

[Java1 on Bootcamp - latest](#) »

Assignment No.1 for Java

Note

Add the link about Page Break on October 3rd.

Develop a program which issues a report specified in the layout image (`Layout01.txt`) with table data below (`tb1.1`) read from Database.

- [Layout01.txt](#)

Caution

You need to use a monospace font when you check *Layout01.txt*. If not, this layout may break on your editor.

Note

The layout image `Layout01.txt` shows the title to be printed and the number of digits each item should have. `x` and numbers like *1,234,567,890* are the place where the data will be printed. And all the codes, names, and other numbers are mere examples and are NOT of the actual values or the actual calculation. With matters not mentioned in these specifications, deal by yourself.

Table Layout of the Inputs (`tb1.1`)

sikyu_tbl		Type in Java	Item Name in English
Item Name	Field Name		
社員番号	sya_bg	String	Employee Number
社員氏名	sya_nm	String	Employee Name
事業所コード	jig_cd	String	Office Code
部コード	bu_cd	String	Department Code
課コード	ka_cd	String	Section Code
基本給	kihon_kyu	BigDecimal	Base Pay
家族手当	kazoku_kyu	BigDecimal	Family Allowance
役職手当	yakusyoku_kyu	BigDecimal	Position Allowance
通常残業時間	tsujo_zan_jikan	BigDecimal	Normal Overtime
深夜残業時間	sinya_zan_jikan	BigDecimal	Late-night Overtime

Specifications

- Print layout MUST follow the example presented in `Layout01.txt`. Digit count of Items and the blank width between items MUST be coordinated as presented in `Layout01.txt`. The line after header MUST be a blank line.



Java1 on Bootcamp - latest

- Line 00 could possibly be a blank line. And also, if the line 00 is NOT a blank line, line 01 could possibly be a blank line.
- Do page-break and print a header when page changes. And count up the page-counter and print it at the same time. This counter starts from 1.
- Output should be printed in ascending order by *Office Code*, *Department Code*, *Section Code* and *Employee Number*.
- Print the subtotals of items below each time *Office Code*, *Department Code* or *Section Code* changes. The Subtotals: *Base Pay*, *Family Allowance*, *Position Allowance*, *Overtime Allowance* and *Total Pay*.
- Must be case-sensitive for codes.
- There is no change the layout depends on the read data of lines of details or totals.
- The lines between totals (the grand and sub) and details or the lines between subtotals must be a blank line.
- Blank lines MUST NOT be printed 2 lines consecutively in a row.
- Print read data without data processing except comma of a numerical value, addition of decimal point and total calculation.
- Blanks MUST be a single-byte character, except those included in the data.
- Consider there is no invalid data, such as a record with no "Office Code". In also, consider there is no possibility of overflow.
- Maximum digit count per line should be 198.
- Develop a program assuming the client as with over 10,000 employees.
- Total Pay* = *Base Pay* + *Family Allowance* + *Position Allowance* + *Overtime Allowance*
- Calculation Formula for *Overtime Allowance* (On Java programming, should use BigDecimal type for all calculation)

```
INT(INT((Base Pay + Family Allowance + Position Allowance) / 160) * Normal Overtime * 1.2) -  
INT(INT((Base Pay + Family Allowance + Position Allowance) / 160) * Late-night Overtime * 1.
```

A + B gives Overtime Allowance

Ex. In the case of...:

- *Base Pay*: 200,000
- *Family Allowance*: 20,000
- *Position Allowance*: 5,000
- *Normal Overtime*: 43h
- *Late-night Overtime*: 5.5h

gives ¥84,148 of Overtime Allowance.

Program Name

Start program with an `main` method of the class `jp.co.worksap.batch.BatchMain` (Same sample program).

Page Break

See also

[Wikipedia](#)

Control Break

See also

[Wikipedia](#)



事業所	部	課	社員番号	氏名	給与
1	1	1	1	社員0001	100000
1	1	1	2	社員0002	200000
1	1	2	10	社員0010	300000
1	1	2	4	社員0004	400000
1	2	2	3	社員0003	500000
1	2	2	6	社員0006	600000
2	2	1	5	社員0005	700000

The result of printing is the following if there is the above data.

Caution

This example is just for Control Break. This does not follow `Layout01.txt`, and this layout might be broken depends on your font.

```

事業所 部 課 社員番号 氏名 給与
  1   1  1  1      社員0001  100,000
  1   1  1  2      社員0002  200,000
           課合計      300,000
  1   1  2 10      社員0010  300,000
  1   1  2  4      社員0004  400,000
           課合計      700,000
           部合計     1,000,000
  1   2  2  3      社員0003  500,000
  1   2  2  6      社員0006  600,000
           課合計     1,100,000
           部合計     1,100,000
           事業所合計  2,100,000
  2   2  1  5      社員0005  700,000
           課合計      700,000
           部合計      700,000
           事業所合計  700,000
           全社合計   2,800,000

```

Additional Information

Data Formatting

You can use useful methods for data formatting by importing this class.

```
jp.co.worksap.printer.common.ReportUtilPlus
```

limitedString(String str, int limit)

This method can use to limit the length of characters. In also, this method can fill in half-width whitespace if the limit value has over the length of characters.

```

limitedString("12345", 2) -> "12"
limitedString("12345", 7) -> "12345  "

```

formatNumWithComma(BigDecimal num, int limit)



Java1 on Bootcamp - latest

```
formatNumWithComma("12345", 2) -> "45"  
formatNumWithComma("12345", 7) -> " 12,345"
```

formatHour(BigDecimal num, int limit)

This method can use to limit the length of characters. In also, this method can fill in half-width whitespace if the limit value has over the length of characters.

```
formatHour(1234.5, 2) -> ".5"  
formatHour(1234.5, 7) -> " 1234.5"
```

[Java1 on Bootcamp - latest](#) »

© Copyright 2016, Works Applications Co., Ltd.. Created using [Sphinx](#) 1.7.4.