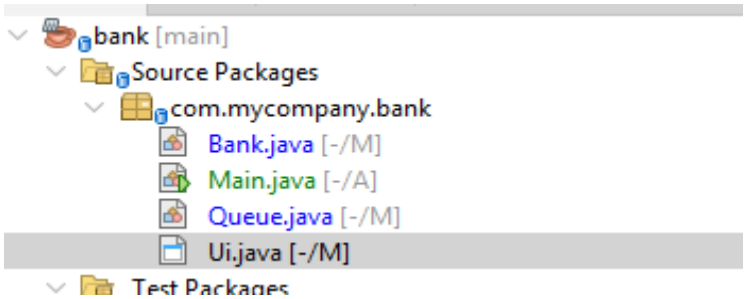
เอกสารอธิบายโปรแกรม Bank

Description

ระบบจัดการคิวธนาคารพัฒนาขึ้นด้วย Java โดยโครงสร้างสถาปัตยกรรมภายในถูกออกแบบขึ้นตามหลักการเขียน Object Oriented Programming และการใช้ โครงสร้างข้อมูลเชิงเส้น เพื่อควบคุมและจัดการออบเจกต์ให้มีประสิทธิภาพสูงสุด

โครงสร้างของโปรแกรม



bank/

└── com.mycompany.bank/

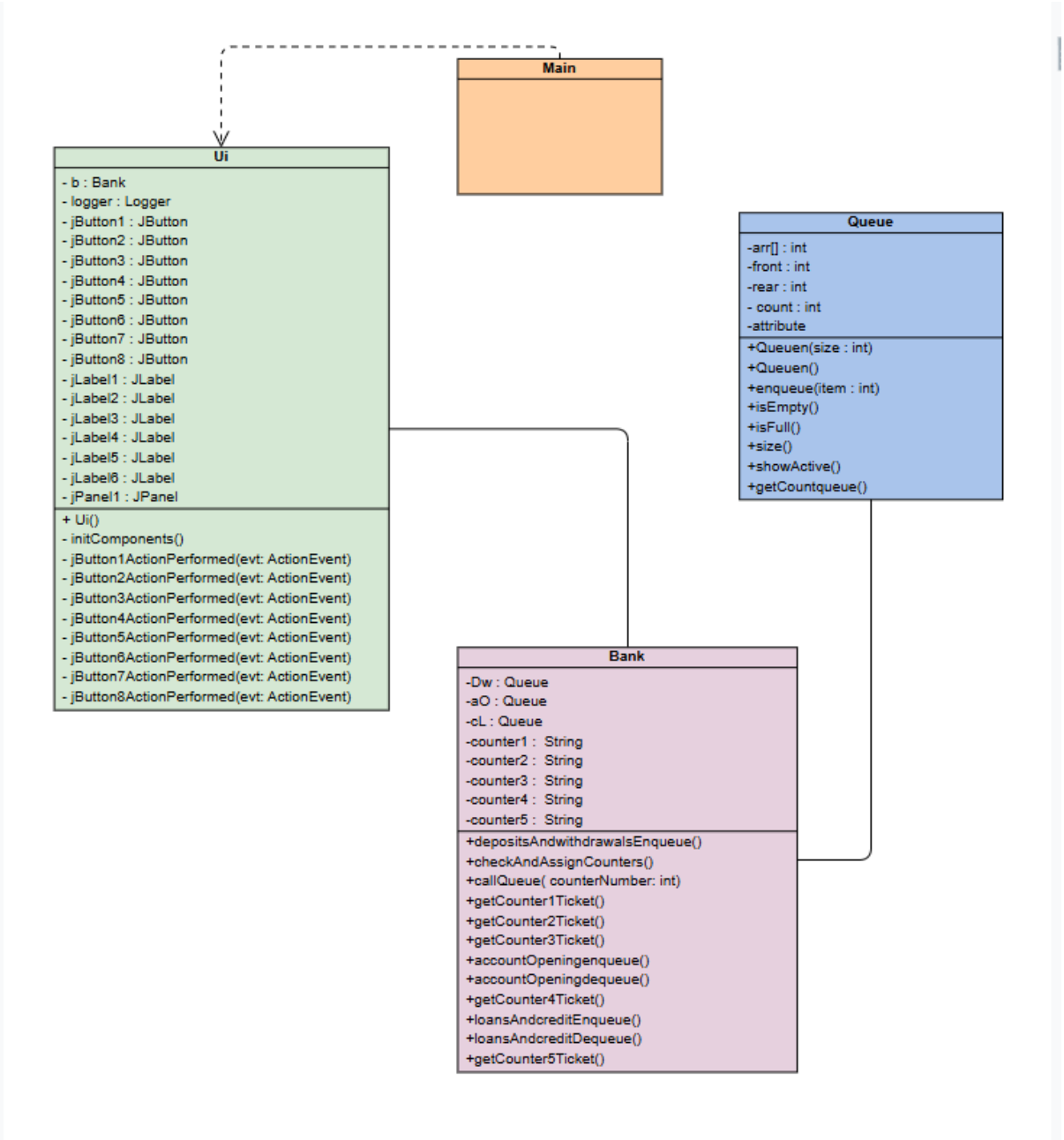
├── Main.java = จุดเริ่มต้นการทำงานของโปรแกรม

