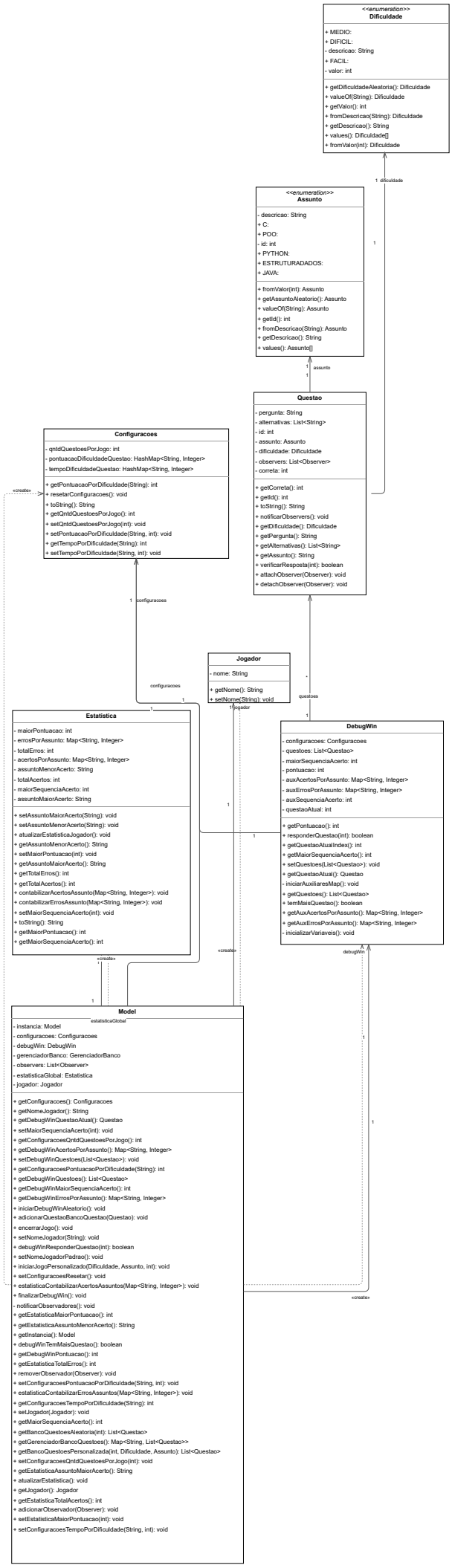
<<enumeration>>
Dificuldade
+ MEDIO:
+ DIFICIL:
- descricao: String
+ FACIL:
- valor: int
+ getDificuldadeAleatoria(): Dificuldade
+ valueOf(String): Dificuldade
+ getValor(): int
+ fromDescricao(String): Dificuldade
+ getDescricao(): String
+ values(): Dificuldade[]
+ fromValor(int): Dificuldade
1 dificuldade
1
<<enumeration>>
Assunto
- descricao: String
+ C:
+ POO:
- id: int
+ PYTHON:
+ ESTRUTURADADOS:
+ JAVA:
+ fromValor(int): Assunto
+ getAssuntoAleatorio(): Assunto
+ valueOf(String): Assunto
+ getId(): int
+ fromDescricao(String): Assunto
+ getDescricao(): String
+ values(): Assunto[]
1 assunto
1
Questao
- pergunta: String
- alternativas: List<String>
- id: int
- assunto: Assunto
- dificuldade: Dificuldade
- observers: List<Observer>
- correta: int
+ getCorreta(): int
+ getId(): int
+ toString(): String
+ notificarObservers(): void
+ getDificuldade(): Dificuldade
+ getPergunta(): String
+ getAlternativas(): List<String>
+ getAssunto(): String
+ verificarResposta(int): boolean
+ attachObserver(Observer): void
+ detachObserver(Observer): void
Configuracoes
- qntdQuestoesPorJogo: int
- pontuacaoDificuldadeQuestao: HashMap<String, Integer>
- tempoDificuldadeQuestao: HashMap<String, Integer>
+ getPontuacaoPorDificuldade(String): int
+ resetarConfiguracoes(): void
+ toString(): String
+ getQntdQuestoesPorJogo(): int
+ setQntdQuestoesPorJogo(int): void
+ setPontuacaoPorDificuldade(String, int): void
+ getTempoPorDificuldade(String): int
+ setTempoPorDificuldade(String, int): void
«create»
1 configuracoes
Jogador
- nome: String
+ getNome(): String
+ setNome(String): void
configuracoes
1
*
questoes
1
Estatistica
- maiorPontuacao: int
- errosPorAssunto: Map<String, Integer>
- totalErros: int
- acertosPorAssunto: Map<String, Integer>
- assuntoMenorAcerto: String
- totalAcertos: int
- maiorSequenciaAcerto: int
- assuntoMaiorAcerto: String
+ setAssuntoMaiorAcerto(String): void
+ setAssuntoMenorAcerto(String): void
+ atualizarEstatisticaJogador(): void
+ getAssuntoMenorAcerto(): String
+ setMaiorPontuacao(int): void
+ getAssuntoMaiorAcerto(): String
+ getTotalErros(): int
+ getTotalAcertos(): int
+ contabilizarAcertosAssunto(Map<String, Integer>): void
+ contabilizarErrosAssunto(Map<String, Integer>): void
+ setMaiorSequenciaAcerto(int): void
+ toString(): String
+ getMaiorPontuacao(): int
+ getMaiorSequenciaAcerto(): int
1
1
1
1
DebugWin
- configuracoes: Configuracoes
- questoes: List<Questao>
- maiorSequenciaAcerto: int
- pontuacao: int
- auxAcertosPorAssunto: Map<String, Integer>
- auxErrosPorAssunto: Map<String, Integer>
