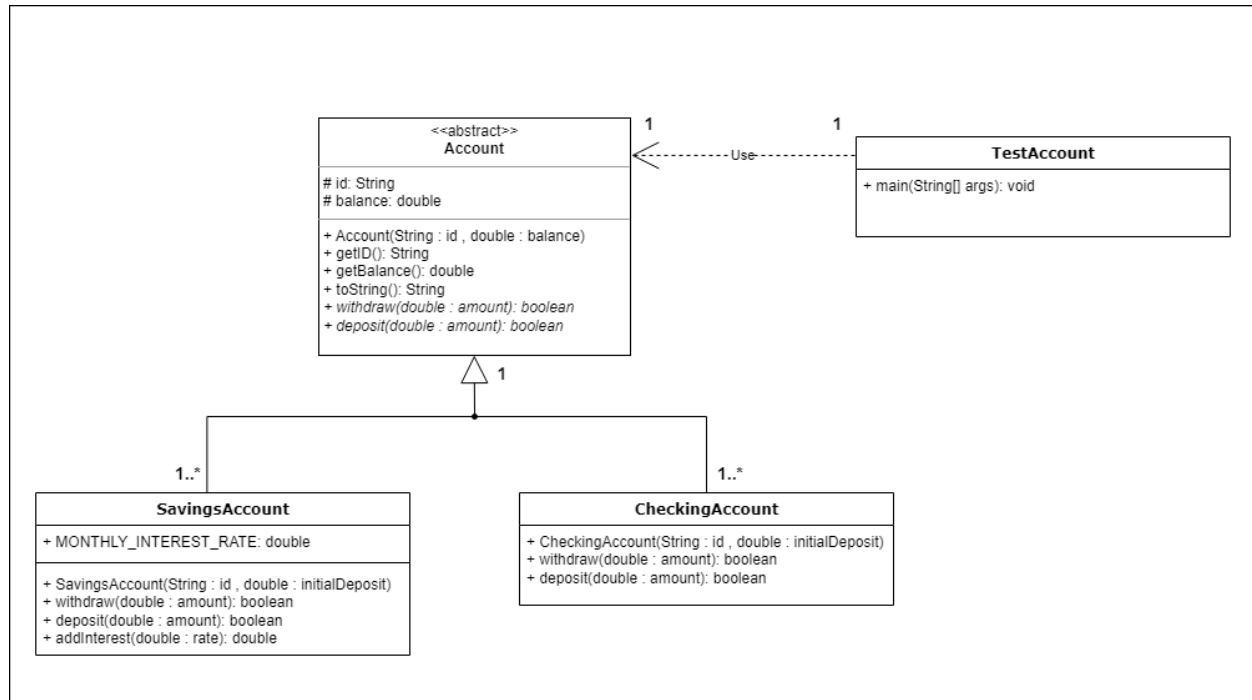


UML diagram



Explanation:

Account Class:

An abstract class with fields 'id' and 'balance'.

Contains an abstract withdraw method and an abstract deposit method.

Implements common methods like getID(), getBalance(), and toString().

SavingsAccount Class:

Extends Account.

Has constants MONTHLY_INTEREST_RATE.

Implements withdraw() and deposit() methods specific to SavingsAccount.

Adds addInterest() method.

CheckingAccount Class:

Extends Account.

Implements withdraw() and deposit() methods specific to CheckingAccount.

TestAccount Class:

Contains the main() method to run the application.

Test cases

This table includes the purpose of each test case, the expected result, and how exceptions are handled.

