

Programming Fundamentals II

ปฏิบัติการ 3

ข้อปฏิบัติ

1. ให้ส่งงานในช่องทางที่ผู้สอนกำหนด
2. ไม่ต้องทำตามลำดับ ถ้าเห็นว่าทำข้อไหนได้ก่อนก็ทำข้อนั้นก่อนได้
3. ปรีกษาเพื่อนได้ แต่ต้องทำงานด้วยตนเองหรือภายในกลุ่มตามที่กำหนด

1. สร้างคลาส **Vector** เพื่อใช้สร้างอ็อบเจกต์แทนเวกเตอร์จากค่าขนาดในแกน x และแกน y

คลาส **Vector** มีเมทอดดังนี้

`public Vector(double x, double y)` เป็น constructor มีพารามิเตอร์เป็นขนาดในแกน x และแกน y

`public double getLength()` คืนค่ากลับเป็นความยาวหรือขนาดของเวกเตอร์ โดยคำนวณจาก

$$\|v\| = \sqrt{x^2 + y^2}$$

`public double getDirection()` คืนค่ากลับเป็นทิศหรือมุมของเวกเตอร์ มีค่าระหว่าง $-\pi$ ถึง π โดยคำนวณจาก

$$\theta = \tan^{-1} \frac{y}{x}$$

`public double dot(Vector v)` คืนค่ากลับเป็นผลคูณ dot product ระหว่างอ็อบเจกต์ **Vector** นี้กับ **Vector v** ซึ่งสามารถคำนวณได้จาก

$$u \cdot v = x_u x_v + y_u y_v$$

`public Vector getUnitVector()` คืนค่ากลับเป็นอ็อบเจกต์ **Vector** ขนาดหนึ่งหน่วยที่มีทิศเดียวกับอ็อบเจกต์ **Vector** นี้ โดยคำนวณจาก

$$v = \frac{v}{\|v\|}$$

`public void print()` แสดงค่าของเวกเตอร์ทางจอภาพ โดยมีรูปแบบเป็น **Vector: x.xxj + y.yyj** พร้อมขึ้นบรรทัดใหม่ เช่น **Vector: 1.00i + 2.50j**

โจทย์กำหนดคลาส **Main** ซึ่งมีเมทอด `main()` ที่จะเรียกใช้คลาส **Vector** เอาไว้ให้ในไฟล์ **Main.java** ผลลัพธ์เมื่อรันควรจะเป็นดังนี้

```
Vector: 1.00i + 1.00j
Vector: -0.50i + 0.50j
v: length = 0.71, direction = 2.36
u . v = 2.00
Vector: 0.71i + 0.71j
```

ข้อแนะนำ

- คลาส `Math` มีเมทอดทางคณิตศาสตร์ให้ใช้หลายตัว เช่น `sqrt()`, `abs()`, `cos()` เป็นต้น เช่น ถ้าต้องการหารากที่สองของ `x` ก็สามารถทำได้โดยเรียก `Math.sqrt(x)` นิสิตสามารถค้นรายละเอียดเพิ่มเติมในการใช้งานได้
- เมทอด `atan2(y, x)` สามารถใช้คำนวณ `arctan` เพื่อหามุมจากด้านตรงข้ามและด้านประชิดได้ โดยผลลัพธ์จะอยู่ในช่วงระหว่าง $-\pi$ ถึง π
- ในบางเมทอดอาจต้องใช้ `this` เพื่ออ้างอิงตัวเอง

2. ปรับปรุงคลาส Account

ใช้ไฟล์ `Account.java` ที่โจทย์ให้ไว้เป็นจุดตั้งต้น ต้องการปรับปรุงคลาส `Account` ให้สามารถรองรับการระงับการใช้งานบัญชีได้ โดยบัญชีที่ถูกระงับจะไม่สามารถฝากหรือถอนได้จนกว่าจะเปิดใช้งานอีกครั้ง

ให้เพิ่มเมทอดต่อไปนี้ในคลาส `Account`

`public void suspend()` ระงับการใช้ กำหนดสถานะบัญชีให้เป็นถูกระงับหรือ `suspended`

`public void reactivate()` เปิดใช้งานอีกครั้ง กำหนดสถานะบัญชีให้กลับเป็นปกติหรือ `active`

