1. Develop a simple banking system that allows users to create accounts, deposit money, withdraw money, and check balance. Implement methods for account creation, deposit, withdrawal, and balance inquiry.

**Methods**:

* createAccount(String accountHolderName, double initialDeposit)
* depositMoney(String accountNumber, double amount)
* withdrawMoney(String accountNumber, double amount)
* checkBalance(String accountNumber)

SOLUTION:

import java.util.HashMap;

import java.util.Map;

class BankAccount {

private String accountNumber;

private String accountHolderName;

private double balance;

public BankAccount(String accountNumber, String accountHolderName, double initialDeposit) {

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.balance = initialDeposit;

}

public String getAccountNumber() {

return accountNumber;

}

public String getAccountHolderName() {

return accountHolderName;

}

public double getBalance() {

return balance;

}

public void deposit(double amount) {

balance += amount;

System.out.println(amount + " deposited successfully.");

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println(amount + " withdrawn successfully.");

} else {

System.out.println("Insufficient balance.");

}

}

}

class Bank {

private Map<String, BankAccount> accounts;

public Bank() {

accounts = new HashMap<>();

}

public void createAccount(String accountHolderName, double initialDeposit) {

String accountNumber = "ACC" + (accounts.size() + 1);

BankAccount newAccount = new BankAccount(accountNumber, accountHolderName, initialDeposit);

accounts.put(accountNumber, newAccount);

System.out.println("Account created successfully. Account number: " + accountNumber);

}

public void depositMoney(String accountNumber, double amount) {

BankAccount account = accounts.get(accountNumber);

if (account != null) {

account.deposit(amount);

} else {

System.out.println("Account not found.");

}

}

public void withdrawMoney(String accountNumber, double amount) {

BankAccount account = accounts.get(accountNumber);

if (account != null) {

account.withdraw(amount);

} else {

System.out.println("Account not found.");

}

}

public void checkBalance(String accountNumber) {

BankAccount account = accounts.get(accountNumber);

if (account != null) {

System.out.println("Account Holder Name: " + account.getAccountHolderName());

System.out.println("Account Number: " + account.getAccountNumber());

System.out.println("Balance: " + account.getBalance());

} else {

System.out.println("Account not found.");

}

}

}

public class Main {

public static void main(String[] args) {

Bank bank = new Bank();

bank.createAccount("John Doe", 1000);

bank.createAccount("Jane Smith", 500);

bank.depositMoney("ACC1", 500);

bank.depositMoney("ACC2", 200);

bank.withdrawMoney("ACC1", 200);

bank.withdrawMoney("ACC2", 100);

bank.checkBalance("ACC1");

bank.checkBalance("ACC2");

bank.depositMoney("ACC3", 100);

}

}

2. Create an expense tracker that allows users to add expenses, categorize them, and view a summary report. Implement methods to add expenses, categorize expenses, and generate reports.

**Methods**:

* addExpense(String description, double amount, String category)
* viewExpensesByCategory(String category)
* generateExpenseReport()

SOLUTION:

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class Expense {

private String description;

private double amount;

private String category;

public Expense(String description, double amount, String category) {

this.description = description;

this.amount = amount;

this.category = category;

}

public String getDescription() {

return description;

}

public double getAmount() {

return amount;

}

public String getCategory() {

return category;

}

}

class ExpenseTracker {

private List<Expense> expenses;

private Map<String, List<Expense>> categorizedExpenses;

public ExpenseTracker() {

expenses = new ArrayList<>();

categorizedExpenses = new HashMap<>();

}

public void addExpense(String description, double amount, String category) {

Expense newExpense = new Expense(description, amount, category);

expenses.add(newExpense);

if (categorizedExpenses.containsKey(category)) {

categorizedExpenses.get(category).add(newExpense);

} else {

List<Expense> categoryList = new ArrayList<>();

categoryList.add(newExpense);

categorizedExpenses.put(category, categoryList);

}

System.out.println("Expense added successfully.");

}

public void viewExpensesByCategory(String category) {

if (categorizedExpenses.containsKey(category)) {

List<Expense> expensesInCategory = categorizedExpenses.get(category);

System.out.println("Expenses in category '" + category + "':");

for (Expense expense : expensesInCategory) {

System.out.println("Description: " + expense.getDescription() +

", Amount: " + expense.getAmount());

}

} else {

System.out.println("No expenses found in category '" + category + "'.");

}

}

public void generateExpenseReport() {

System.out.println("Expense Report:");

for (Expense expense : expenses) {

System.out.println("Category: " + expense.getCategory() +

", Description: " + expense.getDescription() +

", Amount: " + expense.getAmount());

}

}

}

public class Main {

public static void main(String[] args) {

ExpenseTracker tracker = new ExpenseTracker();

tracker.addExpense("Groceries", 50.75, "Food");

tracker.addExpense("Dinner with friends", 80.50, "Entertainment");

tracker.addExpense("Gasoline", 40.25, "Transportation");

tracker.addExpense("Books", 30.00, "Education");

tracker.addExpense("Movie tickets", 25.50, "Entertainment");

tracker.viewExpensesByCategory("Entertainment");

tracker.generateExpenseReport();

}

}