├── Bank.java = คลาสบริหารจัดการคิวและเคาน์เตอร์บริการของธนาคาร

├── Queue.java = โครงสร้างข้อมูลคิวจัดการลำดับข้อมูลด้วยอาร์เรย์

└── Ui.java = คลาสแสดงผลหน้าต่าง GUI และรับคำสั่งจากการกดปุ่มของผู้ใช้

Unified Modeling Language โดยใช้ Class Diagram



ความสัมพันธ์ของโปรแกรม bank

1. ความสัมพันธ์ระหว่าง Main กับ Uiความหมาย: คลาส Main ทำหน้าที่เป็นคลาสหลักของโปรแกรม เมื่อรันโปรแกรม Main จะทำการเรียกสร้างออบเจกต์ของ Ui สร้างหน้าต่าง GUI ขึ้นมาเพื่อแสดงผลให้ผู้ใช้งานเห็น

2. ความสัมพันธ์ระหว่าง Ui กับ Bankความหมาย: คลาส Ui มีการประกาศตัวแปรเพื่อเก็บออบเจกต์ของ Bank ไว้ภายในเพื่อใช้ส่งคำสั่งเมื่อมีการกดปุ่มต่างๆ บนหน้าจอไปให้ระบบธนาคาร

3. ความสัมพันธ์ระหว่าง Bank กับ Queueความหมาย: คลาส Bank มีการเรียกใช้โครงสร้างข้อมูลคิว โดยสร้างตัวแปรออบเจกต์จากคลาส Queue ไว้เป็นส่วนประกอบเพื่อรับหน้าที่จัดการคิวของแต่ละแผนก

หลักการทำงานของ Circular Queue

Circular Queue ยังคงยึดหลักการ **FIFO** แต่จะนำส่วนปลายของอาร์เรย์มาเชื่อมต่อกับส่วนหัว ทำให้เมื่อใส่ข้อมูลจนสุดปลายอาร์เรย์แล้ว ระบบสามารถวนกลับมาใช้พื้นที่ว่างที่ช่องแรกสุดของอาร์เรย์ได้

1. **การเพิ่มข้อมูล (Enqueue):** จะใช้สมการ rear = (rear + 1) % arr.length เพื่อหาตำแหน่งถัดไปที่จะใส่ข้อมูล หาก rear อยู่ที่ตำแหน่งสุดท้าย การหารเอาเศษด้วยขนาดของอาร์เรย์จะทำให้ rear มีค่ากลับมาเป็น 0 และนำข้อมูลไปใส่ที่ช่องแรกได้
2. **การดึงข้อมูล (Dequeue):** เมื่อดึงข้อมูลออก ตำแหน่งคิวแรกสุดจะต้องขยับตาม โดยใช้สมการ front = (front + 1) % arr.length เพื่อให้ตัวชี้ขยับวนเป็นวงกลมในทิศทางเดียวกันกับ rear

Source Code โปรแกรม Bank

1.Main.java = จุดเริ่มต้นการทำงานของโปรแกรม

~ / Main.java [fullscreen]

```java
package com.mycompany.bank;

public class Main {

    public static void main(String[] args) {
        java.awt.EventQueue.invokeLater(() -> {
            Ui myWindow = new Ui();
            myWindow.setVisible(true);
        });
    }
}
```

