

Desafio 2-N2.2

Crie um programa completo de Matrícula de alunos !

```
import java.text.DecimalFormat

// Enum para o status do aluno
enum class Status {
    APROVADO, REPROVADO
}

// Classe aluno com ID único, nome, notas, média e status
data class Aluno(val id: Int, val nome: String, var notas: MutableList<Float> =
mutableListOf()) {
    fun calcularMedia(): Float {
        return if (notas.isNotEmpty()) notas.average().toFloat() else 0.0f
    }

    fun obterStatus(): Status {
        return if (calcularMedia() >= 7.0f) Status.APROVADO else
Status.REPROVADO
    }
}

// Classe turma com ID único e lista de alunos matriculados
data class Turma(val id: Int, val nome: String, var vagas: Int, val alunos:
MutableList<Aluno> = mutableListOf()) {

    // Adiciona um aluno na turma, verificando se há vagas e se o aluno já está
matriculado
    fun adicionarAluno(aluno: Aluno) {
        if (alunos.contains(aluno)) {
            throw IllegalArgumentException("Aluno já matriculado nesta turma.")
        }
        if (vagas <= 0) {
            throw IllegalArgumentException("Turma lotada.")
        }
        alunos.add(aluno)
        vagas--
        println("Aluno ${aluno.nome} matriculado com sucesso na turma
$nome.")
    }
}
```

```

    }

    // Remove um aluno da turma
    fun removerAluno(alunoid: Int) {
        val aluno = alunos.find { it.id == alunoid }
        if (aluno != null) {
            alunos.remove(aluno)
            vagas++
            println("Aluno ${aluno.nome} removido da turma $nome.")
        } else {
            println("Aluno com ID $alunoid não encontrado nesta turma.")
        }
    }

    // Lista os alunos matriculados na turma com suas médias e status
    fun listarAlunos() {
        if (alunos.isEmpty()) {
            println("Nenhum aluno matriculado nesta turma.")
        } else {
            val df = DecimalFormat("#.0")
            println("Nome\t\tMédia\t\tStatus")
            alunos.forEach {
                println("${it.nome}\t\t${df.format(it.calcularMedia())}\t\t${it.obterStatus()}")
            }
        }
    }

    // Função principal com o menu
    fun main() {
        val turmas = mutableListOf(
            Turma(1, "Matemática", 2),
            Turma(2, "Ciência da Computação", 3)
        )

        val alunos = mutableListOf(
            Aluno(1, "João", mutableListOf(8.5f, 9.0f)),
            Aluno(2, "Ana", mutableListOf(6.0f, 8.5f)),
            Aluno(3, "Lucas", mutableListOf(4.5f, 8.5f)),
            Aluno(4, "Mariana", mutableListOf(3.5f, 9.0f)),
            Aluno(5, "Pedro", mutableListOf(6.0f, 4.5f))
        )
    }

```

```

while (true) {
    println(
        """
        MENU:
        1. Consultar Turmas e Alunos
        2. Matricular Aluno em Turma
        3. Consultar Aluno (Média e Status)
        4. Excluir Matrícula de Aluno
        5. Ajustar Notas de Aluno
        6. Sair
        """).trimIndent()
    )
    when (readLine()?.toIntOrNull()) {
        1 -> consultarTurmas(turmas)
        2 -> matricularAluno(turmas, alunos)
        3 -> consultarAluno(turmas)
        4 -> excluirMatricula(turmas)
        5 -> ajustarNotas(alunos)
        6 -> break
        else -> println("Opção inválida.")
    }
}

// Função para consultar todas as turmas e seus alunos
fun consultarTurmas(turmas: List<Turma>) {
    if (turmas.isEmpty()) {
        println("Nenhuma turma cadastrada.")
    } else {
        turmas.forEach {
            println("Turma ${it.id} - Nome: ${it.nome} - Vagas: ${it.vagas}")
            it.listarAlunos()
        }
    }
}

// Função para matricular um aluno em uma turma
fun matricularAluno(turmas: MutableList<Turma>, alunos:
MutableList<Aluno>) {
    println("Digite o ID do aluno:")
    val alunold = readLine()?.toIntOrNull()
    val aluno = alunos.find { it.id == alunold }
    if (aluno == null) {
        println("Aluno não encontrado.")
    }
}

```

```

        return
    }

    println("Turmas disponíveis:")
    turmas.forEach { println("Turma ${it.id} - Nome: ${it.nome} - Vagas:
${it.vagas}") }

    println("Digite o ID da turma para matrícula:")
    val turmaid = readLine()?.toIntOrNull()
    val turma = turmas.find { it.id == turmaid }
    if (turma == null) {
        println("Turma não encontrada.")
        return
    }

    try {
        turma.adicionarAluno(aluno)
    } catch (e: IllegalArgumentException) {
        println(e.message)
    }
}

// Função para consultar a média e status de um aluno específico
fun consultarAluno(turmas: List<Turma>) {
    println("Digite o ID do aluno:")
    val alunoid = readLine()?.toIntOrNull()

    val alunoEncontrado = turmas.flatMap { it.alunos }.find { it.id == alunoid }
    if (alunoEncontrado != null) {
        turmas.forEach { turma ->
            if (alunoEncontrado in turma.alunos) {
                println("Aluno: ${alunoEncontrado.nome}, Turma: ${turma.nome}")
                println("Média: ${alunoEncontrado.calcularMedia()}, Status:
${alunoEncontrado.obterStatus()}")
            }
        }
    } else {
        println("Aluno não encontrado.")
    }
}

// Função para excluir a matrícula de um aluno
fun excluirMatricula(turmas: MutableList<Turma>) {
    println("Digite o ID do aluno para excluir a matrícula:")

