

ЛЕКЦІЯ 13

□ Глибовець А.М.



STRING

- Ми продовжимо розбиратися з стрічками.
- Будемо розбиратися на прикладах.
- Давайте розберемося як порахувати символи в верхньому регістрі, що перебувають в вхідній стрічці.
- Як це зробити?
- Розберемо приклад CountUppercase



STRING

- Напишемо програму, що замінює підстрічки в стрічці.
- Тобто користувач надає нам стрічку і фрагмент, що необхідно замінити.
- Тобто в нас є 3 параметри:
 - оригінальна стрічка
 - підстрічка, яку необхідно замінити
 - підстрічка на яку необхідно замінити
- Який алгоритм?
- Розглянемо приклад ReplaceOccurence



ТОКЕНІЗАЦІЯ

- Токенізація – в нас є стрічка і ми її розбиваємо на токени.
- Токен – частина стрічки, що має з обох сторін проміжки.
- “Добрий день, студенти”
- Які токени у нас тут?
- Як зробити таку токенізацію?
- Давайте розберемося як це зробити в Java.



ТОКЕНІЗАЦІЯ

- Ми скористаємося вже написаними утилітами!
- `import java.util.*;`
- І використаємо клас – `StringTokenizer`
 - `String line = readLine();`
 - `StringTokenizer tokenizer = new StringTokenizer(line);`
- **Методи:**
 - `boolean hasMoreTokens()`
 - перевіряє чи zostалися в стрічці ще токени
 - `String nextToken();`
 - повертає нам наступний токен
- Розглянемо приклад `TokenizerExample`



РОБОТА З ФАЙЛАМИ

- Що таке файл?
- Навіщо нам файли?



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- Читання файлу починається з самого початку
- Читання відбувається побайтово/посимвольно/пострічно
- Загальна процедура:
 - для того, щоб почати працювати з файлом, необхідно спочатку відкрити файл.
 - прочитати інформацію з файлу
 - закрити файл
- Для роботи з файлами нам потрібно імпортувати `java.io.*`;



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- Для відкриття файлу ми скористаємося наступними класами:
 - BufferedReader
 - FileReader
- Приклад:
 - `BufferedReader rf = new BufferedReader(new FileReader("file.txt"));`
 - або
 - `FileReader fr = new FileReader("file.txt");`
 - `BufferedReader rf = new BufferedReader(fr);`
- Питання?



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- Ми з вами відкрили файл, тепер необхідно прочитати з нього інформацію.
- Завдяки `BufferedReader` ми можемо досить просто організувати зчитування інформації стрічку за стрічкою.
- Приклад:
 - `while (true){`
 - `String s = rf.readLine();`
 - `if (s==null) break;`
 - `println(s);`
 - `}`
 - `rd.close();`
- Питання?



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- Чи готові ми з вами працювати з файлом?
- Що відбудеться якщо файлу який ми пробуємо відкрити не існує?
- Відбудеться виключна ситуація – Exception
- Що робить `BufferedReader`?
- `BufferedReader` кидає (throw) повідомлення про виключну ситуацію
- Якщо хтось throw exception хтось має отримати це повідомлення і обробити його, якщо цього не відбудеться програма аварійно припинить свою роботу



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- ▣ Блок коду який може породити exception має бути записаний наступним чином:

- try{
 - ▣ //код що працює з файлом...
- } catch (IOException ex){
 - ▣ робимо щось ...
- }

- ▣ Питання?



ЧИТАННЯ ІНФОРМАЦІЇ З ФАЙЛУ

- Розглянемо приклад MyFileReader
- Питання?



ЗАПИС В ФАЙЛ

- Процедура запису в файл:
 - Відкрити
 - `PrintWriter`
 - Записати
 - `_.println()`
 - Закрити файл
 - `close()`
- Розглянемо приклад `CopyFile`
- Питання?



□ Дякую за увагу.