2.Bank.java = คลาสบริหารจัดการคิวและเคาน์เตอร์บริการของธนาคาร

~ / Bank.java [fullscreen]

```java
package com.mycompany.bank;

public class Bank {

    private Queue Dw = new Queue(20);
    private Queue aO = new Queue(20);
    private Queue cL = new Queue(20);

    private String counter1 = "-";
    private String counter2 = "-";
    private String counter3 = "-";
    private String counter4 = "-";
    private String counter5 = "-";

    public void depositsAndwithdrawalsEnqueue() {
        if (Dw.getCountqueue() > 20) {
            System.out.println("Queue is full!");
        } else {
            int currentQueueNumber = Dw.getCountqueue();

            Dw.enqueue(currentQueueNumber);
            System.out.println("Adding to queue: " + currentQueueNumber);

            checkAndAssignCounters();
        }
    }

    private void checkAndAssignCounters() {
        if (counter1.equals("-") && !Dw.isEmpty()) {
            counter1 = "WD" + Dw.dequeue();
        }
        if (counter2.equals("-") && !Dw.isEmpty()) {
            counter2 = "WD" + Dw.dequeue();
        }
        if (counter3.equals("-") && !Dw.isEmpty()) {
            counter3 = "WD" + Dw.dequeue();
        }
        if (counter4.equals("-") && !aO.isEmpty()) {
            counter4 = "AO" + aO.dequeue();
        }
        if (counter5.equals("-") && !cL.isEmpty()) {
            counter5 = "CL" + cL.dequeue();
        }
    }

    public String callQueue(int counterNumber) {
        String newTicket = "-";

        if (!Dw.isEmpty()) {
            newTicket = "WD" + Dw.dequeue();
        }

        if (counterNumber == 1) {
            counter1 = newTicket;
        } else if (counterNumber == 2) {
            counter2 = newTicket;
        } else if (counterNumber == 3) {
            counter3 = newTicket;
        }

        return newTicket;
    }

    public String getCounter1Ticket() {
        return counter1;
```

```java
    public String getCounter2Ticket() {
        return counter2;
    }

    public String getCounter3Ticket() {
        return counter3;
    }

    public void accountOpeningEnqueue() {
        if (aO.getCountqueue() > 20) {
            System.out.println("Queue is full!");
        } else {
            int currentQueueNumber = aO.getCountqueue();

            aO.enqueue(currentQueueNumber);
            System.out.println("Adding to queue: " + currentQueueNumber);

            checkAndAssignCounters();
        }
    }

    public String accountOpeningdequeue() {
        String newTicket = "-";

        if (!aO.isEmpty()) {
            newTicket = "AO" + aO.dequeue();
        }

        counter4 = newTicket;

        return newTicket;
    }


    public String getCounter4Ticket() {
        return counter4;
    }

    public void loansAndcreditEnqueue() {
        if (cL.getCountqueue() > 20) {
            System.out.println("Queue is full!");
        } else {
            int currentQueueNumber = cL.getCountqueue();

            cL.enqueue(currentQueueNumber);
            System.out.println("Adding to queue: " + currentQueueNumber);

            checkAndAssignCounters();
        }
    }

    public String loansAndcreditDequeue() {
        String newTicket = "-";

        if (!cL.isEmpty()) {
            newTicket = "CL" + cL.dequeue();
        }

        counter5 = newTicket;

        return newTicket;
    }

    public String getCounter5Ticket() {
        return counter5;
    }

}
```

