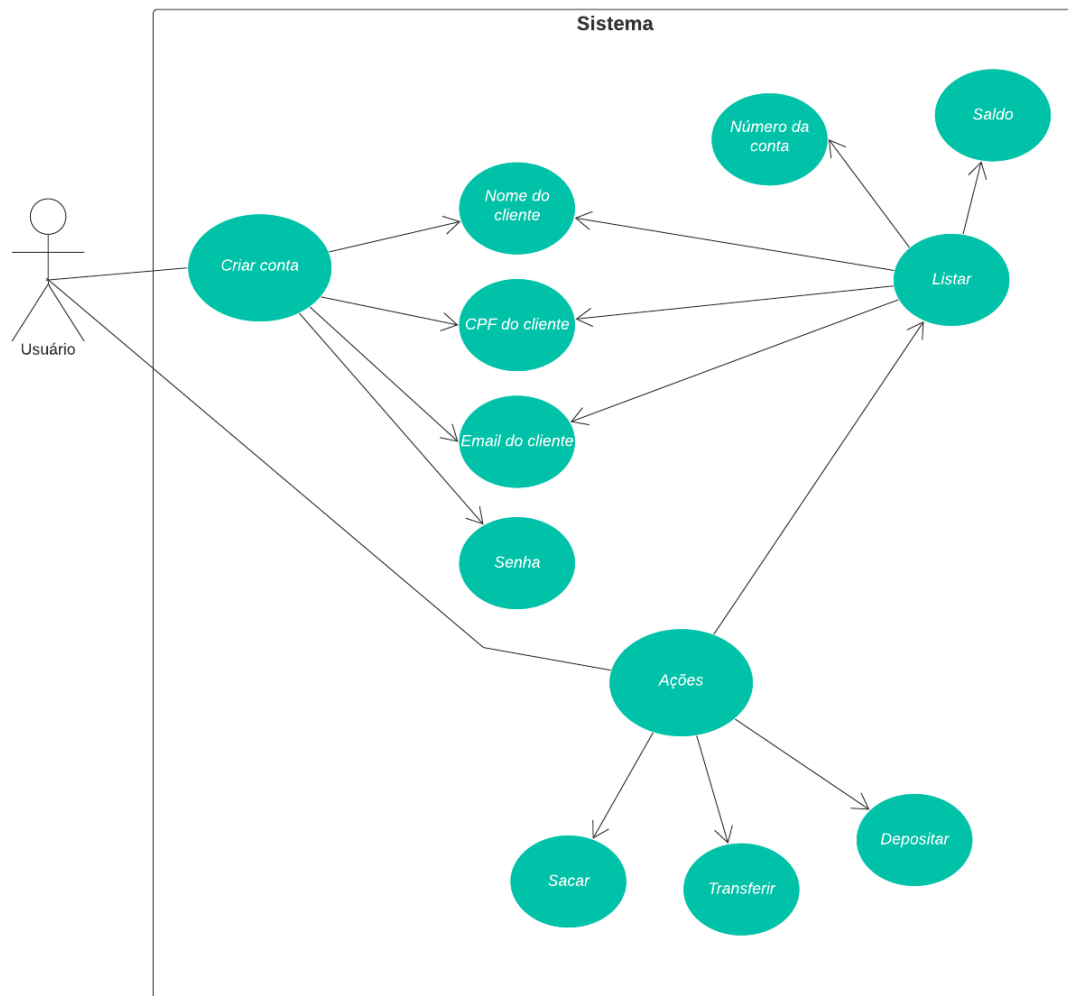


Plano de Testes

<Agência Bancária>

*Luhan Fernandes
Yasmin Branco
Camila Siqueira*



Conteúdo

1	INTRODUÇÃO	3
2	REQUISITOS A TESTAR	3
2.1	ITERAÇÃO 1	3
2.2	ITERAÇÃO 2	3
2.3	ITERAÇÃO 3	3
3	TIPOS DE TESTE	3
3.1	ITERAÇÃO 1	3
3.2	ITERAÇÃO 2	3
4	RECURSOS	3
4.1	AMBIENTE DE TESTE – SOFTWARE & HARDWARE	3
4.2	FERRAMENTAS DE TESTE	3
5	REFERÊNCIAS	3

1 Introdução

O sistema trata-se de um sistema bancário, onde o usuário pode criar contas bancárias contendo E-mail, nome, CPF, senha e data de criação. As outras ações que o usuário pode utilizar são: Fazer depósito, sacar da conta, fazer uma transferência e listar as contas. O sistema foi feito com a linguagem de programação *java*

2 Requisitos a Testar

Esta seção contém os requisitos que são objetos dos testes a serem realizados. Esses requisitos são divididos, por iteração, em casos de uso e requisitos não funcionais conforme descrito abaixo.

2.1 Iteração 1: Criar Conta



Casos de Uso

Identificador do Caso de Uso	Nome do Caso de Uso
<Id UC1>	<UC1> Teste de criação de conta com nome válido
<Id UC2>	<UC2> Teste para garantir que um número de conta único seja atribuído.
<Id UC3>	<UC3> Teste para verificar a criação de senhas válidas.
<Id UC4>	<UC4> Teste para a validação de E-mail.

Requisitos Não-Funcionais

Identificador do Requisito	Nome do Requisito
<Id Req1>	<Req1> O cliente deve fornecer informações pessoais para abrir uma conta.
<Id Req2>	<Req2> Garantir que um número de conta único seja atribuído a cada conta.
<Id Req3>	<Req3> Validar se as informações pessoais do cliente são registradas corretamente.
<Id Req4>	<Req4> O sistema deve validar o E-mail utilizado pelo usuário.

2.2 Iteração 2: Depositar

Casos de Uso

Identificador do Caso de Uso	Nome do Caso de Uso
<Id UC1>	<UC1> Verificar se o saldo da conta é atualizado corretamente.
<Id UC2>	<UC2> Testar se o sistema permite que o cliente deposite fundos em uma conta.
<Id UC3>	<UC3> Testar se o tipo de entrada é válido.
<Id UC4>	<UC4> Verificar senha

Requisitos Não-Funcionais

Identificador do Requisito	Nome do Requisito
<Id Req1>	<Req1> O sistema deve atualizar o saldo da conta.
<Id Req2>	<Req2> O cliente deve fornecer o número da conta e o valor a ser depositado.
<Id Req3>	<Req3> O cliente deve inserir tipos de dados válidos para cada entrada.
<Id Req4>	<Req4> O sistema deve verificar a senha da conta.

2.3 Iteração 3: Transferir

Casos de Uso

Identificador do Caso de Uso	Nome do Caso de Uso
<Id UC1>	<UC1> Verificar se o limite do saldo é respeitado nas transações.
<Id UC2>	<UC2> Verificar se o saldo da conta é atualizado após a transferência
<Id UC3>	<UC3> Garantir que a conta da transação seja existente

Requisitos Não-Funcionais

Identificador do Requisito	Nome do Requisito
----------------------------	-------------------

<Id Req1>	<Req1> O sistema deve verificar se há saldo suficiente para a transação.
<Id Reg2>	<Req2> O sistema deve verificar a senha da conta.
<Id Req3>	<Req3> O sistema deve atualizar o armazenamento das contas

2.4 Iteração 4: Sacar

Casos de Uso

Identificador do Caso de Uso	Nome do Caso de Uso
<Id UC1>	<UC1> Verificar se o valor do saque é igual ou menor que o saldo.
<Id UC2>	<UC2> Verificar senha.
<Id UC3>	<UC3> Garantir que o saldo da conta seja atualizado corretamente.

Requisitos Não-Funcionais

Identificador do Requisito	Nome do Requisito
<Id Req1>	<Req1> O sistema deve verificar se há saldo suficiente para saque.
<Id Reg2>	<Req2> O sistema deve verificar a senha para que ocorra o saque.
<Id Req3>	<Req3> O sistema deve atualizar o saldo da conta.

3 Tipos de Teste

3.1 Iteração 1

Exemplo:

Objetivo:	Testar os requisitos do sistema que envolvem a coleta de informações pessoais do cliente, criar um registro único de conta associado a esse cliente e armazenar as informações da conta no sistema.
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	Programadores (ou equipe da subcontratada)

3.2 Iteração 2

Objetivo:	Testar os requisitos do sistema que envolvem permitir que os clientes depositem dinheiro em suas contas bancárias. Essa funcionalidade é fundamental para que os clientes possam adicionar fundos às suas contas.
------------------	---

Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	Programadores (ou equipe da subcontratada)

3.3 Iteração 3

Objetivo:	Testar os requisitos do sistema que envolvem permitir que os clientes transfiram fundos de uma conta para outra, seja entre suas próprias contas ou para contas de terceiros.
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	Programadores (ou equipe da subcontratada)

3.4 Iteração 4

Objetivo:	Testar os requisitos do sistema que envolvem permitir que os clientes retirem fundos de suas contas bancárias. Essa funcionalidade é fundamental para que os clientes possam acessar o dinheiro que têm disponível em suas contas para uso pessoal ou outras necessidades.
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	Programadores (ou equipe da subcontratada)

4 Recursos

De extrema importância para o bom andamento dos testes, os recursos a serem utilizados durante os testes são descritos nessa seção. Os recursos estão divididos nas subseções que se seguem.