`public boolean isActive()` ตรวจสอบสถานะบัญชีว่าปกติหรือไม่ ถ้าปกติจะคืนค่ากลับเป็น `true` ถ้าถูกระงับอยู่จะคืนค่ากลับเป็น `false`

และเพิ่มการตรวจสอบดังนี้

ถ้าบัญชีถูกระงับอยู่ จะไม่อนุญาตให้ฝากหรือถอน ถ้ามีการเรียกใช้ `deposit()` หรือ `withdraw()` ในขณะที่บัญชีถูกระงับให้สิ่งให้เกิด exception ชนิด `IllegalStateException` โดยให้ข้อความแจ้งไปในตัว exception ว่า `Account is suspended.`

หมายเหตุ: เราจะใช้ `IllegalStateException` เมื่อเกิดการทำงานในสถานะที่ไม่ได้รับอนุญาตให้ทำงานนั้น

โจทย์กำหนดคลาส `MainApp` ซึ่งมีเมทอด `main()` ที่จะเรียกใช้คลาส `Account` เอาไว้ให้ในไฟล์ `MainApp.java` ผลลัพธ์เมื่อรันควรจะเป็นดังนี้

```
Account 1234-0001 has been created with a balance of 100.00 baht.
Account 1234-0001 now has a balance of 200.00 baht.
Account is suspended.
Trying to deposit...
Illegal state: Account is suspended.
Trying to withdraw...
```

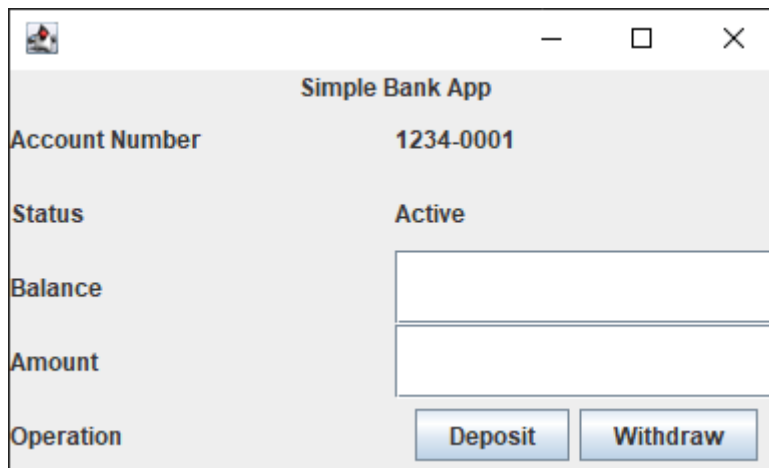
```
Illegal state: Account is suspended.  
Account is active.  
Account 1234-0001 now has a balance of 200.00 baht.  
Account 1234-0001 now has a balance of 300.00 baht.
```

ข้อแนะนำ

- พิจารณากำหนด instance variable เพิ่มเติมเพื่อเก็บสถานะของบัญชี

3. ปรับแต่ง GUI ให้กับ SimpleBankApp

ใช้ไฟล์ SimpleBankApp.java ที่เจตยให้ไว้เป็นจุดเริ่มต้น แก้ไขคลาส SimpleBankApp ให้แสดงผลดังนี้



โดยมีข้อกำหนดต่อไปนี้

- ขนาดของหน้าต่างคือ 400x240
- หน้าต่างควรปรากฏขึ้นกลางจอภาพ
- ปุ่มและกล่องข้อความต่าง ๆ ยังไม่ต้องการตอบสนอง

การบ้าน

1. สร้างคลาส PolarVector และปรับปรุงคลาส Vector

สืบเนื่องจากปฏิบัติการ 3 ข้อ 1 ให้สร้างคลาส PolarVector เพิ่มเติม เพื่อแทนเวกเตอร์ที่กำหนดด้วยพิกัดเชิงมุม โดยการสร้างจะต้องระบุความยาว (length) และทิศ (direction) โดยระบุเป็นมุมมีหน่วยเป็น radian (มีค่าระหว่าง $-\pi$ ถึง π)

คลาส PolarVector มีเมทอดดังนี้

`public PolarVector(double length, double direction)` เป็น constructor มีพารามิเตอร์เป็นขนาดและทิศในหน่วย radian (มีค่าระหว่าง $-\pi$ ถึง π)

