

Arquivos

1. Crie uma classe `ArquivoIP` que leia um arquivo texto ("`entrada.txt`") contendo uma lista de endereços IP e gere um outro arquivo ("`ips.txt`"), contendo um relatório dos endereços IP válidos e inválidos. Lembrando que um endereço IP possui o formato `x.y.z.w`, onde `x`, `y`, `z` e `w` são números no intervalo `[0, 255]`. O seu programa deve prover tratamento de exceções para falhas na abertura/leitura e fechamento do arquivo (`std::ios::exceptions`).

Exemplo do arquivo de entrada:

```
200.135.80.9
192.168.1.1
8.35.67.74
257.32.4.5
85.345.1.2
1.2.3.4
9.8.234.5
192.168.0.256
```

Exemplo do arquivo gerado como saída:

```
[Endereços válidos:]
```

```
200.135.80.9
192.168.1.1
8.35.67.74
1.2.3.4
```

```
[Endereços inválidos:]
```

```
257.32.4.5
85.345.1.2
9.8.234.5
192.168.0.256
```

2. Uma empresa está tendo problema com o controle de despesas de viagens de seus funcionários. Para tentar resolver tal situação, ela contratou você para desenvolver um programa que indique: o total de gastos com despesas de viagens de funcionários durante o último mês. Além disso, o funcionário com maior gasto, e o gasto médio e proporcional de cada funcionário. Para isso ela irá te fornecer o arquivo abaixo, chamado "`funcionarios.txt`":

Exemplo do arquivo de entrada:

```
julio R$3003,72
miguel R$1245,98
lidiana R$1234,64
maria R$912,57
caio R$9874,58
paula R$789,45
```

Este arquivo contém informações relativas aos gastos com viagens do último mês, organizado da seguinte forma: cada linha contém informações de um funcionário, isto é, seu identificador (nome) e seu gasto mensal separado por espaço. A partir deste arquivo, você deve criar um programa (classe `Funcionario`) que gere um relatório, chamado `"relatorio.txt"`, no seguinte formato:

Exemplo do arquivo gerado como saída:

```
Nro. Funcionario Gasto %
1 julio R$3003,72 18%
2 miguel R$1245,98 7%
3 lidiana R$1234,64 7%
4 maria R$912,57 5%
5 caio R$9874,58 58%
6 paula R$789,45 5%
Gasto mensal total: R$17060,94
Gasto médio por funcionário: R$2843,49
```

O seu programa deve prover tratamento de exceções para falhas na abertura/leitura e fechamento do arquivo.

Considerações Gerais!

- Exercício individual.
- Entrega: conforme agendado no PVANET Moodle;
- Conforme estrutura abaixo apresentada crie um projeto para resolução de cada exercício (ex.: `pratica8.exercicio1.zip`, `pratica8.exercicio2.zip`, etc). Cada projeto deve conter os arquivos `.h`, `.cpp`, e `main.cpp` criados para resolução do exercício. Envie, através do PVANet Moodle, uma pasta compactada (`.rar` ou `.zip`) contendo todos os projetos (também compactados). A pasta compactada deve conter informações do aluno (ex.: `julio_reis-pratica8.zip`).

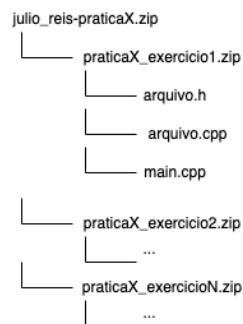


Figura 1: Estrutura de diretórios.

- O seu `main.cpp` deve conter, minimamente, instruções para criação (instanciação de objetos) e chamadas das funções implementadas (TODAS!!!). Para teste, você pode usar os exemplos fornecidos.