4.1 Ambiente de Teste – Software & Hardware

OS: Windows 10

Browser: Chrome

Processador: Core i5 7700u

Memoria ram: 24gb ddr4

Compilador: Eclipse IDE 2023-09

VM: Java JDK 17

4.2 Ferramentas de Teste

A ferramenta utilizada para os testes foi o framework JUnit. Esse framework fornece ferramentas e recursos que auxiliam a escrita de testes unitários.

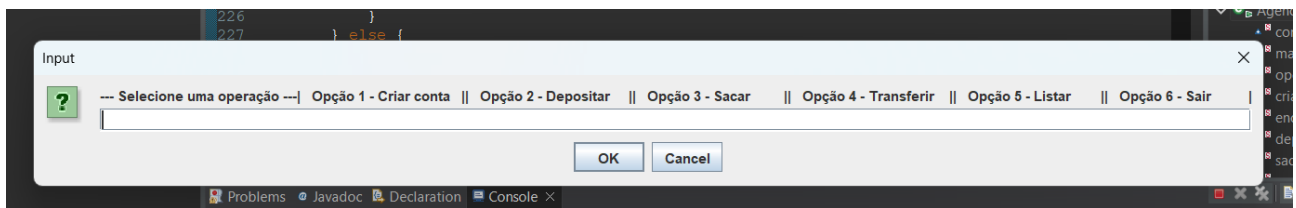
Também foram realizados em sua maioria testes manuais, simulando a utilização a partir da experiência de usuário no funcionamento do software.

5 Referências

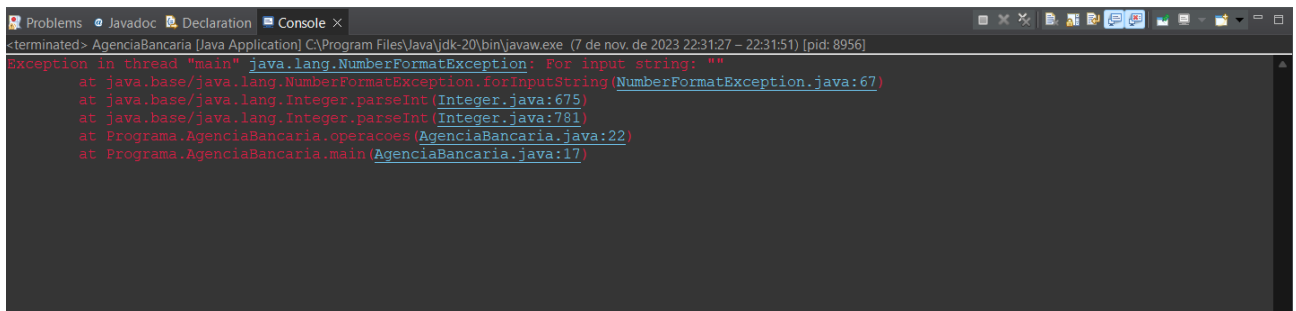
<https://github.com/luhanfernandes/AgenciaBancaria.git>

Menu inicial

Entrada: valor nulo



Saída:



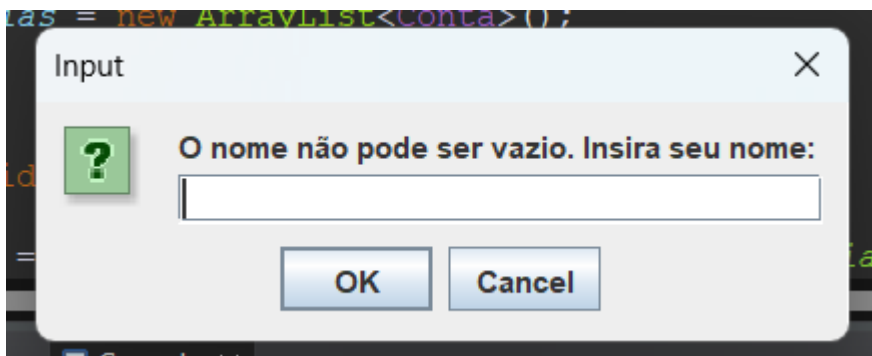
❌ *TESTE FAIL!* ❌

Teste Iteração 1: Criar conta

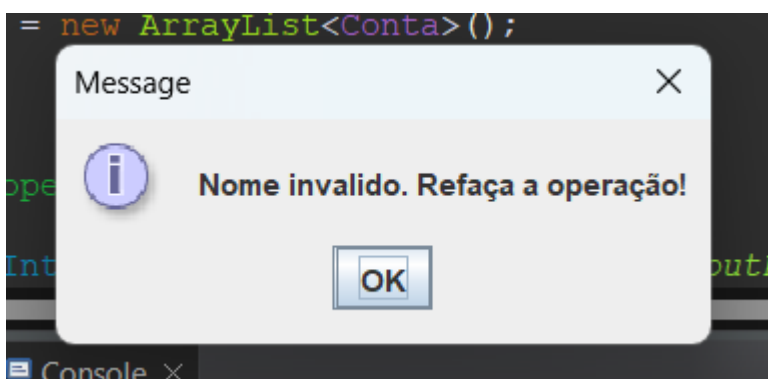
Teste 1: Nome de usuário válido

Entrada 1: valor nulo

Saída 2: tratamento de erro



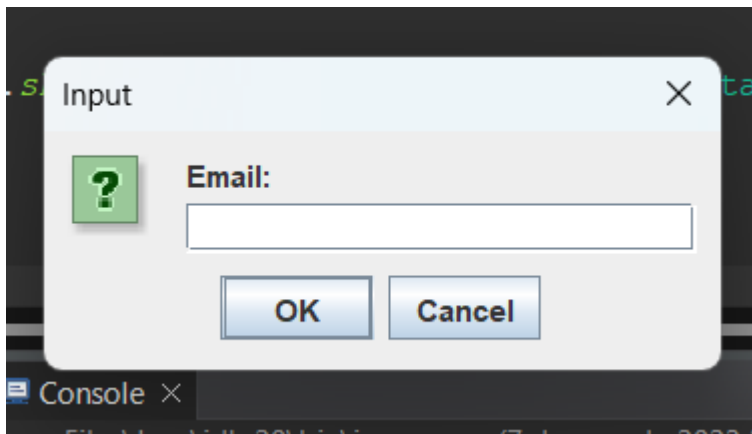
Saída 2:



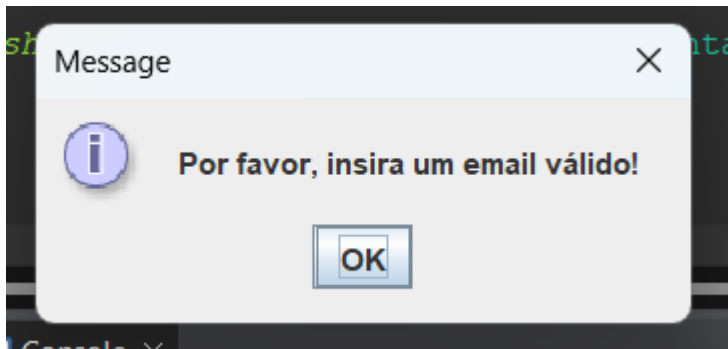
✅ *TESTE APROVADO!* ✅

Teste 2: E-mail válido

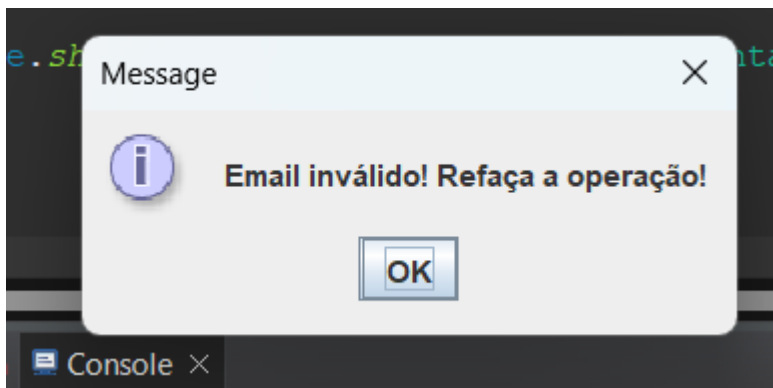
Entrada: teste



Saída:



Saída no caso de um email invalido inserido novamente:

