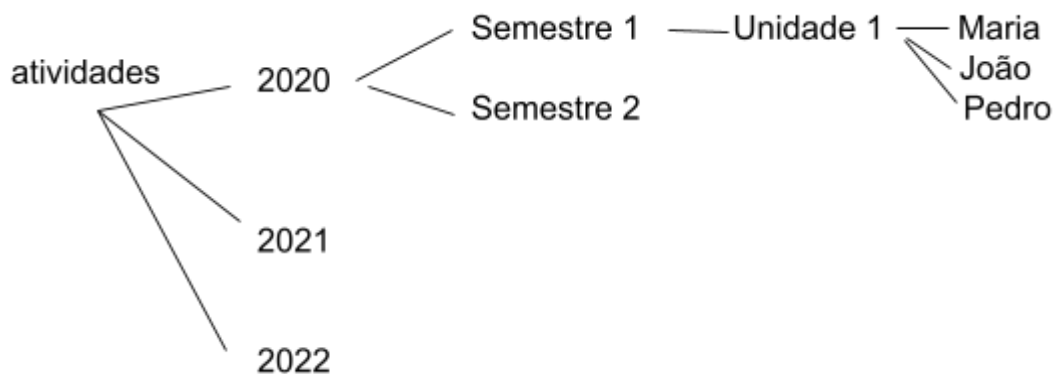


Lista de Exercícios 1 — Unidade 3

Atividade não avaliativa!

1. Um professor precisa lançar as notas dos alunos, porém, cada nota está contida num arquivo nomeado pela matrícula seguida do nome do aluno, por exemplo, 2020123_joão. As notas devem ser lançadas em ordem alfabética, então, para diminuir o trabalho do professor, escreva um programa que renomeia todos os arquivos de uma pasta que estejam no formato matrícula_nome para nome_matrícula.
2. As atividades dos alunos estão organizadas numa estrutura de diretórios do seguinte modo: todos os arquivos se encontram dentro do diretório atividades que possui uma pasta para cada ano. Por sua vez, as atividades de cada ano estão organizadas em semestres e os semestres divididos em unidades. O diagrama abaixo ilustra esta estrutura. Crie uma aplicação recursiva que receba o nome de um aluno e liste todas as suas atividades indicando o ano, semestre e unidade.



3. Crie uma aplicação que receba um inteiro N e recupere a linha N de um arquivo de texto.
4. Crie uma aplicação que receba o nome de um arquivo de texto e então separe as palavras de cada linha e as armazene num `std::vector`.
5. Crie uma classe em que os objetos representam contadores, isto é, ele inicia com zero e possui métodos para incrementar, reiniciar e verificar o estado atual do contador.
6. Crie uma classe em que os objetos são responsáveis pela geração de identificadores. Deve ter um método para gerar um novo identificador, isto é, um valor inteiro e ele é sempre diferente em todo o sistema, ou seja, nunca são gerados identificadores iguais mesmo sendo invocado o método em objetos diferentes.
7. Crie uma classe em que os objetos geram valores aleatórios. Deve ter um método para gerar um novo valor aleatório dado um intervalo inteiro, isto é,

dado o intervalo $[A,B]$ o método deve gerar um valor aleatório dentro desse intervalo. Além disso, ela deve ter um método para escolher aleatoriamente um elemento de um `std::vector` e outro método para embaralhar ou misturar os elementos de um `std::vector` sem modificar seus elementos, somente as posições.

8. Crie uma classe em que os objetos representam uma turma de alunos contendo objetos que representam o professor, os alunos e as notas dos mesmo. Quais informações são pertinentes para representar o professor e os alunos?
9. Crie uma classe para representar imagens em 8 tons de cinza, isto é, as imagens são matrizes de dimensão predeterminada e os tons são representados pelos valores de 0 a 7. Deve haver um método para mudar o tom de um determinado pixel (posição da matriz) como também para desenhar uma linha de um pixel a outro; um método para mudar a cor de todos os pixels vizinhos para uma nova cor, similar a ação preenche com cor do paint; e um método para obter um recorte da imagem que deve retornar uma nova imagem correspondendo a uma região da imagem.
10. Crie uma classe em que os objetos representam um mapa bidimensional, isto é, uma matriz em que cada posição permite a passagem ou não. Assuma que um personagem pode se deslocar nesse mapa nas direções vertical e horizontal desde que a posição de destino esteja livre. O objeto deve ter um método para permitir ou não a passagem; outro para determinar a posição inicial do personagem; e outro para determinar se existe um caminho da posição atual do personagem até outra especificada. Desafio: crie um método para determinar o caminho mais curto da posição atual do personagem até outra especificada.
11. Implemente a busca binária de forma iterativa.
12. Implemente a busca binária de forma recursiva em memória complexidade de memória constante $O(1)$.
13. Utilize o algoritmo de busca binária para criar um método de inserção de elementos de forma ordenada em um `std::vector`.
14. Utilize uma estrutura de dados de pilha para verificar se os parênteses numa expressão matemática de uma linguagem de programação está correta, isto é, cada abre parênteses possui o seu respectivo fecha parêntese vice-versa.
15. Utilize uma estrutura de dados de pilha para inverter os caracteres de uma string.
16. Implemente as duas questões anteriores de forma recursiva.
17. Dada duas listas ligadas ordenadas, reorganize os nós de modo a formar uma única lista com todos os elementos sem utilizar alocação de memória e que também esteja ordenada.