3. Queue.java = โครงสร้างข้อมูลคิวจัดการลำดับข้อมูลด้วยอาร์เรย์

~ / Queue.java [fullscreen]

```java
package com.mycompany.bank;

public class Queue {

    private int arr[];
    private int front, rear, count;
    private int countQueue = 1;

    public Queue() {
        this(0);
    }

    public Queue(int size) {
        this.arr = new int[size];
        this.count = 0;
        this.front = -1;
        this.rear = -1;
    }

    public void enqueue(int item) {
        if (!isFull()) {
            if (isEmpty()) {
                this.front = 0;
                this.rear = 0;
                arr[rear] = item;
                countQueue++;
                count++;
            } else {
                rear = (rear + 1) % arr.length;
                arr[rear] = item;
                countQueue++;
                count++;
            }
        } else {
            System.out.println("Queue is full");
        }
    }

    public int dequeue() {
        int data = -1;
        if (!isEmpty()) {
            data = arr[this.front];
            if (front == rear) {
                this.front = -1;
                this.rear = -1;
            } else {
                front = (front + 1) % arr.length;
            }
            count--;
        } else {
            System.out.print("empty");
        }
        return data;
    }

    public boolean isEmpty() {
        return this.front == -1 && this.rear == -1;
    }

    boolean isFull() {
        return count == arr.length;
    }

    int size() {
        return this.arr.length;
    }


    public void showActive() {
        System.out.print("active(" + count + ") =");
        for (int i = 0; i < count; i++) {
            System.out.print(" " + arr[(front + i) % arr.length]);
        }
        System.out.println();
    }

    public int getCountqueue() {
        return countQueue;
    }

}
```

