Senaizinho

Descrição:

Criar um projeto de aplicação console chamado SENAIzinho no visual studio code para implementar um sistema de controle de usuários de uma escola de informática, o prédio tem 10 salas de aulas no total e cada sala tem capacidade para 10 alunos.

A aplicação deverá implementar as seguintes classes:

|  |
| --- |
| **Aluno** |
| public string Nome  public DateTime DataNascimento  public string Curso  public int numeroSala |
|  |

A classe aluno armazena as informações pertinentes aos alunos registrados na escola e não implementa nenhum método especifico.

|  |
| --- |
| **Sala** |
| public int capacidadeAtual  public int capacidadeTotal  public int numeroSala  public string[] Alunos |
| public Sala(int, int);  public string AlocarAluno (string)  public string RemoverAluno (string)  public String MostrarAlunos () |

A classe Sala, representa uma sala de aula da escola, ela deve implementar seguintes métodos:

1. O método construtor deve receber dois inteiros um para o número da sala e outro com a capacidade total da sala.
2. O método **AlocarAluno** deverá receber o nome do aluno a ser registrado na sala, registar esse nome no vetor de alunos da sala e retornar uma mensagem dizendo se o aluno foi cadastrado com sucesso ou caso a sala já esteja lotada, uma mensagem de erro dizendo que não foi possível registrar o aluno na sala.
3. O método **RemoverAluno**, deve receber o nome do aluno a ser retirado da sala, remover o nome do aluno do vetor de alunos na sala e retornando uma mensagem dizendo que o aluno foi removido com sucesso da sala, ou uma mensagem de erro, caso o aluno não exista na sala ou se a sala estiver vazia.
4. O método **MostrarAluno**, deve retornar uma string com o nome de todos os alunos registrados na sala de aula.

|  |
| --- |
| **Program** |
| int limiteAlunos  int limiteSalas  Aluno[] alunos  Sala[] salas  int alunosCadastrados  int salasCadastradas  bool querSair |
| public void CadastrarAluno()  public void CadastrarSala()  public void AlocarAluno()  public void RemoverAluno()  public void VerificarSalas()  public void VerificarAlunos() |

1. A aplicação principal deverá manter uma lista com todos os alunos matriculados na escola e outra com todas as salas existentes.
2. O método **CadastrarAluno**, deve verificar se a escola comporta a matrícula de um novo aluno e caso isso seja possível, pedir para o usuário o nome e a idade do aluno e registrar esse aluno no vetor de alunos da escola. A escola pode ter no máximo 100 alunos.
3. O método **CadastrarSala**, deve verificar se a escola tem espaço para mais uma sala e se isso for possível solicitar ao usuário que entre com o número da sala a ser aberta, inicializando a sala com a sua capacidade máxima, que é de 10 alunos. A escola pode ter o máximo de 10 salas.
4. O método **AlocarAluno**, deve pedir para o usuário o nome do aluno e a sala onde esse aluno será registrado, verificar se o aluno e a sala existem na escola e se possível, atualizar o aluno e a sala com as informações necessárias.
5. O método **RemoverAluno**, deve pedir o nome do aluno e remover esse aluno da sala onde ele está cadastrado.
6. O método **VerificarSalas**, deve imprimir um relatório de todas as salas da escola com as seguintes informações: Número da Sala, Quantidade de alunos registrados na sala e a lista dos nomes de todos os alunos da sala.
7. O método **VerificarAlunos**, deve imprimir um relatório de todos os alunos matriculados na escola, com os seguintes dados: Nome do aluno e o número da sala que ele está matriculado.
8. A aplicação deverá implementar um menu de ações que permita ao usuário realizar as operações acima.

1 - Cadastrar Aluno

2 - Cadastrar Sala

3 - Alocar Aluno

4 - Remover Aluno

5 - Verificar Salas

6 - Verificar Alunos

0 – Sair