

Plano de Testes PL01 - Objetivo é validar se a classe Obra atende aos requisitos propostos			
Caso	Descrição	Entrada	Saída Esperada
1	Verificar se a adição de obras está funcionando corretamente	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo	A obra deve ser adicionada sem nenhum erro/excessão
2	Verificar se a adição de obras rejeita obras com título vazio	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. O título deve possuir o valor "". Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo	Deve ser lançada a exceção IllegalArgumentException
3	Verificar se a adição de obras rejeita obras com ano vazio	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. O ano deve possuir o valor "". Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo	Deve ser lançada a exceção IllegalArgumentException
4	Verificar se a adição de obras rejeita obras com ano negativo	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. O ano deve possuir o valor "-1". Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo	Deve ser lançada a exceção IllegalArgumentException
5	Verificar se a adição de obras rejeita a entrada duplicada de uma obra	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo. Instancie um segundo objeto do tipo Obra com o construtor padrão e adicione via setter as mesmas informações do primeiro objeto. Após isso, adicione a segunda obra utilizando o mesmo método mencionado anteriormente	Deve ser lançada a exceção IllegalArgumentException
6	Verificar se a adição de obras rejeita obras com ano inválido	Crie uma instância de Termo com o construtor padrão. Após isso, crie uma instância de Obra com o construtor padrão, adicionando via setter título, ano e categoria. O ano deve possuir o valor "abc". Feito isso, adicione a obra utilizando o método "adicionandoObra" do objeto do tipo Termo	Deve ser lançada a exceção IllegalArgumentException
Plano de Testes PL02 - Objetivo é validar se a classe Personagem atende aos requisitos propostos			
Caso	Descrição	Entrada	Saída Esperada
1	Verificar se a adição de personagens está funcionando corretamente	Crie uma instância de Personagem, Termo e Glossario utilizando o construtor padrão. Crie os atributos "caracteristica" e "feitos" do tipo String e os preencha com dados. Crie um ArrayList do tipo String chamado "atores" e adicione pelo menos um dado a ele com o método .add(). Por fim, utilize o método adicionandoPersonagem() do objeto do tipo Personagem e insira como parâmetro características, feitos, atores, termo e glossario	O personagem deve ser adicionado sem nenhum erro/exceção
2	Verificar se a adição de personagens rejeita personagens com características vazia	Crie uma instância de Personagem, Termo e Glossario utilizando o construtor padrão. Crie os atributos "características" e "feitos" do tipo String e os preencha com dados. O atributo características deve receber o valor "". Crie um ArrayList do tipo String chamado "atores" e adicione pelo menos um dado a ele com o método .add(). Por fim, utilize o método adicionandoPersonagem() do objeto do tipo Personagem e insira como parâmetro características, feitos, atores, termo e glossario	Deve ser lançada a exceção IllegalArgumentException
3	Verificar se a adição de personagens rejeita personagens com feitos vazio	Crie uma instância de Personagem, Termo e Glossario utilizando o construtor padrão. Crie os atributos "características" e "feitos" do tipo String e os preencha com dados. O atributo feitos deve receber o valor "". Crie um ArrayList do tipo String chamado "atores" e adicione pelo menos um dado a ele com o método .add(). Por fim, utilize o método adicionandoPersonagem() do objeto do tipo Personagem e insira como parâmetro características, feitos, atores, termo e glossario	Deve ser lançada a exceção IllegalArgumentException
4	Verificar se a adição de personagens rejeita personagens sem atores informados	Crie uma instância de Personagem, Termo e Glossario utilizando o construtor padrão. Crie os atributos "características" e "feitos" do tipo String e os preencha com dados. Crie um ArrayList do tipo String chamado "atores" e não adicione nenhum dado a ele. Por fim, utilize o método adicionandoPersonagem() do objeto do tipo Personagem e insira como parâmetro características, feitos, atores, termo e glossario	Deve ser lançada a exceção IllegalArgumentException
5	Verificar se a inclusão de atores está ocorrendo corretamente	Crie uma instância de Personagem utilizando o construtor padrão e adicione um ator ao objeto através do método incluirAtores(), inserindo como parâmetro um valor do tipo String que não seja vazio	O ator deve ser adicionado sem nenhum erro/excessão
6	Verificar se a inclusão de atores rejeita a inclusão de atores vazio	Crie uma instância de Personagem utilizando o construtor padrão e adicione um ator ao objeto através do método incluirAtores(), inserindo como parâmetro o valor ""	Deve ser lançada a exceção IllegalArgumentException
7	Verificar se a inclusão de atores rejeita a inclusão de atores duplicado	Crie uma instância de Personagem utilizando o construtor padrão e adicione um ator ao objeto através do método incluirAtores(), inserindo como parâmetro um valor do tipo String que não seja vazio. Logo em seguida, adicione outro ator ao objeto através do método incluirAtores(), inserindo como parâmetro o mesmo valor informado anteriormente	Deve ser lançada a exceção IllegalArgumentException