4.Ui.java = คลาสแสดงผลหน้าต่าง GUI และรับคำสั่งจากการกดปุ่มของผู้ใช้

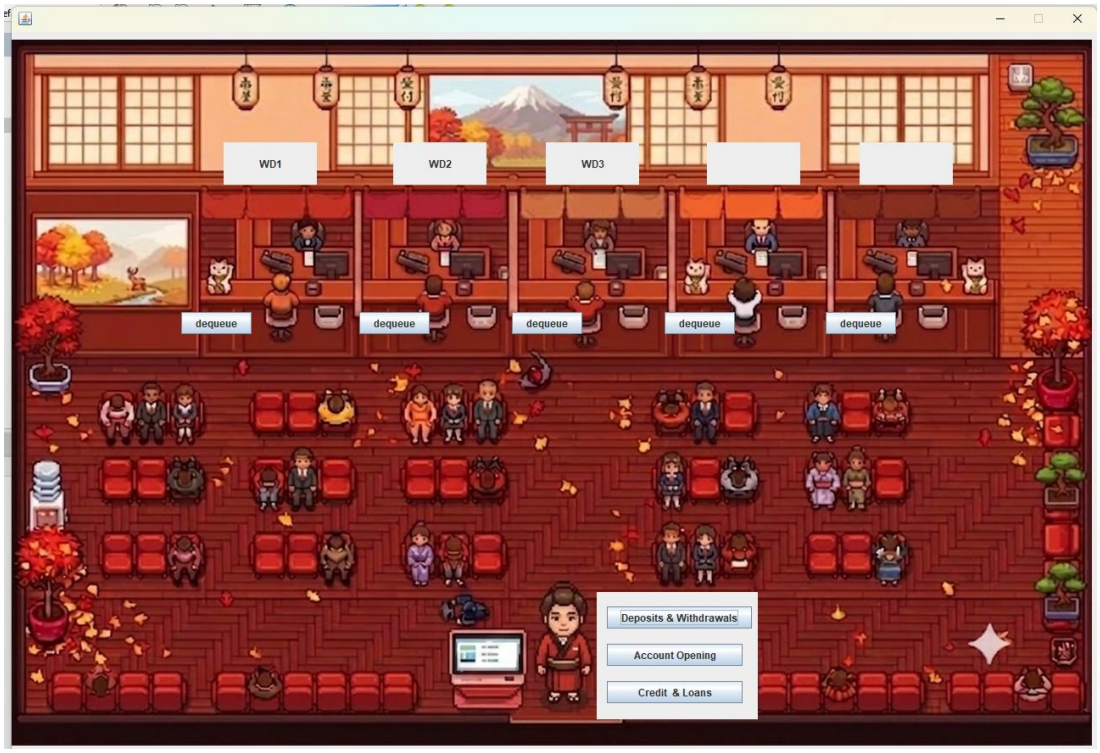
~ / Ui.java [fullscreen]

```java
package com.mycompany.bank;

public class Ui extends javax.swing.JFrame {

    private Bank b = new Bank();
    private static final java.util.logging.Logger logger = java.util.logging.Logger.getLogger(Ui.class.getName());

    public Ui() {
        initComponents();
    }

    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton8 = new javax.swing.JButton();
        jButton7 = new javax.swing.JButton();
        jButton6 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jButton4 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        jLabel6 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jButton8.setText("dequeue");
        jButton8.addActionListener(this::jButton8ActionPerformed);
        getContentPane().add(jButton8, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 330, -1, -1));

        jButton7.setText("dequeue");
        jButton7.addActionListener(this::jButton7ActionPerformed);
        getContentPane().add(jButton7, new org.netbeans.lib.awtextra.AbsoluteConstraints(770, 330, -1, -1));

        jButton6.setText("dequeue");
        jButton6.addActionListener(this::jButton6ActionPerformed);
        getContentPane().add(jButton6, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 330, -1, -1));

        jButton5.setText("dequeue");
        jButton5.addActionListener(this::jButton5ActionPerformed);
        getContentPane().add(jButton5, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 330, -1, -1));

        jButton2.setText("dequeue");
        jButton2.addActionListener(this::jButton2ActionPerformed);
        getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 330, -1, -1));

        jButton4.setText("Account Opening");
        jButton4.addActionListener(this::jButton4ActionPerformed);

        jButton3.setText("Credit  & Loans");
        jButton3.addActionListener(this::jButton3ActionPerformed);

        jButton1.setText("Deposits & Withdrawals");
        jButton1.addActionListener(this::jButton1ActionPerformed);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
                .addContainerGap(18, Short.MAX_VALUE)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 160, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 160, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1))
                .addGap(12, 12, 12))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(17, 17, 17)
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jButton4)
                .addGap(18, 18, 18)
                .addComponent(jButton3)
                .addContainerGap(28, Short.MAX_VALUE))
        );

        getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 660, 190, 150));

        jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel6.setName(""); // NOI18N
        jLabel6.setOpaque(true);
        getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(1000, 130, 110, 50));

        jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel5.setName(""); // NOI18N
        jLabel5.setOpaque(true);
        getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 130, 110, 50));

        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setName(""); // NOI18N
        jLabel4.setOpaque(true);
        getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 130, 110, 50));

        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel3.setName(""); // NOI18N
        jLabel3.setOpaque(true);
        getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 130, 110, 50));

        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setName(""); // NOI18N
        jLabel2.setOpaque(true);
        getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 130, 110, 50));

        jLabel1.setIcon(new javax.swing.ImageIcon("C:\\Users\\User\\Documents\\NetBeansProjects\\ENGSE216\\bank\\src\\main\\resourc
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1280, 850));

        pack();
    }// </editor-fold>
```

```java
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        b.depositsAndwithdrawalsEnqueue();
        System.out.println("Deposit and withdrawal queue button clicked!");

        jLabel2.setText(b.getCounter1Ticket());
        jLabel3.setText(b.getCounter2Ticket());
        jLabel4.setText(b.getCounter3Ticket());
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String ticket = b.callQueue(1);
        jLabel2.setText(ticket);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        String ticket = b.callQueue(2);
        jLabel3.setText(ticket);
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        String ticket = b.callQueue(3);
        jLabel4.setText(ticket);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        b.accountOpeningEnqueue();
        System.out.println("Account Opening queue button clicked!");
        jLabel5.setText(b.getCounter4Ticket());
    }

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        String ticket = b.accountOpeningdequeue();
        jLabel5.setText(ticket);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        b.loansAndcreditEnqueue();
        System.out.println("Loans & Credit queue button clicked!");
        jLabel6.setText(b.getCounter5Ticket());
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
       String ticket = b.loansAndcreditDequeue();
       jLabel6.setText(b.getCounter5Ticket());
    }


    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration
}
```

Result

การเพิ่มข้อมูล (Enqueue)




```
--- exec:3.5.1:exec (default-cli) @ bank ---
Adding to queue: 1
Deposit and withdrawal queue button clicked!
Adding to queue: 2
Deposit and withdrawal queue button clicked!
Adding to queue: 3
Deposit and withdrawal queue button clicked!
Adding to queue: 4
Deposit and withdrawal queue button clicked!
Adding to queue: 5
Deposit and withdrawal queue button clicked!
```

การดึงข้อมูล (Dequeue)