Engenharia de Telecomunicações
Programação Orientada a Objetos

Prof. Sergio Prolo
sergio.prolo@ifsc.edu.br

Lista 1: Ambiente Java, git e Gradle

31/03/2025

Informações iniciais

- No repositório do GitHub, há uma pasta separada para cada exercício. Dentro de cada, você deve criar um projeto Java utilizando o Gradle e configurar seus arquivos para que a tarefa `run` seja executada de modo 'silencioso'.
- Durante o desenvolvimento, é importante realizar *commits* frequentes. Por exemplo: um *commit* após a criação do projeto, outro após implementar uma funcionalidade, e assim por diante. Ao concluir um exercício, não esqueça de fazer um *push* para atualizar o repositório no GitHub.
- Na raiz do projeto, edite o arquivo `Readme.md` e inclua o seu nome. Esse arquivo deve ser atualizado a cada novo exercício resolvido, com instruções claras de como executar a aplicação Java correspondente.
- Não envie para o repositório arquivos binários ou gerados automaticamente, como arquivos `.class`, arquivos da IDE ou outros arquivos temporários. Use o `.gitignore` para evitar isso (veja os slides sobre git).
- Todas as aplicações desenvolvidas serão avaliadas com base em critérios de legibilidade, clareza e organização do código [15 pontos].

Atenção

Plágio não é tolerado

- Você deve ser o único(a) responsável por fazer a entrega para essa atividade. Todo o código ou texto deverá ser produzido exclusivamente por você, exceto trechos de códigos que possam ter sido fornecidos como parte do enunciado.
- Você pode discutir com outros estudantes com o intuito de esclarecer pontos, porém você não deverá copiar trechos de códigos, textos ou soluções de qualquer fonte (e.g. colegas da mesma turma ou de turmas anteriores, repositórios de códigos na Internet ou soluções providas por serviços como Copilot e ChatGPT).

Aplicação 1 [15 pontos]

Arte ASCII

Desenvolva um aplicativo Java que imprima no terminal uma de três formas: um triângulo retângulo, um losango ou um retângulo vazado. O tipo de figura a ser impressa e suas dimensões são escolhidos pelo usuário através de argumentos de linha de comando. Em especial, a dimensão do losango deve ser ímpar. Caso o usuário forneça argumentos inválidos, o programa deve informar a forma correta de uso.

Exemplos de execução:

```
gradle run --args "triangulo 5"

    *
   **
  ***
 ****
*****
```

```
gradle run --args "losango 5"

  *
 ***
*****
 ***
  *
```

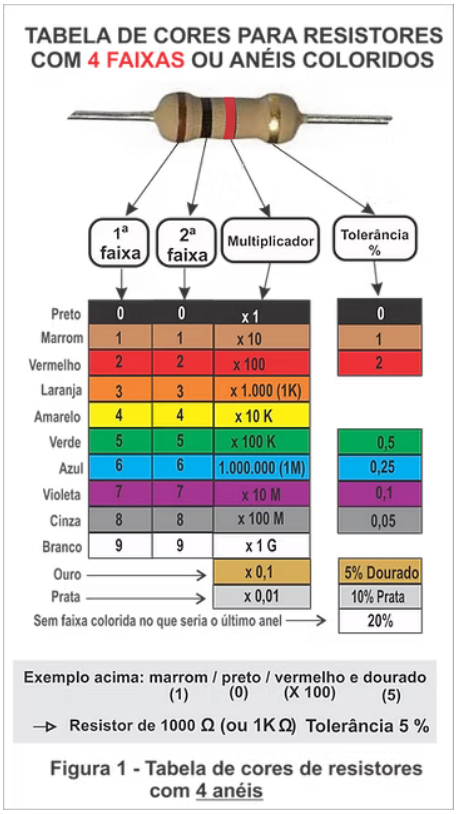
```
gradle run --args "retangulo 8 5"

********
*      *
*      *
*      *
********
```

Aplicação 2 [20 pontos]

Decodificador de resistência

Resistor é um componente eletrônico que transforma energia elétrica em energia térmica, limitando a corrente elétrica em um circuito. A resistência de um resistor, medida em Ohms (Ω), é representada por meio de faixas coloridas impressas em seu corpo. As duas primeiras faixas indicam os dois dígitos iniciais do valor, a terceira faixa indica um multiplicador, e a quarta faixa (opcional) indica a tolerância da resistência. A Figura 1 apresenta os valores para cada cor, bem como um exemplo de cálculo.

Figura 1: Código de cores do resistor com 4 faixas



Fonte: Tabela de Cores RF | SABER 360

Desenvolva um aplicativo em Java que permita ao usuário fornecer, como argumentos de linha de comando, o nome das cores das faixas de um resistor e imprima na tela o valor da resistência correspondente. Caso o usuário forneça argumentos inválidos, o programa deve informar a forma correta de uso.

Exemplo de execução:

```
gradle run --args "amarelo branco verde azul"

Resistência: 4,9 M Ohms (+- 0,25%)
```

Aplicação 3 [25 pontos]

Tabuleiro de Batalha Naval

Batalha naval é um jogo de tabuleiro em que dois jogadores precisam afundar a frota do adversário. Cada jogador possui um tabuleiro com dimensões 10×10. No início da partida, os jogadores posicionam seus navios na vertical ou horizontal, não sendo permitido a sobreposição de navios. A Tabela 1 apresenta uma lista dos navios em jogo.

Desenvolva um aplicativo em Java que gere um tabuleiro, posicione de forma aleatória a frota de um jogador e imprima na tela o tabuleiro gerado. Para casas que contém um navio posicionado, imprima o símbolo do navio. Para as casas que não contém navios, imprima o caractere ponto. Separe as casas da mesma linha com espaços.

Tabela 1: Lista de navios do jogo Batalha Naval

Navio	**Tamanho** (casas)	**Símbolo**
Porta-aviões	5	P
Encouraçado	4	E
Cruzador	3	C
Submarino	3	S
Contratorpedeiro	2	N

Exemplo de execução:

```
gradle run

C C C . . . . . . .
. . . . . . . . . .
. . . . . S S S . .
. . . . . . . . . .
. P P P P P . . . .
. . . . . . . E . .
. . . . . . . E . .
. . . . . . . E . .
. . . . . . . E . .
. . . N N . . . . .
```

Aplicação 4 [25 pontos]

Validação de tabuleiro

Desenvolva um programa em Java que receba, como redirecionamento de entrada, um arquivo texto contendo um tabuleiro gerado no exercício 1. O programa deve verificar se o tabuleiro é válido (ou seja, segue as regras do jogo). Caso o tabuleiro seja válido, o programa deve imprimir "`Tabuleiro válido`".

Exemplos de execução:

```
gradle run < tabuleiro.txt

Tabuleiro válido
```

Caso o tabuleiro seja inválido, o programa deve informar o motivo da não conformidade. Um tabuleiro é considerado inválido quando:

- não tem a dimensão correta de 10×10 casas [3 pontos];
- inclui navios desconhecidos [4 pontos];
- não inclui um navio de cada tipo [5 pontos];
- inclui múltiplos navios do mesmo tipo [5 pontos]; ou
- inclui navios que não estão na horizontal ou vertical [8 pontos].

Documento licenciado sob Creative Commons “Atribuição 4.0 Internacional”.