Trabalho (20 pontos)
Data de entrega: 16/01/2024
A entrega deve ser feita pelo SIGAA até as 23:59 do dia 16/01/2024
Este trabalho será avaliado tanto na disciplina teórica quanto na prática

Trabalho 1 - Sistema de Gerenciamento de Sobreviventes e Recursos

Após um evento catastrófico que devastou o mundo, um grupo de sobreviventes se organiza em um refúgio subterrâneo, chamado Vault 101. A sobrevivência da comunidade depende de uma gestão eficiente dos recursos disponíveis, das habilidades dos sobreviventes e das missões realizadas para explorar a superfície em busca de novos suprimentos.

Você foi escolhido para desenvolver um sistema que permita gerenciar os sobreviventes, suas habilidades, os recursos disponíveis e as missões realizadas. O sistema será essencial para manter o Vault 101 funcionando de forma eficiente e para garantir a sobrevivência da comunidade.

Lista de classes e Atributos Detalhados do sistema

Classe Pessoa

1. **nome**: Nome da pessoa (Ex.: "John Doe", "Sarah Connor").
2. **idade**: Idade da pessoa (número inteiro maior que 0).
3. **identificador**: ID único, como uma string alfanumérica (Ex.: "S12345", "SURV001").

Classe Sobrevivente (subclasse de pessoa)

1. **habilidades** (máximo de 3 por sobrevivente): Lista de habilidades que o sobrevivente pode possuir. Possíveis valores:
    - "Engenharia" (habilidade de construir e reparar máquinas).
    - "Medicina" (habilidade de tratar ferimentos e doenças).
    - "Combate" (habilidade de lutar com armas ou corpo a corpo).
    - "Exploração" (habilidade de navegar e mapear áreas desconhecidas).
    - "Culinária" (habilidade de preparar comida com recursos escassos).
    - "Diplomacia" (habilidade de negociar com outros grupos).
    - "Hackeamento" (habilidade de invadir sistemas de segurança).
    - "Furtividade" (habilidade de evitar detecção durante missões).

O jogador deve selecionar no máximo 3 habilidades por sobrevivente.

2. **status**: Situação atual do sobrevivente. Possíveis valores:
    - "Ativo" (em plenas condições para atuar em missões).
    - "Doente" (temporariamente impossibilitado de participar de missões).
    - "Ferido" (parcialmente ativo, mas com penalidades).

- "Mortos" (não pode mais ser usado no sistema).

Classe Recurso

1. **nome**: Tipo de recurso. Possíveis valores:
   - "Água" (fundamental para hidratação).
   - "Comida" (necessária para manter os sobreviventes ativos).
   - "Munição" (usada para armas de defesa e combate).
   - "Remédios" (usados para curar doenças e ferimentos).
   - "Partes Mecânicas" (usadas para construir ou consertar itens no Vault).
2. **quantidade**: Número inteiro maior ou igual a 0, representando o total disponível.

Classe Missao

1. **nome**: Nome da missão (Ex.: "Explorar a Zona Vermelha", "Buscar suprimentos na cidade").
2. **objetivo**: Propósito da missão (Ex.: "Coletar suprimentos", "Resgatar sobreviventes", "Procurar informações").
3. **local**: Nome do local onde a missão será realizada (Ex.: "Cidade Arruinada", "Refinaria Abandonada", "Estação de Trem").
4. **sobreviventes**: Lista de sobreviventes designados para a missão (máximo de 5 por missão). Essa lista deve armazenar apenas o identificador do sobrevivente. Entretanto, deve existir um método que imprime os sobreviventes alocados para a missão.
5. **recursosColetados**: Lista de recursos encontrados durante a missão.
6. **status**: Situação da missão. Possíveis valores:
   - "Em andamento" (ainda não concluída).
   - "Sucesso" (concluída com êxito).
   - "Fracasso" (falhou por algum motivo, como falta de recursos ou mortes).

Classe Vault

1. **sobreviventes**: Lista de sobreviventes registrados no Vault.
2. **recursos**: Lista de recursos armazenados no Vault.
3. **missoes**: Lista de missões realizadas ou em andamento

Requisitos

1. Faça um **diagrama de classes** que represente o sistema descrito acima.
2. O sistema deve utilizar listas alocadas dinamicamente para armazenar sobreviventes, recursos e missões.
3. Implemente todos os getters e os setters.
4. Você pode escolher usar Java ou C++ para implementar o exercício.
5. Crie um programa principal (main) que inclua o seguinte menu:
    - **1:** Cadastrar sobrevivente
    - **2:** Adicionar habilidade a um sobrevivente
    - **3:** Remover habilidade de um sobrevivente
    - **4:** Adicionar recurso ao Vault
    - **5:** Consumir recurso
    - **6:** Registrar missão
    - **7:** Adicionar sobrevivente a uma missão
    - **8:** Exibir sobreviventes do Vault
    - **9:** Exibir recursos do Vault
    - **10:** Exibir missões realizadas
    - **11:** Exibir sobreviventes cadastrados em uma missão
    - **11:** Sair

Documentação

A entrega deve incluir uma documentação do programa contendo:

1. **Introdução:** Explique o objetivo do trabalho e como ele atende às necessidades descritas no cenário.
2. **Decisões de Implementação:** Detalhe as escolhas feitas na estruturação do código e justificativas.
3. **Casos de Teste:** Descreva pelo menos 5 casos de uso e como eles foram testados no programa.
4. **Resultados dos Testes Realizados:** Inclua evidências (prints ou logs) dos testes realizados.
5. **Análise dos Resultados:** Avalie a funcionalidade e desempenho do sistema.

Avaliação

- **Implementação do Diagrama de Classes e Código (8 pontos).**
- **Funcionamento do Menu e Métodos do Sistema (6 pontos).**
- **Documentação do Programa (4 pontos).**
- **Clareza e Organização do Código (2 pontos).**