`public double getProjectionX()` คืนค่ากลับเป็นขนาดของเวกเตอร์บนแกน x (หรือการทอดเวกเตอร์บนแกน x) โดยคำนวณจาก

$$x = r \cos \theta$$

`public double getProjectionY()` คืนค่ากลับเป็นขนาดของเวกเตอร์บนแกน y (หรือการทอดเวกเตอร์บนแกน y) โดยคำนวณจาก

$$y = r \sin \theta$$

`public Vector toVector()` คืนค่ากลับเป็นอ็อบเจกต์ **Vector** ที่สร้างจากขนาดในแกน x และแกน y ของอ็อบเจกต์ **PolarVector** นี้

`public void print()` แสดงค่าของเวกเตอร์ทางจอภาพ โดยมีรูปแบบเป็น **PolarVector: $r.r<a.aa$** (r คือ ความยาว a คือทิศ) พร้อมขึ้นบรรทัดใหม่ เช่น **PolarVector: 0.50<1.57**

และปรับปรุงคลาส **Vector** โดยเพิ่มเติมเมทอด

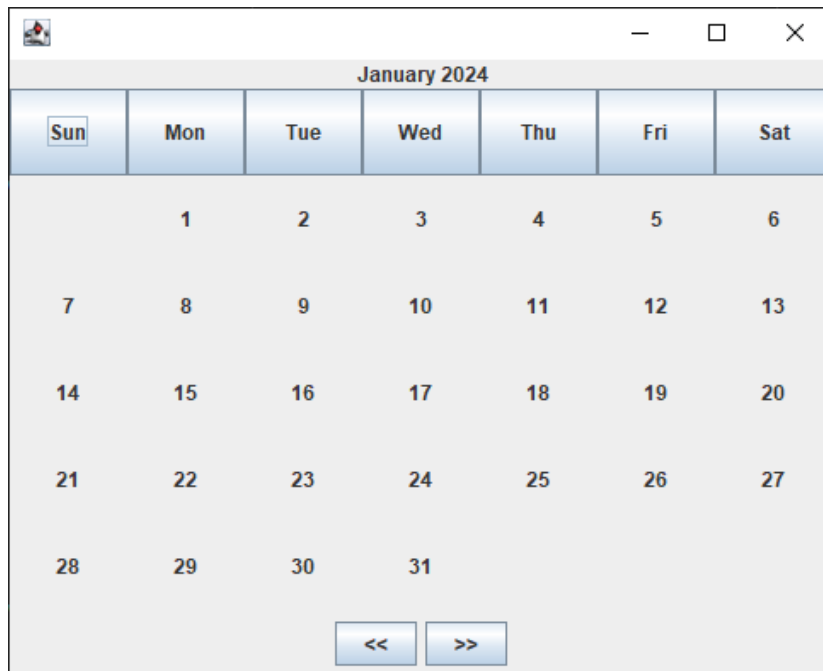
`public PolarVector toPolarVector()` คืนค่ากลับเป็นอ็อบเจกต์ **PolarVector** ที่สร้างจากพิกัดเชิงมุม (ความยาวและทิศ) ของอ็อบเจกต์ **Vector** นี้

โจทย์กำหนดคลาส **Main** ซึ่งมีเมทอด `main()` ที่จะเรียกใช้คลาส **Vector** เอาไว้ให้ในไฟล์ **Main.java** ผลลัพธ์เมื่อรันควรจะเป็นดังนี้

```
Vector: 1.50i + 0.50j
v: length = 1.58, direction = 0.32
PolarVector: 1.58<0.32
p: x = 1.50, y = 0.50
The next two lines should be equal:
Vector: 1.50i + 0.50j
Vector: 1.50i + 0.50j
```

2. สร้าง GUI สำหรับ CalendarApp

สร้างคลาส **CalendarApp** ให้สร้างหน้า GUI สำหรับปฏิทินเดือนมกราคม ค.ศ. 2024 ตามตัวอย่างด้านล่างนี้



ข้อกำหนด

- หน้าต่างควรปรากฏขึ้นกลางจอภาพ
- ขนาดของหน้าต่างในตัวอย่างนี้คือ 500x400 จุด แต่นิสิตสามารถสร้างให้มีขนาดแตกต่างจากนี้เพื่อความสวยงามหรือจะใช้ `pack()` ก็ได้
- ปุ่มต่าง ๆ ในหน้าต่างยังไม่ต้องมีการตอบสนอง