8	Verificar se a remoção de atores está ocorrendo corretamente	Crie uma instância de Personagem utilizando o construtor padrão e adicione um ator ao objeto através do método incluirAtores(), inserindo como parâmetro um valor do tipo String que não seja vazio. Logo em seguida, remova o ator através do método removerAtores(), inserindo como parâmetro o ator instanciado anteriormente	O ator deve ser removido sem nenhum erro/excessão
Plano de Testes PL03 - Objetivo é validar se a classe Termo atende aos requisitos propostos			
Caso	Descrição	Entrada	Saída Esperada
1	Verificar se a inclusão de obras está ocorrendo normalmente	Crie uma instância de Termo e Obra utilizando o construtor padrão e adicione via setter no objeto do tipo Obra título, ano e categoria que não sejam vazios. Logo em seguida, utilize o método incluirObras do objeto do tipo Termo e insira como parâmetro a instância do objeto do tipo Obra	A obra deve ser adicionada sem nenhum erro/excessão
2	Verificar se a remoção de obras está ocorrendo normalmente	Crie uma instância de Termo e Obra utilizando o construtor padrão e adicione via setter no objeto do tipo Obra título, ano e categoria que não sejam vazios. Logo em seguida, utilize o método incluirObras() do objeto do tipo Termo e insira como parâmetro a instância do objeto do tipo Obra. Por fim, utilize o método removerObras() do objeto do tipo Termo e insira como parâmetro o objeto do tipo Obra instanciado anteriormente	A obra deve ser removida sem nenhum erro/excessão
Plano de Testes PL04 - Objetivo é validar se a classe Glossario atende aos requisitos propostos			
Caso	Descrição	Entrada	Saída Esperada
1	Validar se a consulta de termos está sendo realizada corretamente	Crie uma nova instância de Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Adicione o termo ao glossario com o método do tipo Glossario incluirTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente. Crie outra instância de Termo utilizando o construtor padrão e atribua a ele um valor com o método do tipo Glossario consultarTermos(), inserindo como parâmetro o mesmo nome e finalidade informados no primeiro termo. Utilize o comando System.out.println para exibir o nome e a finalidade do segundo objeto instanciado, utilizando o método get para ambos	Deve retornar o mesmo nome e finalidade do primeiro objeto do tipo Termo instanciado
2	Validar se a consulta de termos rejeita a consulta de um termo com nome vazio	Crie uma nova instância de Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Adicione o termo ao glossario com o método do tipo Glossario incluirTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente. Crie outra instância de Termo utilizando o construtor padrão e atribua a ele um valor com o método do tipo Glossario consultarTermos(), inserindo como parâmetro o valor "" e a mesma finalidade informada no primeiro termo. Utilize o comando System.out.println para exibir o nome e a finalidade do segundo objeto instanciado, utilizando o método get para ambos	Deve ser lançada a exceção IllegalArgumentException
3	Validar se a consulta de termos rejeita a consulta de um termo inexistente	Crie uma nova instância de Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Adicione o termo ao glossario com o método do tipo Glossario incluirTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente. Crie outra instância de Termo utilizando o construtor padrão e atribua a ele um valor com o método do tipo Glossario consultarTermos(), inserindo como parâmetro um nome diferente do informado para o primeiro termo e a mesma finalidade informada no primeiro termo. Utilize o comando System.out.println para exibir o nome e a finalidade do segundo objeto instanciado, utilizando o método get para ambos	Deve ser lançada a exceção IllegalArgumentException
4	Validar se a inclusão de termos está ocorrendo normalmente.	Crie uma nova instância de Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Adicione o termo ao glossario com o método do tipo Glossario incluirTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente	O objeto do tipo Termo deve ser incluído sem nenhum erro/excessão
5	Validar se a remoção de termos está ocorrendo normalmente.	Crie uma nova instância de Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Adicione o termo ao glossario com o método do tipo Glossario incluirTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente. Remova o objeto do tipo Termo do glossario utilizando o método do tipo Glossario removerTermos(), inserindo como parâmetro o objeto do tipo Termo criado anteriormente	O objeto do tipo Termo deve ser removido sem nenhum erro/excessão
Plano de Testes PL05 - Objetivo é validar se a classe Local atende aos requisitos propostos			
Caso	Descrição	Entrada	Saída Esperada
1	Validar se a adição de locais está ocorrendo corretamente	Crie uma instância de Local, Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Crie um atributo do tipo String chamado historia e informe um valor para ele que não seja vazio. Adicione o local utilizando o método adicionandoLocal() do tipo Local, informando como parâmetro os objetos do tipo Termo e Glossario e o atributo historia	O local deve ser adicionado sem nenhum erro/excessão
2	Validar se a adição de locais rejeita local com historia vazio	Crie uma instância de Local, Glossario e Termo utilizando o construtor padrão. Adicione via setter no objeto do tipo Termo nome, descricao e finalidade que não sejam vazios. Crie um atributo do tipo String chamado historia e informe o valor "". Adicione o local utilizando o método adicionandoLocal() do tipo Local, informando como parâmetro os objetos do tipo Termo e Glossario e o atributo historia	Deve ser lançada a exceção IllegalArgumentException