# Currency Converter

import java.math.BigDecimal;  
import java.math.RoundingMode;  
import java.util.EnumMap;  
import java.util.Map;  
import java.util.Scanner;  
  
// Currency enum  
enum Currency {  
 *USD*("United States Dollar"),  
 *EUR*("Euro"),  
 *GBP*("British Pound"),  
 *JPY*("Japanese Yen"),  
 *AUD*("Australian Dollar"),  
 *CAD*("Canadian Dollar"),  
 *CHF*("Swiss Franc");  
  
 private final String fullName;  
  
 Currency(String fullName) {  
 this.fullName = fullName;  
 }  
  
 public String getFullName() {  
 return fullName;  
 }  
}  
  
// Currency Converter Class  
public class CurrencyConverter {  
 private static final Map<Currency, BigDecimal> *exchangeRates* = new EnumMap<>(Currency.class);  
  
 static {  
 // Exchange rates relative to USD  
 *exchangeRates*.put(Currency.*USD*, BigDecimal.*ONE*);  
 *exchangeRates*.put(Currency.*EUR*, new BigDecimal("0.92"));  
 *exchangeRates*.put(Currency.*GBP*, new BigDecimal("0.79"));  
 *exchangeRates*.put(Currency.*JPY*, new BigDecimal("149.50"));  
 *exchangeRates*.put(Currency.*AUD*, new BigDecimal("1.52"));  
 *exchangeRates*.put(Currency.*CAD*, new BigDecimal("1.35"));  
 *exchangeRates*.put(Currency.*CHF*, new BigDecimal("0.88"));  
 }  
  
 // Method to convert between two currencies  
 public static BigDecimal convert(BigDecimal amount, Currency from, Currency to) {  
 if (amount.compareTo(BigDecimal.*ZERO*) < 0) {  
 throw new IllegalArgumentException("Amount must be non-negative.");  
 }  
  
 BigDecimal fromRate = *exchangeRates*.getOrDefault(from, BigDecimal.*ONE*);  
 BigDecimal toRate = *exchangeRates*.getOrDefault(to, BigDecimal.*ONE*);  
  
 // Conversion: amount in USD \* targetRate / baseRate  
 BigDecimal amountInUSD = amount.divide(fromRate, 6, RoundingMode.*HALF\_UP*);  
 return amountInUSD.multiply(toRate).setScale(2, RoundingMode.*HALF\_UP*);  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Print supported currencies  
 System.*out*.println("Supported Currencies:");  
 for (Currency currency : Currency.*values*()) {  
 System.*out*.printf("%s - %s%n", currency.name(), currency.getFullName());  
 }  
  
 try {  
 // Input: amount to convert  
 System.*out*.print("\nEnter the amount to convert: ");  
 BigDecimal amount = scanner.nextBigDecimal();  
  
 // Input: source currency  
 System.*out*.print("Enter source currency code (e.g., USD): ");  
 Currency fromCurrency = Currency.*valueOf*(scanner.next().toUpperCase());  
  
 // Input: target currency  
 System.*out*.print("Enter target currency code (e.g., EUR): ");  
 Currency toCurrency = Currency.*valueOf*(scanner.next().toUpperCase());  
  
 // Perform conversion  
 BigDecimal convertedAmount = *convert*(amount, fromCurrency, toCurrency);  
  
 // Output result  
 System.*out*.printf("\n%.2f %s = %.2f %s%n",  
 amount, fromCurrency.name(),  
 convertedAmount, toCurrency.name()  
 );  
 } catch (IllegalArgumentException e) {  
 System.*err*.println("Error: Invalid input. Please enter valid currency codes.");  
 } finally {  
 scanner.close();  
 }  
 }  
}

# Bar Graph Showing Rates

import javax.swing.\*;  
import java.awt.\*;  
import java.math.BigDecimal;  
import java.util.Map;  
  
public class BarGraph extends JPanel {  
  
 private final Map<String, BigDecimal> exchangeRates;  
  
 public BarGraph(Map<String, BigDecimal> exchangeRates) {  
 this.exchangeRates = exchangeRates;  
 }  
  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
  
 // Cast Graphics to Graphics2D for better control  
 Graphics2D g2d = (Graphics2D) g;  
  
 // Enable anti-aliasing for smoother graphics  
 g2d.setRenderingHint(RenderingHints.*KEY\_ANTIALIASING*, RenderingHints.*VALUE\_ANTIALIAS\_ON*);  
  
 // Set up the axis labels  
 g2d.drawString("Currency Exchange Rates (USD Base)", 10, 20);  
  
 // Set up the bar width and spacing  
 int barWidth = 50;  
 int spaceBetweenBars = 20;  
 int startX = 50;  
  
 // Find the highest exchange rate to scale the bars  
 BigDecimal maxRate = exchangeRates.values().stream().max(BigDecimal::compareTo).orElse(BigDecimal.*ONE*);  
 int graphHeight = 300; // Height of the graph area  
 int offsetY = 30; // Start drawing below this offset  
  
 // Draw each bar  
 int i = 0;  
 for (Map.Entry<String, BigDecimal> entry : exchangeRates.entrySet()) {  
 String currency = entry.getKey();  
 BigDecimal rate = entry.getValue();  
  
 // Calculate the height of the bar (scaled relative to the highest exchange rate)  
 int barHeight = (int) (rate.doubleValue() / maxRate.doubleValue() \* graphHeight);  
  
 // Set color for the bars  
 g2d.setColor(new Color(100, 150, 255));  
  
 // Draw the bar  
 g2d.fillRect(startX + (i \* (barWidth + spaceBetweenBars)), graphHeight + offsetY - barHeight, barWidth, barHeight);  
  
 // Set color for the text (values on top of bars)  
 g2d.setColor(Color.*BLACK*);  
  
 // Display the currency value on top of the bar  
 g2d.drawString(rate.toString(), startX + (i \* (barWidth + spaceBetweenBars)) + 5, graphHeight + offsetY - barHeight - 5);  
  
 // Display the currency code below the bar  
 g2d.drawString(currency, startX + (i \* (barWidth + spaceBetweenBars)) + 5, graphHeight + offsetY + 15);  
  
 i++;  
 }  
 }  
  
 public static void main(String[] args) {  
 // Sample exchange rates data (relative to USD)  
 Map<String, BigDecimal> exchangeRates = Map.*of*(  
 "USD", BigDecimal.*ONE*,  
 "EUR", new BigDecimal("0.92"),  
 "GBP", new BigDecimal("0.79"),  
 "JPY", new BigDecimal("149.50"),  
 "AUD", new BigDecimal("1.52"),  
 "CAD", new BigDecimal("1.35"),  
 "CHF", new BigDecimal("0.88")  
 );  
  
 // Create and display the bar graph  
 JFrame frame = new JFrame("Currency Exchange Rates");  
 BarGraph barGraph = new BarGraph(exchangeRates);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.getContentPane().add(barGraph);  
 frame.setSize(800, 500);  
 frame.setVisible(true);  
 }  
}