pkg

model

Usuario<Abstract>
- idUsuario : int
- nome : String
- validador : ValidadorEmprestimo
- limiteEmprestimo : int

Emprestimo
- idEmprestimo : int
- status : String
- dataEmprestimo : LocalDate
- dataParaDevolucao : LocalDate

AlunoGraduacao

AlunoPosGraduacao

- usuario<Abstract>

Exemplar
- idExemplar : int
- status : String

Livro
- idLivro : int
- titulo : String
- editora : String
- autores : String
- edicao : int
- anoPublicacao : int

Professor
- registroNotificacoes : int

Reserva
- idReserva : int
- dataReserva : LocalDate

- livro
1..*

strategy

<<interface>>
ValidadorEmprestimo
+ validar() : void

Validador1
+ validar() : void

Validador2
+ validar() : void

observer

<<interface>>
Observador
+ notificar() : void

<>
Sujeito
+ addObservador(observer : Observador) : void
+ removerObservador(observer : Observador) : void
+ notificarObservadores() : void

command

Command

ComandoRealizarDevolucao
+ executar(stringComando[] : String) : void

ComandoRealizarEmprestimo
+ executar(stringComando[] : String) : void

ComandoRealizarConsultaLivro
+ executar(stringComando[] : String) : void

ComandoRegistrarObservador
+ executar(stringComando[] : String) : void

InicializadorComandos

ComandoRealizarReserva
+ executar(stringComando[] : String) : void

ComandoRealizarConsultaUsuario
+ executar(stringComando[] : String) : void

ComandoRealizarConsultaNotificacao
+ executar(stringComando[] : String) : void

system

InterfaceUsuario
- inicializarComandos : HashMap<String,Command>
- InterfaceUsuario()
- obterComandoConsole() : String
- executarComando() : void
+ realizarLoopInteracaoUsuario() : void

FachadaBiblioteca
- instancia : FachadaBiblioteca
- FachadaBiblioteca()
+ obterInstancia() : FachadaBiblioteca
+ realizarEmprestimo(stringComando[] : String) : void
+ realizarDevolucao(stringComando[] : String) : void
+ realizarReserva(stringComando[] : String) : void