|  |  |
| --- | --- |
| Звіт з лабораторної роботи № 3  Сучасні технології програмування – 1 : Функціональне програмування  Студент : Андрощук Максим Група : ІП-81  Мова виконання: Erlang | |
| Завдання № 1-3 | Реалізувати форматований вивід обраних таблиць в cli за запитом  load(“filename”) |
| Реалізація | 1. Run() – функція запускає програму. 2. Run(true) – запускається функція, яка очікує на вхід команду 3. FileName(Command) – функція, яка у команді знаходить ім’я файлу 4. Choose\_RowsCount(FileName) – повертає кількість рядків, які будуть прочитані із файлу(в залежності від розширення файлу, кількість рядків різна), це пов’язано із тим, що консоль в Erlang може виводити на екран лише певну кількість рядків, якщо виводити усі рядки із файлу, початок файлу не буде видно. 5. Read\_Line(File, CountLine) – функція, яка читає визначену кількість рядків із файлу та повертає список рядків. У даній функції прийшлось дописати костиль, який пов’язаний з особливостями Erlang. В одному із файлів порожні колонки позначаються комою, проте функція string: split вирізає повністю регулярний вираз із рядка(тобто вирізається кома та перший непробільний символ). Тому ми шукаємо в рядку 2 коми підряд(це означає два порожніх рядки, але для регулярного виразу вони означають 1 порожній рядок) і замінюємо на 4 коми підряд. Також, для того щоб перший непробільний символ не пропадав, ми доставляємо одну кому кожногу разу, коли бачимо наш регулярний вираз(тоді перша кома відповідає комі в регулярному виразі, а друга кома відповідає непробільному символу). 6. Choose\_Pattern(FileName) – функція, яка вибирає вираз, по якому будуть визначатись колонки(якщо це .csv файл, то це буде “,\\S “ ,що означає кома та непробільний знак, такий вираз потрібен через те, що в деяких колонках в містяться цілі речення, які відокремлюються комами, тому “сплітити” тільки по комі не вдасться. Якщо це .tsv файл, то патерном буде “\t”, адже там колонки розділяються табами) 7. StartParse(RowsList, Pattern) – функція, на вхід якій подається регулярний вираз та список рядків. Дана функція опрацьовує перший рядок, який відповідає назвам колонок і визначає кількість цих самих колонок і довжину однієї колонки(це зроблено для того, щоб організувати форматований вивід). Після чого дана функція запускає парсинг усіх рядків. 8. Number\_of\_columns(List, Num) – приймає список і число 0, повертає кількість колонок 9. Parse(RowsList, ColNum, ColumnLength, Pattern) – функція, яка парсить усі дані. Вона бере голову списка( рядок даних), ділить рядок згідно патерну(регулярного виразу) на колонки, запускає функцію консольного виводу і запускає сама себе уже з хвостом списку в якості вхідного параметру. 10. ConsoleOutput(Str, ColumnLength, ColNum, ColNumAll, Pattern) – функція, яка виводить дані у консоль, запускається рекурсивно стільки разів, скільки колонок, довжина однієї колонки потрібна для того, щоб рядок поміщався рівно у один рядок в консолі. Остання колонка обрізається так, щоб зайняти увесь вільний простір в рядку. Якщо файл .tsv дані не обрізаються, адже в останній колонці уся інформація |
| Результати | Результати виводу у консоль 4 файлів:  1)  C:\Users\Lenovo\Desktop\Output_mp-assistants.png    2)  C:\Users\Lenovo\Desktop\Output_map_zal_sk19.png  3)  C:\Users\Lenovo\Desktop\Output_mp-posts_full.png  4)  C:\Users\Lenovo\Desktop\Output_plenary_register_mps-skl9.png |