- auxSequenciaAcerto: int
- questaoAtual: int
+ getPontuacao(): int
+ responderQuestao(int): boolean
+ getQuestaoAtualIndex(): int
+ getMaiorSequenciaAcerto(): int
+ setQuestoes(List<Questao>): void
+ getQuestaoAtual(): Questao
+ iniciarAuxiliaresMap(): void
+ getQuestoes(): List<Questao>
+ temMaisQuestao(): boolean
+ getAuxAcertosPorAssunto(): Map<String, Integer>
+ getAuxErrosPorAssunto(): Map<String, Integer>
+ inicializarVariaveis(): void
«create»
«create»
debugWin
1
1
1
Model
estatisticaGlobal
- instancia: Model
- configuracoes: Configuracoes
- debugWin: DebugWin
- gerenciadorBanco: GerenciadorBanco
- observers: List<Observer>
- estatisticaGlobal: Estatistica
- jogador: Jogador
+ getConfiguracoes(): Configuracoes
+ getNomeJogador(): String
+ getDebugWinQuestaoAtual(): Questao
+ setMaiorSequenciaAcerto(int): void
+ getConfiguracoesQntdQuestoesPorJogo(): int
+ getDebugWinAcertosPorAssunto(): Map<String, Integer>
+ setDebugWinQuestoes(List<Questao>): void
+ getConfiguracoesPontuacaoPorDificuldade(String): int
+ getDebugWinQuestoes(): List<Questao>
+ getDebugWinMaiorSequenciaAcerto(): int
+ getDebugWinErrosPorAssunto(): Map<String, Integer>
+ iniciarDebugWinAleatorio(): void
+ adicionarQuestaoBancoQuestao(Questao): void
+ encerrarJogo(): void
+ setNomeJogador(String): void
+ debugWinResponderQuestao(int): boolean
+ setNomeJogadorPadrao(): void
+ iniciarJogoPersonalizado(Dificuldade, Assunto, int): void
+ setConfiguracoesResetar(): void
+ estatisticaContabilizarAcertosAssuntos(Map<String, Integer>): void
+ finalizarDebugWin(): void
- notificarObservadores(): void
+ getEstatisticaMaiorPontuacao(): int
+ getEstatisticaAssuntoMenorAcerto(): String
+ getInstancia(): Model
+ debugWinTemMaisQuestao(): boolean
+ getDebugWinPontuacao(): int
+ getEstatisticaTotalErros(): int
+ removerObservador(Observer): void
+ setConfiguracoesPontuacaoPorDificuldade(String, int): void
+ estatisticaContabilizarErrosAssuntos(Map<String, Integer>): void
+ getConfiguracoesTempoPorDificuldade(String): int
+ setJogador(Jogador): void
+ getMaiorSequenciaAcerto(): int
+ getBancoQuestoesAleatoria(int): List<Questao>
+ getGerenciadorBancoQuestoes(): Map<String, List<Questao>>
+ getBancoQuestoesPersonalizada(int, Dificuldade, Assunto): List<Questao>
+ setConfiguracoesQntdQuestoesPorJogo(int): void
+ getEstatisticaAssuntoMaiorAcerto(): String
+ atualizarEstatistica(): void
+ getJogador(): Jogador
+ getEstatisticaTotalAcertos(): int
+ adicionarObservador(Observer): void
+ setEstatisticaMaiorPontuacao(int): void
+ setConfiguracoesTempoPorDificuldade(String, int): void
«create»