

DT Day4Taskmain

main

Assignment2 – main.scala [Assignment2]Day4Task – main.scala [Day4Task]Day4 – caseTypes.scala [Day4]

Project

Day4Task ~/Documents/Scala Training/Day4Task

.bsp

.idea

project [Day4Task-build] sources root

src

main

scala

data.csv

file.csv

main

test

target

.gitignore

build.sbt

External Libraries

Scratches and Consoles

main.scalaSource.csvdata.csvfile.csv

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

```
def main(): Unit = {  
  
    val filePath = "/Users/abhishekanand/Documents/Scala Training/Day4Task/src/main/scala/data.csv"  
  
    val bufferedSource = Source.fromFile(filePath)  
    var listOfEmployees: List[Employee] = List()  
    var index: Int = 0;  
    for (line <- bufferedSource.getLines ) {  
        if (index > 0) {  
            val Array(serialNo, name, city, salary, department) = line.split(",").map(_.trim)  
            val emp: Employee = Employee(serialNo.toInt, name, city, salary.toDouble, department);  
            listOfEmployees = listOfEmployees :+ emp;  
        }  
        index += 1;  
    }  
  
    val salesEmpWithSalaryMoreThan5000 = listOfEmployees.filter(emp => emp.salary >= 50000 && emp.depat.equalsIgnoreCase("sales"))  
    println("\nEmployees with Department sales and salary more than 5000")  
    salesEmpWithSalaryMoreThan5000.foreach(println)  
    val employeeNamesInCapsStartingWithJ = listOfEmployees.map(emp => emp.name.toUpperCase).filter(item => item.startsWith("J"));  
    println("\nEmployee names in capital letters starting with J")  
    employeeNamesInCapsStartingWithJ.foreach(println)  
  
    val groupedEmployees = listOfEmployees.groupBy(_.depat)  
  
    println("\nDepartment based stats of the employees")  
    val departmentStats = groupedEmployees.view.mapValues { employees =>  
        val totalEmployees = employees.length  
        val totalSalary = employees.map(_.salary).sum  
        val averageSalary = totalSalary / employees.length.toDouble  
        (totalEmployees, totalSalary, averageSalary)  
    }  
  
    departmentStats.foreach { case (department, (totalEmployees, totalSalary, averageSalary)) =>  
        println(s"Department: $department, Total Employees: $totalEmployees, Total Salary: $totalSalary, Average Salary: $averageSalary")  
    }  
    bufferedSource.close()  
}  
  
case class Employee(sno: Int, name: String, city: String, salary: Double, depat: String) new *
```

main()

Runmain

Day4Task > src > main > scala > main.scala5:3 LF UTF-8 [T] 2 spaces

