**Time Calculation for any Time Zone**

CODING:

import java.time.LocalDateTime;

import java.time.ZoneId;

import java.time.ZonedDateTime;

import java.time.format.DateTimeFormatter;

import java.util.Scanner;

class BaseTimeCalculator {

private DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss z");

protected DateTimeFormatter getFormatter() {

return formatter;

}

public String getCurrentTime() {

LocalDateTime currentTime = LocalDateTime.now();

ZonedDateTime zonedDateTime = currentTime.atZone(ZoneId.systemDefault());

return zonedDateTime.format(getFormatter());

}

}

class TimeZoneCalculator extends BaseTimeCalculator {

public String getCurrentTime(ZoneId zoneId) {

LocalDateTime currentTime = LocalDateTime.now();

ZonedDateTime zonedDateTime = currentTime.atZone(zoneId);

return zonedDateTime.format(getFormatter());

}

public String getCurrentTime(String zoneId) {

try {

ZoneId validZoneId = ZoneId.of(zoneId);

return getCurrentTime(validZoneId);

} catch (Exception e) {

return "Invalid time zone: " + zoneId;

}

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

TimeZoneCalculator calculator = new TimeZoneCalculator();

System.out.println("Current time in default time zone: " + calculator.getCurrentTime());

System.out.print("Enter a time zone (e.g., Asia/Tokyo, America/New\_York): ");

String userTimeZone = scanner.nextLine();

System.out.println("Current time in " + userTimeZone + ": " + calculator.getCurrentTime(userTimeZone));

scanner.close();

}

}

OUTPUT:

GITHUB LINK:

https://github.com/GOPIKA2308/JAVA-MICROPROJECT.git