Test Case	Purpose	Expected Result
TC1: Valid Deposit in Savings Account	Ensure valid deposit works correctly	Deposit succeeds, balance increases
TC2: Valid Deposit in Checking Account	Ensure valid deposit works correctly	Deposit succeeds, balance increases
TC3: Valid Withdrawal in Savings Account	Ensure valid withdrawal works correctly and respects the minimum balance	Withdrawal succeeds if balance remains above £10, else fails
TC4: Valid Withdrawal in Checking Account	Ensure valid withdrawal works correctly and does not go negative	Withdrawal succeeds if balance remains non-negative, else fails
TC5: Invalid Numeric Input for Menu Choice	Ensure non-numeric input for menu choice is handled	Error message displayed: "Invalid input. Please enter a numeric value."
TC6: Invalid Menu Choice (Out of Range)	Ensure invalid numeric input for menu choice is handled	Error message displayed: "Invalid choice. Please enter a number between 1 and 4."
TC7: Invalid Account Type (Non-numeric)	Ensure non-numeric input for account type is handled	Error message displayed: "Invalid input. Please enter a numeric value."
TC8: Invalid Account Type (Out of Range)	Ensure account type outside 1 or 2 is handled	Error message displayed: "Invalid account type."
TC9: Negative Deposit Amount	Ensure negative deposit amount is handled	Error message displayed: "Negative value entered."
TC10: Negative Withdrawal Amount	Ensure negative withdrawal amount is handled	Error message displayed: "Negative value entered."

TC11: Invalid Numeric Input for Deposit Amount	Ensure non-numeric input for deposit amount is handled	Error message displayed: "Invalid input. Please enter a numeric value."
TC12: Invalid Numeric Input for Withdrawal Amount	Ensure non-numeric input for withdrawal amount is handled	Error message displayed: "Invalid input. Please enter a numeric value."
TC13: View Customer Details	Ensure customer details are displayed correctly	Customer details for both accounts displayed correctly
TC14: Add Interest	Ensure add interest is only applicable for Savings Account	Account balance is updated with interest for Savings Account; no action for Checking Account
TC15: Exit Program	Ensure program exits gracefully	Program exits with message: "Thank you for using bank App.!"

Explanation of Exceptions and Results

- Valid Deposit and Withdrawal (TC1 - TC4):
Purpose: These tests ensure that valid operations on both account types work as expected, correctly increasing or decreasing the balance.
Expected Result: The balance should update appropriately, and the methods should return true.
- Invalid Numeric Input for Menu Choice (TC5):
Purpose: Ensures that when a user inputs a non-numeric value for the menu choice, the program handles the Error gracefully.
Expected Result: A NumberFormatException is caught, and an Error message is displayed, guiding the user to enter a numeric value.
- Invalid Menu Choice (Out of Range) (TC6):
Purpose: Ensures that entering a numeric value outside the valid range (1 to 5) is handled.
Expected Result: An Error message is displayed, guiding the user to enter a valid choice between 1 and 5.
- Invalid Account Type (Non-numeric) (TC7):
Purpose: Ensures that non-numeric input for account type is handled appropriately.
Expected Result: A NumberFormatException is caught, and an Error message is displayed, guiding the user to enter a numeric value.

- Invalid Account Type (Out of Range) (TC8):
Purpose: Ensures that numeric inputs outside the valid range for account type are handled.
Expected Result: An Error message is displayed, indicating that the account type is invalid.
- Negative Deposit and Withdrawal Amount (TC9 - TC10):
Purpose: Ensures that negative deposit or withdrawal amounts are handled.
Expected Result: An `IllegalArgumentException` is caught, and an Error message is displayed, indicating that a negative value was entered.
- Invalid Numeric Input for Deposit and Withdrawal Amount (TC11 - TC12):
Purpose: Ensures that non-numeric input for deposit or withdrawal amount is handled.
Expected Result: A `NumberFormatException` is caught, and an Error message is displayed, guiding the user to enter a numeric value.
- View Customer Details (TC13):
Purpose: Ensures that customer details are displayed correctly for both accounts.
Expected Result: The details (ID and balance) of both accounts are displayed correctly.
- Add Interest to Savings Account (TC14):
Purpose: Ensures that interest is correctly added to the Savings Account.
Expected Result: The balance of the Savings Account is updated with the calculated interest, and the new balance is displayed.
- Exit Program (TC15):
Purpose: Ensures that the program exits gracefully.
Expected Result: The program exits with a thank-you message, confirming a proper exit.