според която да се подредят данните. Номерата на колоните от 1. Може да се подаде и параметър, указващ дали подреждането да е в нарастващ или намаляващ ред.

При сравняване на два елемента от един и същ тип се прилага естествената за типа наредба, При сравняване на данни от различен тип се прилага следното правило:

Числата се сравняват естествено.

При сравняване на дата с който и да е друг тип, датата се преобразува към число, което представлява броя дни, изминали след 01.01.1900.

При сравняване на низ с число, ако той може да се преобразува към число (т.е. е валидно число) се работи с тази стойност. Ако не е валидно число се приема стойност 0.

Следва примерна таблица и няколко начина, по които се сортира.

1	01.12.2000	text1
2.123	1900.02.21	baba
some data	Text data	12.4
text	432	123

Подредба в нарастващ ред по колона 1

some data	Text data	12.4
text	432	123
1	01.12.2000	text1
2.123	1900.02.21	baba

Подредба в нарастващ ред по колона 2

some data	Text data	12.4
2.123	1900.02.21	baba
text	432	123
1	01.12.2000	text1

Подредба в нарастващ ред по колона 3

2.123	1900.02.21	baba
1	01.12.2000	text1
some data	Text data	12.4
text	432	123

**Добавка за практикума:**

Представете адреса на всяка клетка като двойка от координати - ред/колона. Реализирайте подходящи оператори за индексване на ред / колона / клетка. Тези оператори трябва да позволяват четене, но и запис на данните в съответната клетка.

Реализирайте оператор за добавяне на ред / колона в края на таблицата.