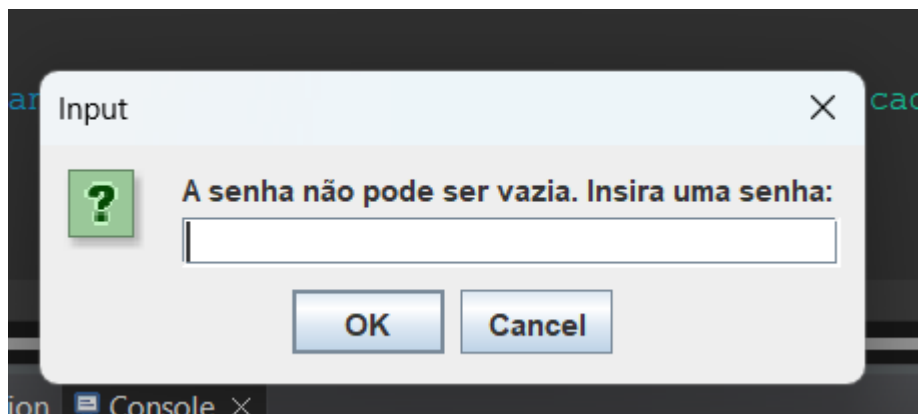


✓ **TESTE APROVADO!** ✓

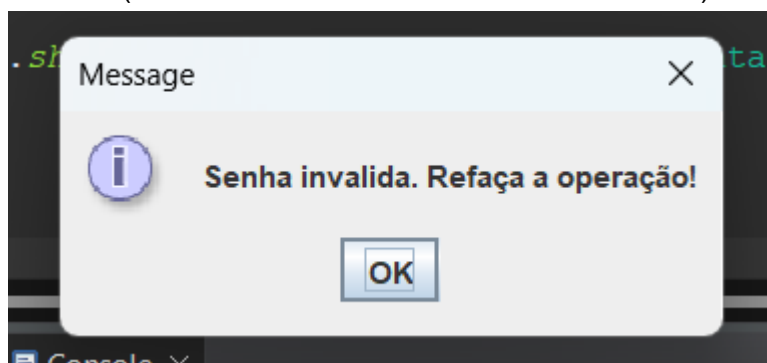
Teste 3: Senha vazia

Entrada: valor nulo

Saída 1:



Saída 2: (no caso de inserir novamente um valor nulo)

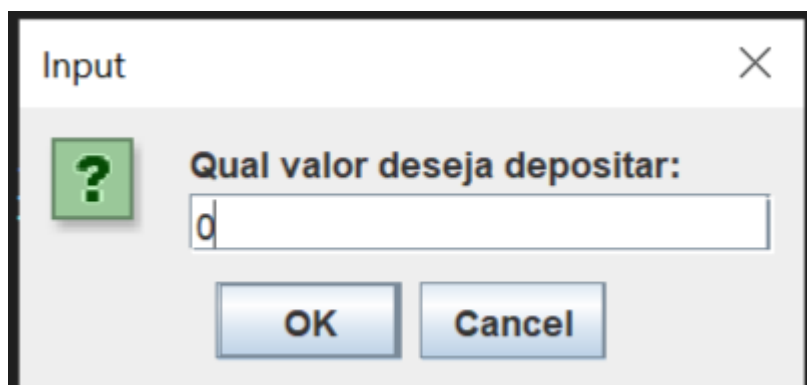


✓ TESTE APROVADO! ✓

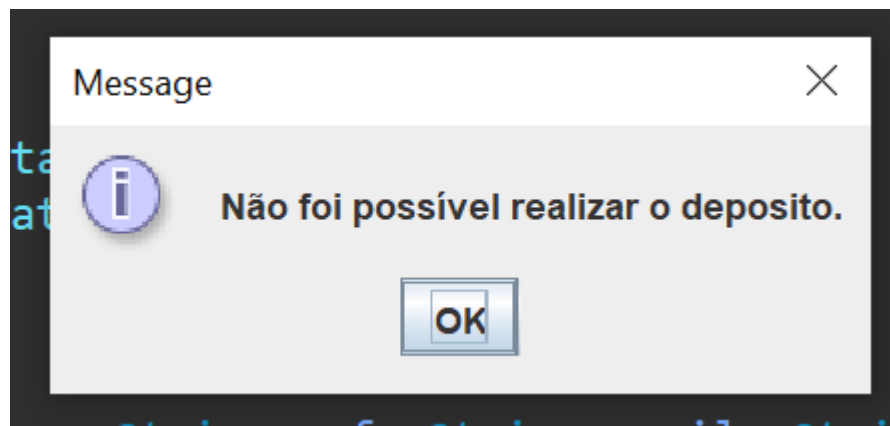
Teste Iteração 2: Depósito

Teste 1: Valor do depósito 0

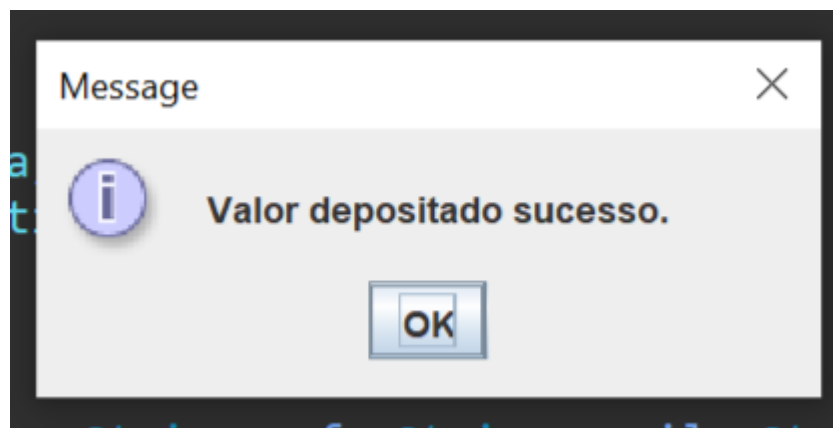
Entrada:



Saída 1:



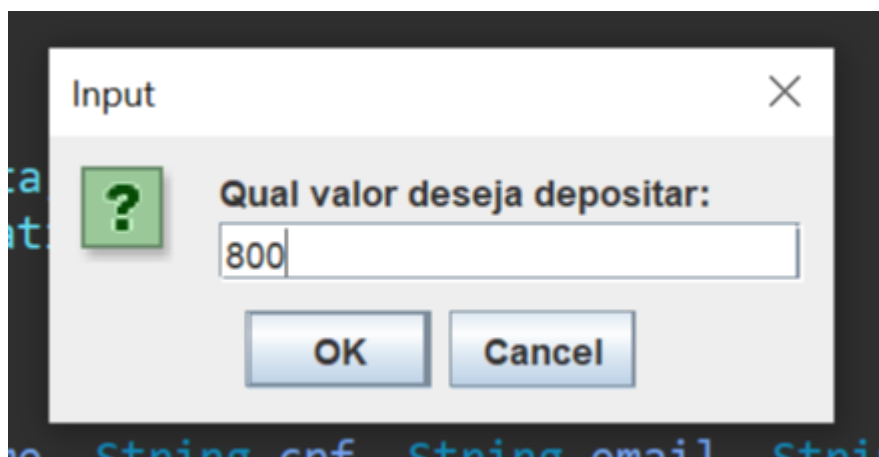
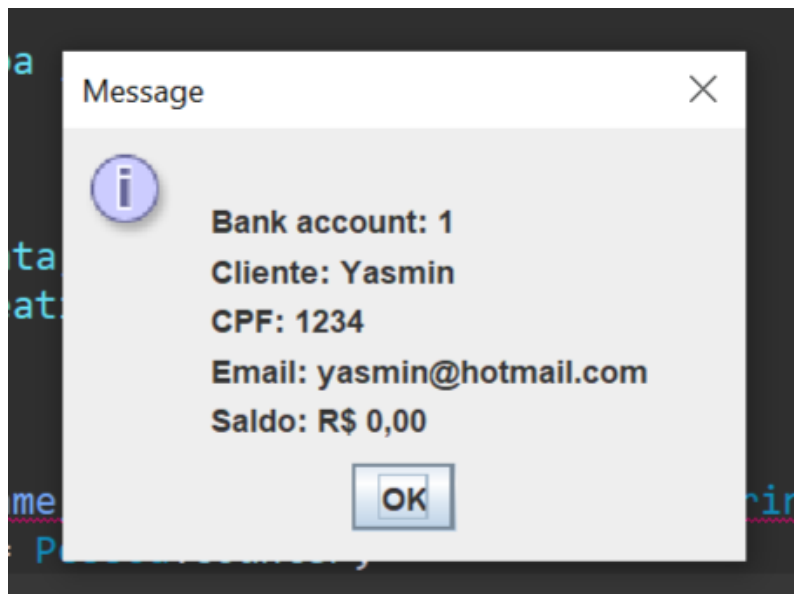
Saída 2:



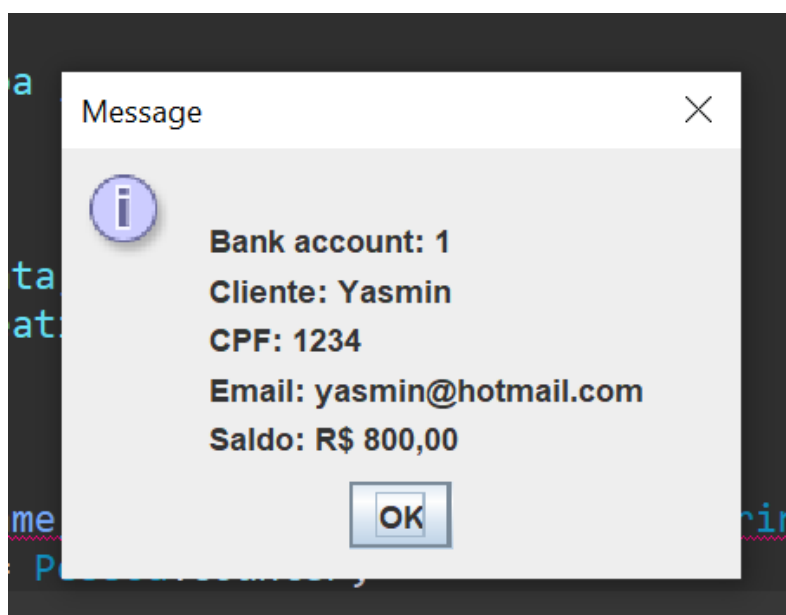
✓ TESTE APROVADO! ✓

Teste 2: Atualização de saldo na conta

Entrada: valor do depósito



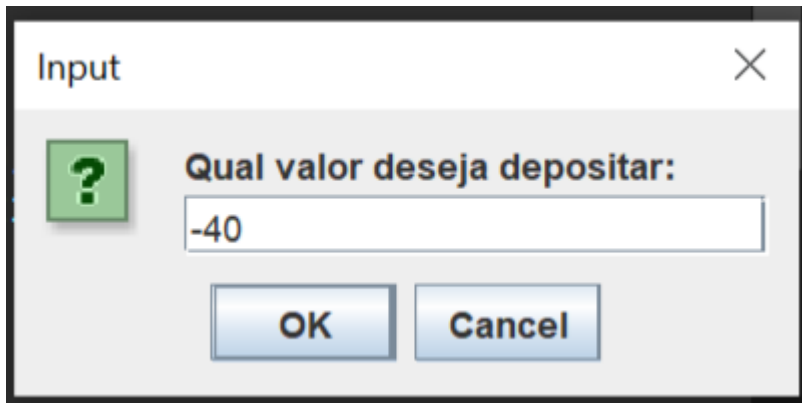
Saída:



✓ TESTE APROVADO! ✓

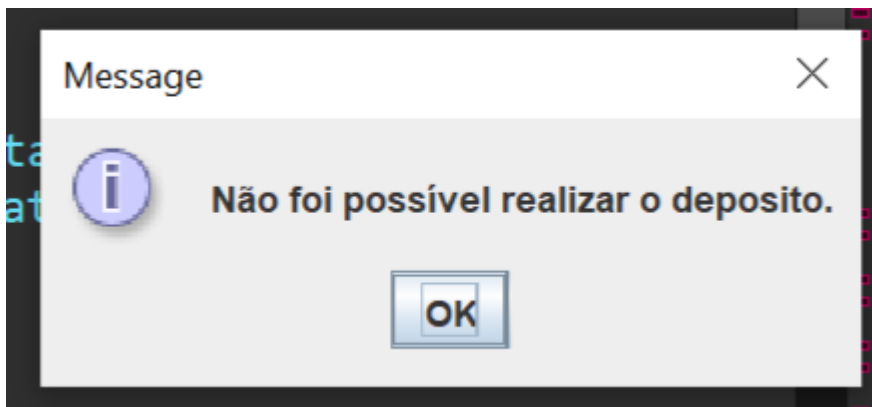
Teste 2: Valor negativo

Entrada:



An input dialog box titled "Input" with a close button (X) in the top right corner. It features a green question mark icon on the left. The text "Qual valor deseja depositar:" is displayed above a text input field. The input field contains the value "-40". Below the input field are two buttons: "OK" and "Cancel".

Saída:

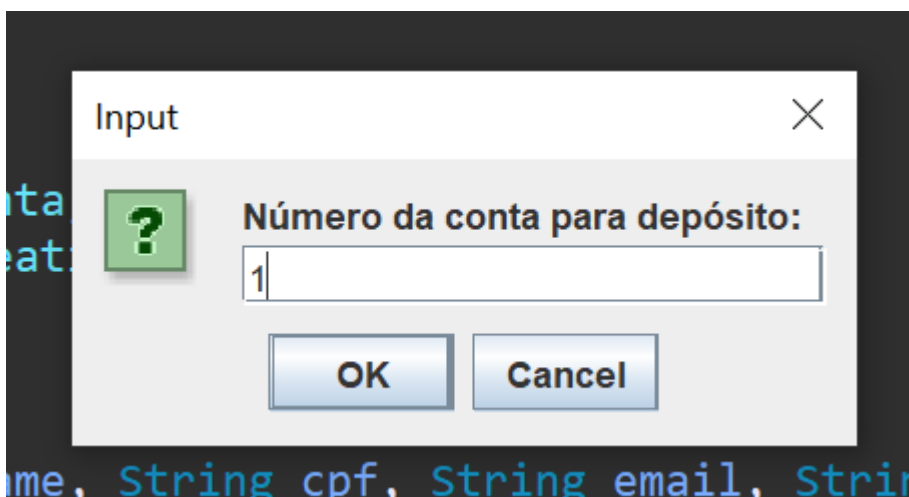


A message dialog box titled "Message" with a close button (X) in the top right corner. It features a blue information icon (i) on the left. The text "Não foi possível realizar o depósito." is displayed. Below the text is a single "OK" button.

✓ TESTE APROVADO! ✓

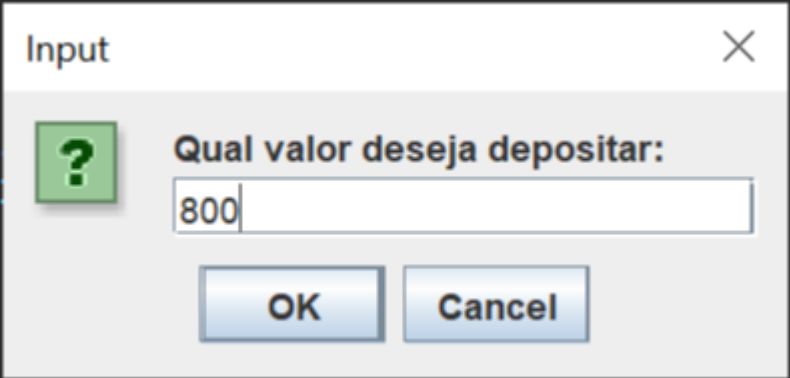
Teste 3: Depósito corretamente efetuado, com valor e conta

Primeira Entrada: Número da conta



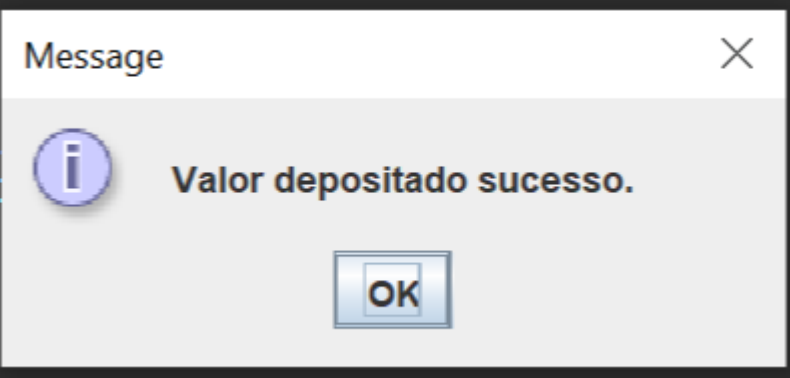
An input dialog box titled "Input" with a close button (X) in the top right corner. It features a green question mark icon on the left. The text "Número da conta para depósito:" is displayed above a text input field. The input field contains the value "1". Below the input field are two buttons: "OK" and "Cancel".

Segunda Entrada: Valor do depósito



The image shows a standard Java Swing 'Input' dialog box. It has a title bar with the word 'Input' and a close button (X). Inside the dialog, there is a green square icon with a white question mark. To the right of the icon, the text 'Qual valor deseja depositar:' is displayed. Below this text is a text input field containing the number '800'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Saída:



The image shows a standard Java Swing 'Message' dialog box. It has a title bar with the word 'Message' and a close button (X). Inside the dialog, there is a blue circular icon with a white lowercase 'i'. To the right of the icon, the text 'Valor depositado sucesso.' is displayed. At the bottom center of the dialog, there is a single 'OK' button.