```

```

val alunold = readLine()?.toIntOrNull()

val turma = turmas.find { it.alunos.any { aluno -> aluno.id == alunold } }
if (turma != null) {
    turma.removerAluno(alunold!!)
} else {
    println("Aluno não encontrado em nenhuma turma.")
}
}

// Função para ajustar as notas de um aluno
fun ajustarNotas(alunos: MutableList<Aluno>) {
    println("Digite o ID do aluno:")
    val alunold = readLine()?.toIntOrNull()
    val aluno = alunos.find { it.id == alunold }

    if (aluno == null) {
        println("Aluno não encontrado.")
        return
    }

    println("Digite as novas notas do aluno (separadas por vírgula):")
    val notasInput = readLine()?.split(",")?.mapNotNull { it.toFloatOrNull() }

    if (notasInput != null) {
        aluno.notas.clear()
        aluno.notas.addAll(notasInput)
        println("Notas ajustadas com sucesso para o aluno ${aluno.nome}.")
    } else {
        println("Entrada inválida.")
    }
}
}

```

Saída:

MENU:

1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair

1

Turma 1 - Nome: Matemática - Vagas: 2

Nenhum aluno matriculado nesta turma.

Turma 2 - Nome: Ciência da Computação - Vagas: 3

Nenhum aluno matriculado nesta turma.

MENU:

1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair

2

Digite o ID do aluno:

Joao

Aluno não encontrado.

MENU:

1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair

2

Digite o ID do aluno:

2

Turmas disponíveis:

Turma 1 - Nome: Matemática - Vagas: 2

Turma 2 - Nome: Ciência da Computação - Vagas: 3

Digite o ID da turma para matrícula:

2

Aluno Ana matriculado com sucesso na turma Ciência da Computação.

```
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
2
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 2
Turma 2 - Nome: Ciência da Computação - Vagas: 3
Digite o ID da turma para matrícula:
2
Aluno Ana matriculado com sucesso na turma Ciência da Computação.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
1
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 2
Turma 2 - Nome: Ciência da Computação - Vagas: 2
Digite o ID da turma para matrícula:
1
Aluno João matriculado com sucesso na turma Matemática.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
3
Digite o ID do aluno:
2
Aluno: Ana. Turma: Ciência da Computação
```

```
2
Digite o ID do aluno:
1
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 2
Turma 2 - Nome: Ciência da Computação - Vagas: 2
Digite o ID da turma para matrícula:
1
Aluno João matriculado com sucesso na turma Matemática.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
3
Digite o ID do aluno:
2
Aluno: Ana, Turma: Ciência da Computação
Média: 7.25, Status: APROVADO
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
3
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 1
Turma 2 - Nome: Ciência da Computação - Vagas: 2
Digite o ID da turma para matrícula:
2
Aluno Lucas matriculado com sucesso na turma Ciência da Computação.
MENU:
1. Consultar Turmas e Alunos
```



```
Turma 2 - Nome: Ciência da Computação - Vagas: 2
Digite o ID da turma para matrícula:
2
Aluno Lucas matriculado com sucesso na turma Ciência da Computação.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
4
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 1
Turma 2 - Nome: Ciência da Computação - Vagas: 1
Digite o ID da turma para matrícula:
1
Aluno Mariana matriculado com sucesso na turma Matemática.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
5
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 0
Turma 2 - Nome: Ciência da Computação - Vagas: 1
Digite o ID da turma para matrícula:
2
Aluno Pedro matriculado com sucesso na turma Ciência da Computação.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
```

```
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
5
Turmas disponíveis:
Turma 1 - Nome: Matemática - Vagas: 0
Turma 2 - Nome: Ciência da Computação - Vagas: 1
Digite o ID da turma para matrícula:
2
Aluno Pedro matriculado com sucesso na turma Ciência da Computação.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
2
Digite o ID do aluno:
7
Aluno não encontrado.
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
1
Turma 1 - Nome: Matemática - Vagas: 0
Nome      Média      Status
João      8.8      APROVADO
Mariana   6.2      REPROVADO
Turma 2 - Nome: Ciência da Computação - Vagas: 0
Nome      Média      Status
Ana       7.2      APROVADO
Lucas     6.5      REPROVADO
Pedro     5.2      REPROVADO
```

```
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
1
Turma 1 - Nome: Matemática - Vagas: 0
Nome      Média      Status
João      8.8        APROVADO
Mariana   6.2        REPROVADO
Turma 2 - Nome: Ciência da Computação - Vagas: 0
Nome      Média      Status
Ana       7.2        APROVADO
Lucas     6.5        REPROVADO
Pedro     5.2        REPROVADO
MENU:
1. Consultar Turmas e Alunos
2. Matricular Aluno em Turma
3. Consultar Aluno (Média e Status)
4. Excluir Matrícula de Aluno
5. Ajustar Notas de Aluno
6. Sair
5
Digite o ID do aluno:
5
Digite as novas notas do aluno (separadas por vírgula):
7.0
Notas ajustadas com sucesso para o aluno Pedro.
```

OBS: Havia 5 alunos que eu deixei “cadastrado” para realizar as matricula.