✓ *TESTE APROVADO!* ✓

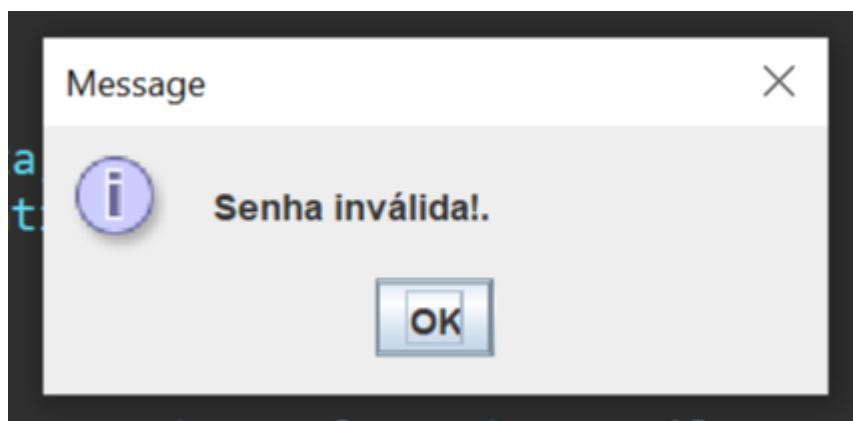
Teste 4: Validação de senha

Entrada 1: Senha incorreta



The image shows a standard Java Swing 'Input' dialog box. It has a title bar with the word 'Input' and a close button (X). Inside the dialog, there is a green square icon with a white question mark. To the right of the icon, the text 'Digite a senha da conta:' is displayed. Below this text is a text input field containing the number '7890'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

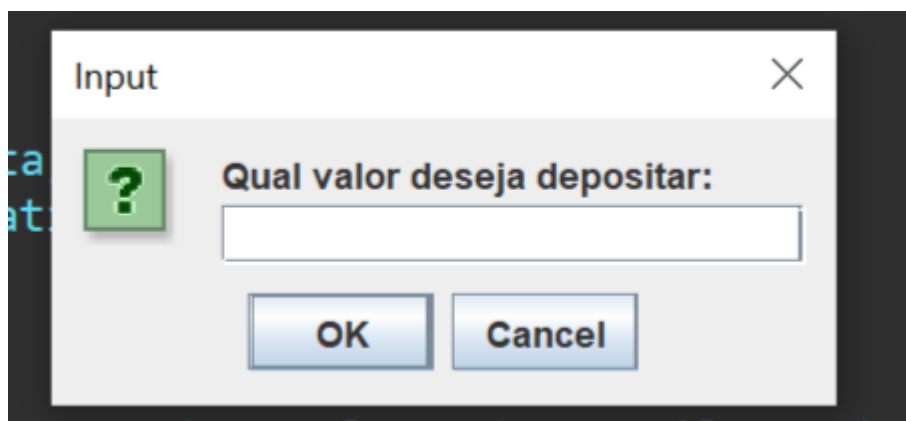
Saída 1:



Entrada 2: Senha correta



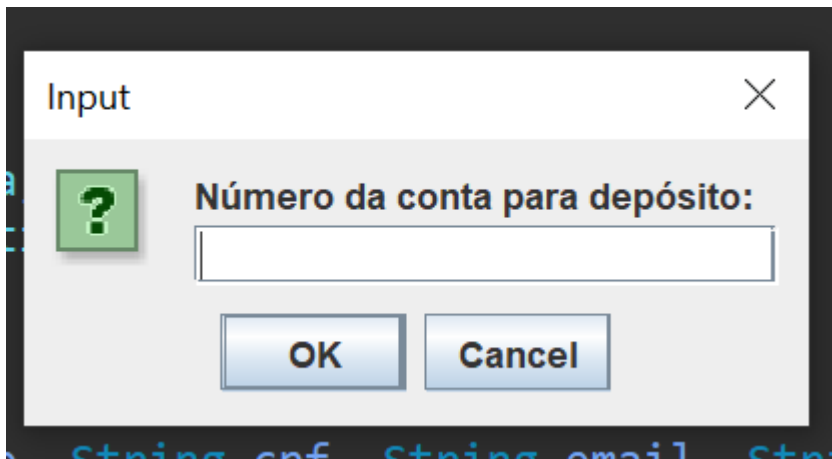
Saída 2: Senha válida



✓ TESTE APROVADO! ✓

Teste 5: Conta com valor nulo

Entrada: Valor nulo



Saída:

```
Exception in thread "main" java.lang.NumberFormatException: For input string: ""
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:675)
    at java.base/java.lang.Integer.parseInt(Integer.java:781)
    at Programa.AgenciaBancaria.depositar(AgenciaBancaria.java:137)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:33)
    at Programa.AgenciaBancaria.depositar(AgenciaBancaria.java:160)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:33)
    at Programa.AgenciaBancaria.listarContas(AgenciaBancaria.java:231)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:45)
    at Programa.AgenciaBancaria.criarConta(AgenciaBancaria.java:119)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:29)
    at Programa.AgenciaBancaria.depositar(AgenciaBancaria.java:160)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:33)
    at Programa.AgenciaBancaria.criarConta(AgenciaBancaria.java:119)
    at Programa.AgenciaBancaria.operacoes(AgenciaBancaria.java:29)
    at Programa.AgenciaBancaria.main(AgenciaBancaria.java:17)
```

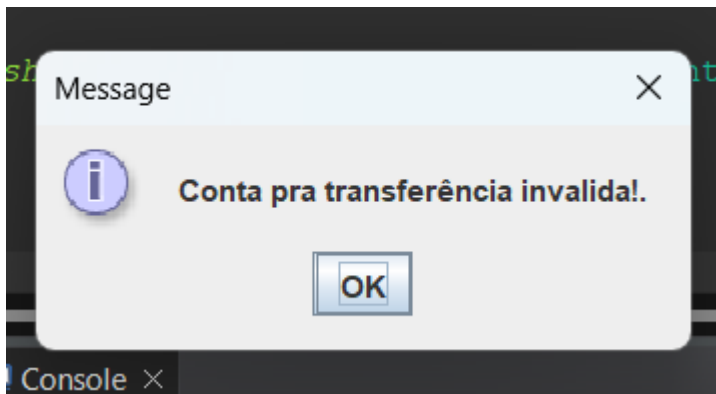
✗ TESTE FAIL! ✗

Teste Iteração 3: Transferir

Teste 1: Número de conta inexistente

Entrada: número de conta 2, que ainda não havia sido cadastrado

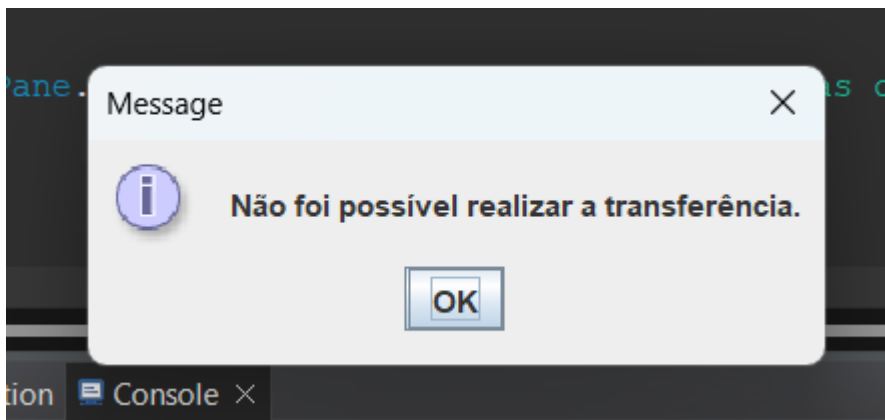
Saída:



Teste 2: Transferência de valor maior do que o saldo disponível na conta

Entrada: transferência de 200 reais, com conta de saldo 22 reais

Saída:

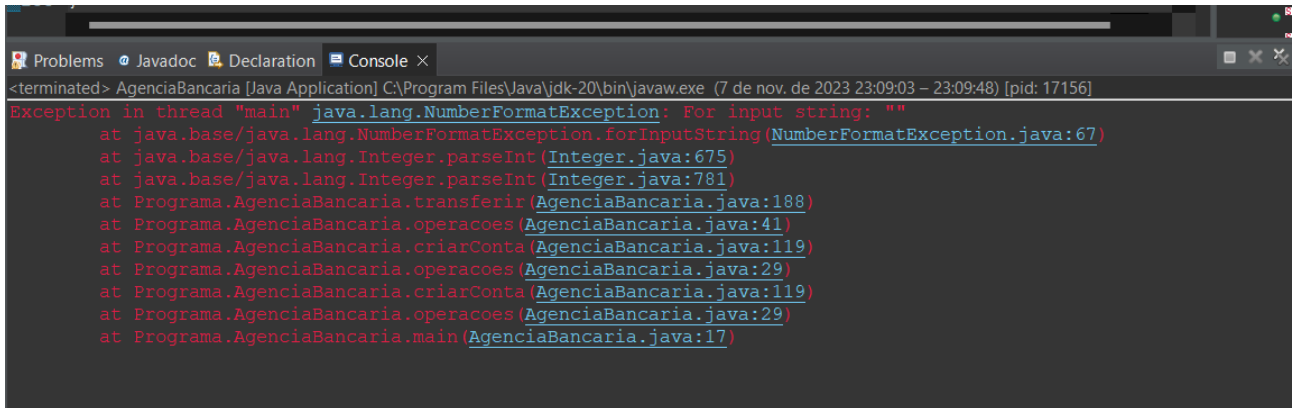


✓ TESTE APROVADO! ✓

Teste 3: Transferência com valor nulo

Entrada: valor nulo (nada foi colocado como entrada)

Saída:

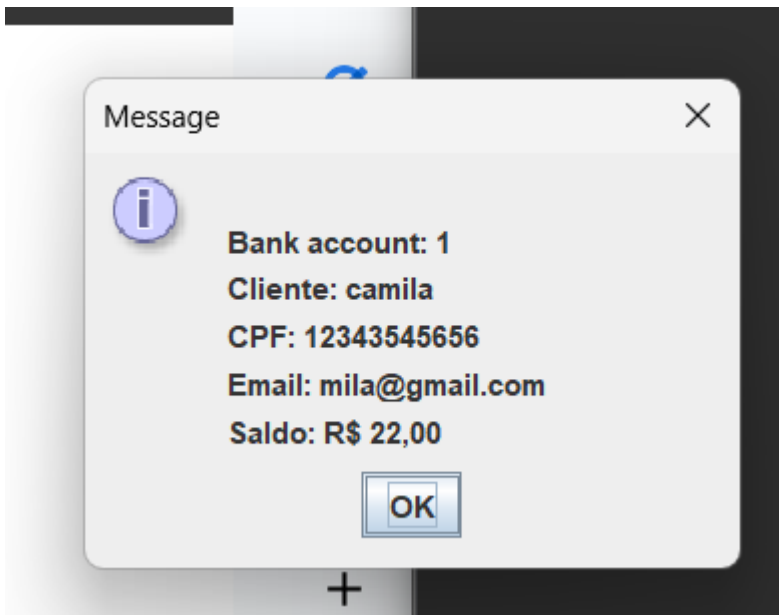


```
<terminated> AgenciaBancaria [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (7 de nov. de 2023 23:09:03 - 23:09:48) [pid: 17156]
Exception in thread "main" java.lang.NumberFormatException: For input string: ""
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:675)
    at java.base/java.lang.Integer.parseInt(Integer.java:781)
    at Programa.AgencyBancaria.transferir(AgencyBancaria.java:188)
    at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:41)
    at Programa.AgencyBancaria.criarConta(AgencyBancaria.java:119)
    at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:29)
    at Programa.AgencyBancaria.criarConta(AgencyBancaria.java:119)
    at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:29)
    at Programa.AgencyBancaria.main(AgencyBancaria.java:17)
```

✗ *TESTE FAIL!* ✗

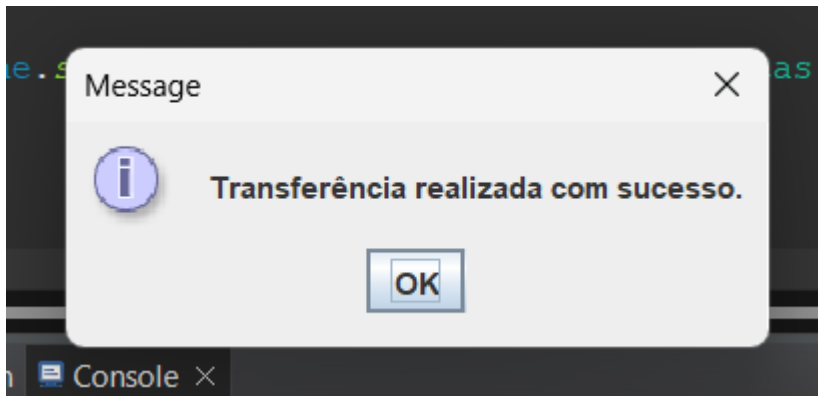
Teste 4: Transferência com valor dentro dos limites da conta

Saldo da conta:



Valor de transferência: 20 reais

Saída:

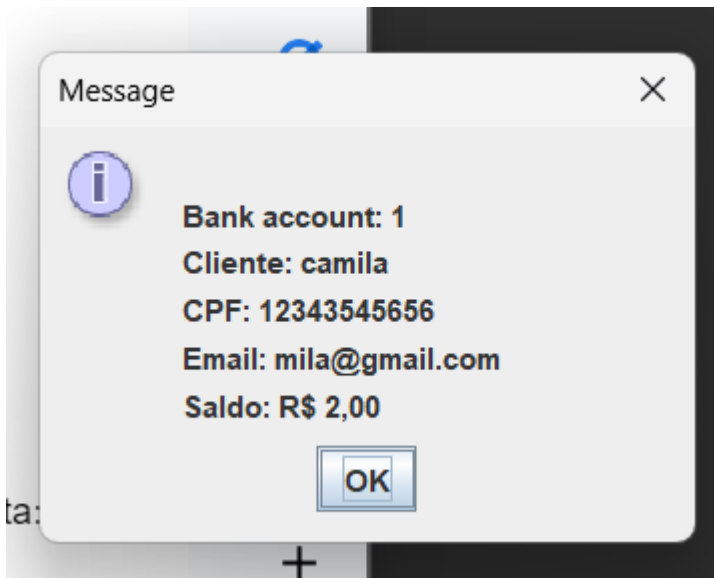


✓ **TESTE APROVADO!** ✓

Teste 4: Verificação de atualização automática do saldo da conta

Método: listar

Saída:

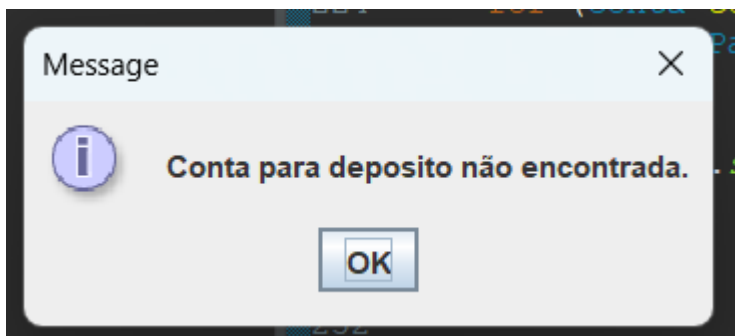


✓ **TESTE APROVADO!** ✓

Teste 5: Transferência para conta inexistente

Entrada: Conta de número 2, que ainda não havia sido cadastrada

Saída:

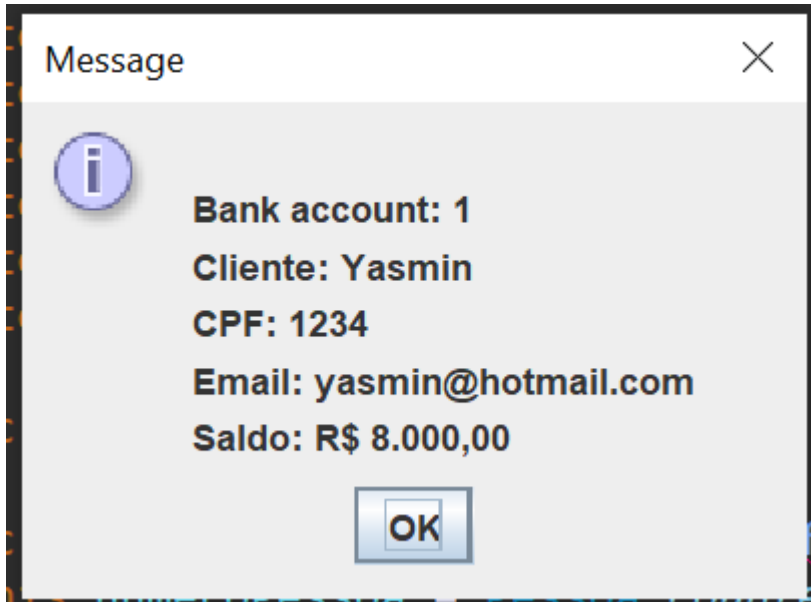


✓ **TESTE APROVADO!** ✓

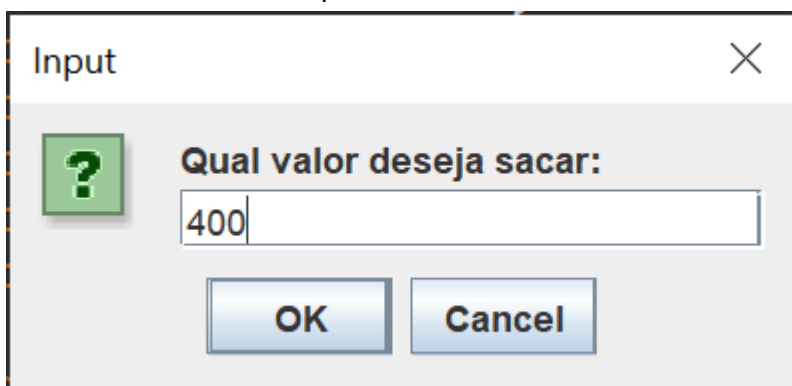
Teste Iteração 4: Sacar

Teste 1: Verificação de valor de saque

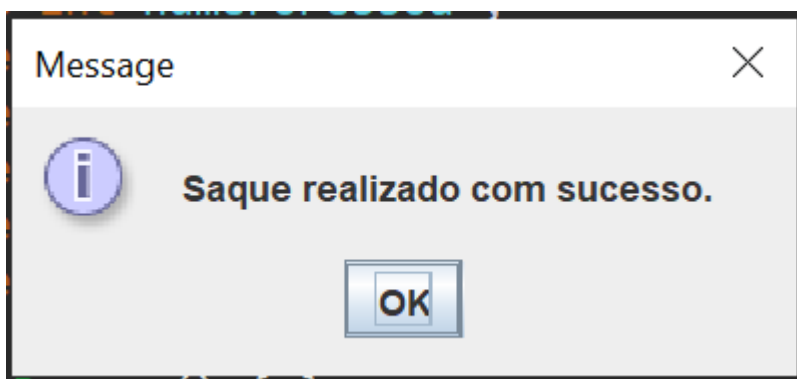
Saldo na conta:



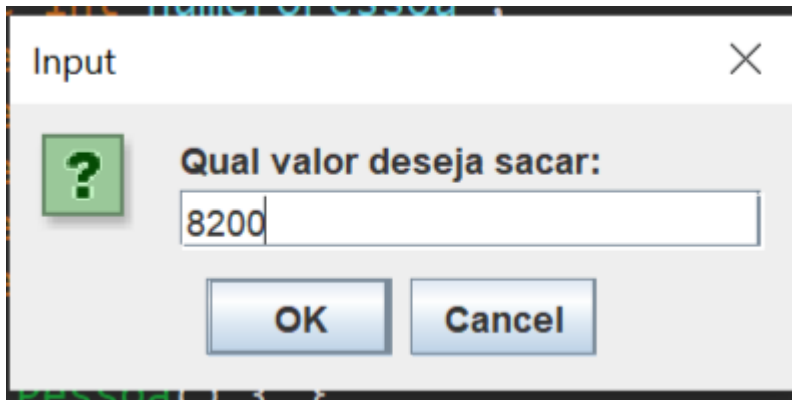
Entrada 1: Valor menor que o saldo



Saída 1:



Entrada 2: Valor maior do que saldo



Input

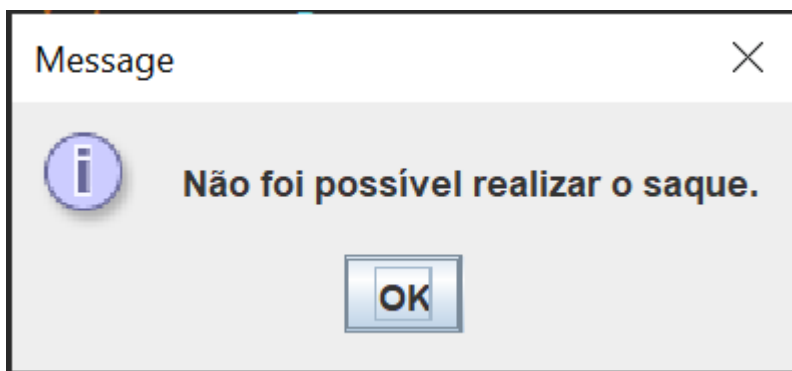
Qual valor deseja sacar:

8200

OK Cancel

Detailed description: This is a standard Windows-style input dialog box. It has a title bar with the word 'Input' and a close button (X). The main area has a light gray background. On the left, there is a green square icon with a white question mark. To its right, the text 'Qual valor deseja sacar:' is displayed. Below this text is a text input field containing the number '8200'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Saída 2:



Message

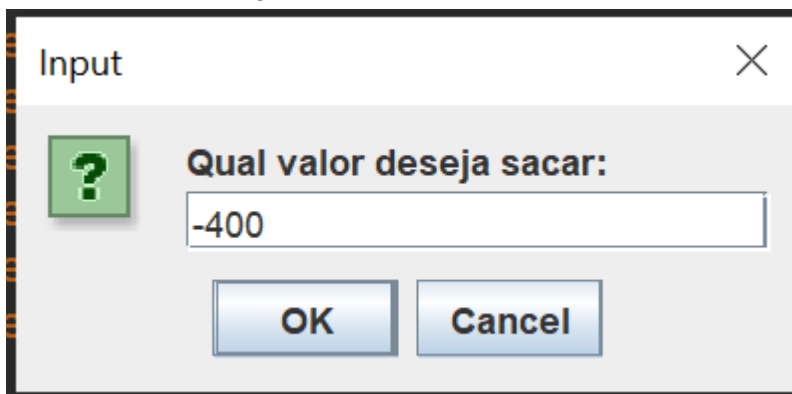
Não foi possível realizar o saque.

OK

Detailed description: This is a standard Windows-style message dialog box. It has a title bar with the word 'Message' and a close button (X). The main area has a light gray background. On the left, there is a blue circular icon with a white lowercase 'i'. To its right, the text 'Não foi possível realizar o saque.' is displayed. At the bottom center of the dialog, there is a single button labeled 'OK'.

✓ TESTE APROVADO! ✓

Entrada 3: Valor negativo



Input

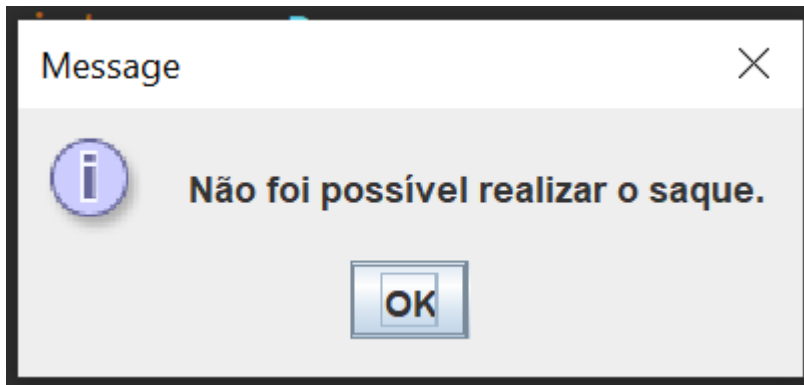
Qual valor deseja sacar:

-400

OK Cancel

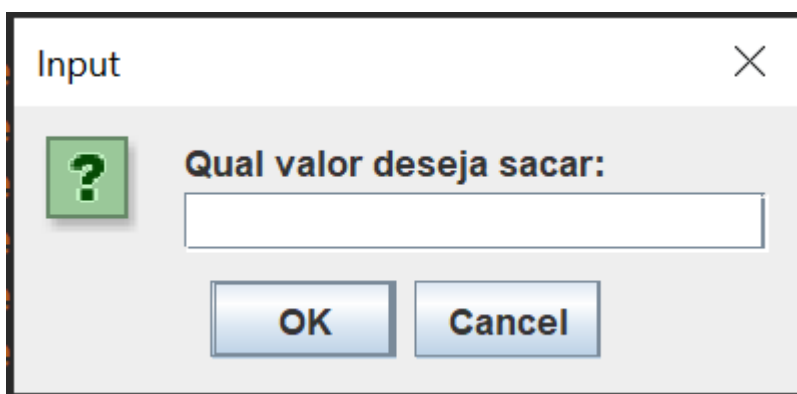
Detailed description: This is a standard Windows-style input dialog box, similar to the one in the first image. It has a title bar with the word 'Input' and a close button (X). The main area has a light gray background. On the left, there is a green square icon with a white question mark. To its right, the text 'Qual valor deseja sacar:' is displayed. Below this text is a text input field containing the number '-400'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Saída 3:



✓ *TESTE APROVADO!* ✓

Entrada 4: Valor nulo



Saída 4:

```
at java.base/java.lang.Double.parseDouble(Double.java:792)
at Programa.AgencyBancaria.sacar(AgencyBancaria.java:174)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:37)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.sacar(AgencyBancaria.java:183)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:37)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.sacar(AgencyBancaria.java:183)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:37)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.depositar(AgencyBancaria.java:160)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:33)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.sacar(AgencyBancaria.java:183)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:37)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.depositar(AgencyBancaria.java:160)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:33)
at Programa.AgencyBancaria.criarConta(AgencyBancaria.java:119)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:29)
at Programa.AgencyBancaria.listarContas(AgencyBancaria.java:231)
at Programa.AgencyBancaria.operacoes(AgencyBancaria.java:45)
at Programa.AgencyBancaria.main(AgencyBancaria.java:17)
```

✗ *TESTE FAIL!* ✗

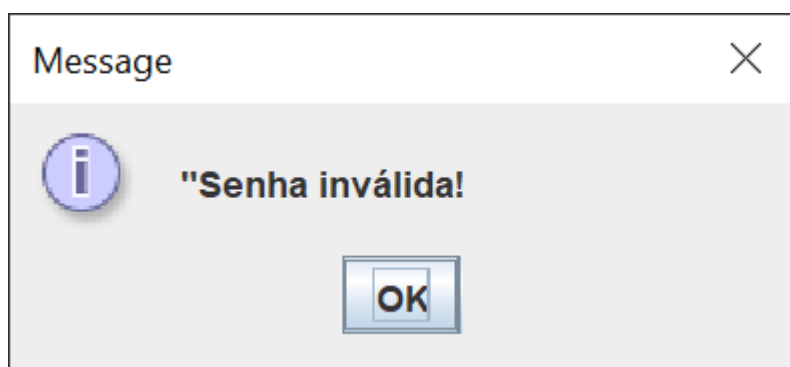
Teste 2: Verificação de senha

Entrada 1: Senha incorreta



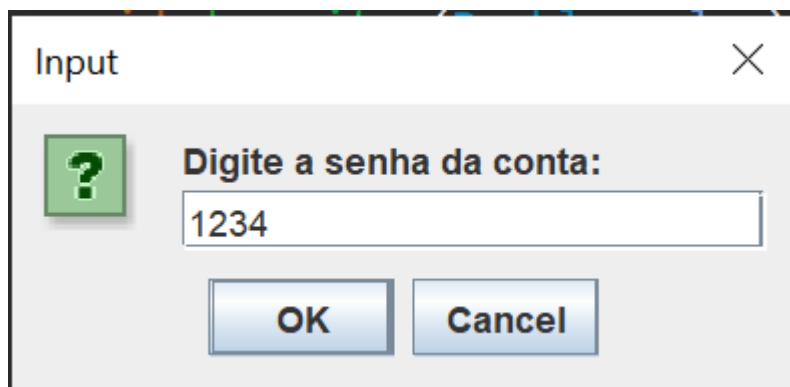
A screenshot of a Windows-style 'Input' dialog box. The title bar says 'Input' with a close button (X) on the right. The dialog has a light gray background. On the left, there is a green square icon with a white question mark. To the right of the icon, the text 'Digite a senha da conta:' is displayed in bold. Below this text is a text input field containing the number '8910'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Saída 1:



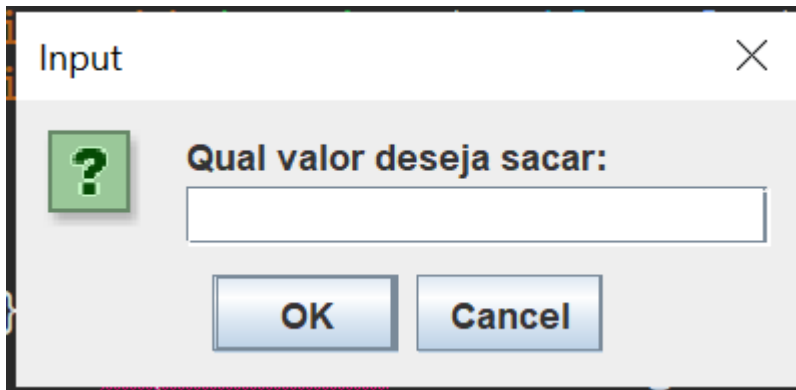
A screenshot of a Windows-style 'Message' dialog box. The title bar says 'Message' with a close button (X) on the right. The dialog has a light gray background. On the left, there is a purple circular icon with a white lowercase 'i'. To the right of the icon, the text '"Senha inválida!' is displayed in bold. At the bottom center of the dialog, there is a single button labeled 'OK'.

Entrada 2: Senha correta



A screenshot of a Windows-style 'Input' dialog box, identical in layout to the first one. The title bar says 'Input' with a close button (X) on the right. The dialog has a light gray background. On the left, there is a green square icon with a white question mark. To the right of the icon, the text 'Digite a senha da conta:' is displayed in bold. Below this text is a text input field containing the number '1234'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Saída 2: Senha verificada

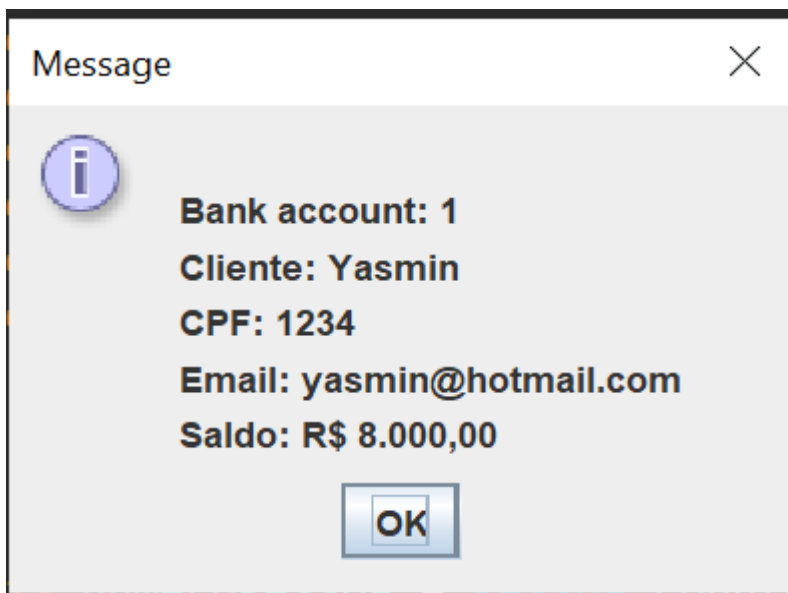


A dialog box titled "Input" with a close button (X) in the top right corner. On the left, there is a green square icon with a white question mark. To the right of the icon, the text "Qual valor deseja sacar:" is displayed. Below this text is a white text input field. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

✓ *TESTE APROVADO!* ✓

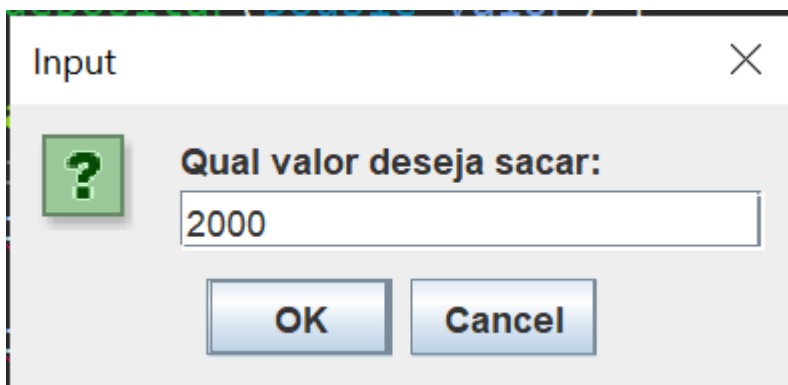
Teste 3: Atualização do saldo após o saque

Saldo da conta:



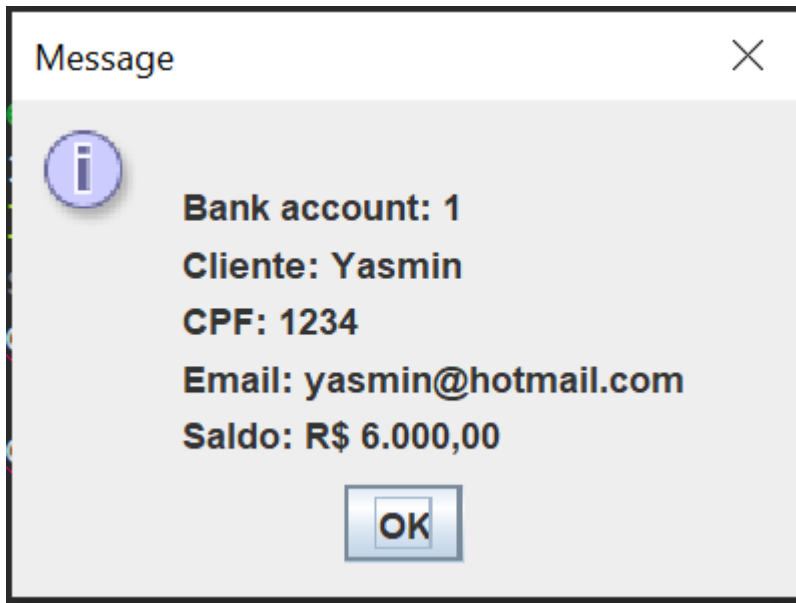
A dialog box titled "Message" with a close button (X) in the top right corner. On the left, there is a purple circular icon with a white lowercase 'i'. To the right of the icon, the following text is displayed: "Bank account: 1", "Cliente: Yasmin", "CPF: 1234", "Email: yasmin@hotmail.com", and "Saldo: R\$ 8.000,00". At the bottom center of the dialog, there is an "OK" button.

Entrada:



A dialog box titled "Input" with a close button (X) in the top right corner. On the left, there is a green square icon with a white question mark. To the right of the icon, the text "Qual valor deseja sacar:" is displayed. Below this text, the input field contains the value "2000". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Saída:



✓ *TESTE APROVADO!* ✓

Controle de qualidade

O cálculo realizado para descobrir a taxa de falhas será feito por:

$$TF = (N / T) * 100$$

TF = Taxa de Falha

N = Número de Testes que Falharam

T = Total de Testes Realizados

Neste trabalho foram realizados 19 testes, dos quais 4 identificaram falhas, neste caso o cálculo ficaria:

$$TF = (19 / 4) * 100 = 21\%$$

A taxa de falha resultou em 21% o que indica que há espaço para aprimoramento do código. É importante analisar a qualidade, ou seja, analisar quais tipos de falhas estão ocorrendo, pode também ser necessário incluir testes mais rigorosos. De toda forma, a melhoria contínua é essencial para reduzir essa taxa de falha e aumentar a satisfação do usuário.

Numero de testes vs Taxa de falha

