## SE 471 Software Architecture

# Lab 4

| Student Name | | Student CSUSM ID | Contribution percentage |
|---|---|---|---|
| 1 | EJ Lilagan | lilag002 | 33.33% |
| 2 | Dalynna Nguyen | nguye1051 | 33.33% |
| 3 | Neo Argatides | argat001 | 33.33% |

/60

## Grading Rubrics (for instructor only):

| Criteria | 1. Beginning | 2. Developing | 3. Proficient | 4. Exemplary |
|---|---|---|---|---|
| **Program: functionality** ***correctness*** | 0-9 | 10-14 | 15-19 | 20 |
| | | | | |
| **Program: functionality** ***Behavior Testing*** | 0-9 | 10-14 | 15-19 | 20 |
| | | | | |
| **Program: quality ->** ***Readability*** | 0-9 | 10-14 | 15-19 | 20 |
| | | | | |
| **Program: quality ->** ***Modularity*** | 0-9 | 10-14 | 15-19 | 20 |
| | | | | |
| **Program: quality ->** ***Simplicity*** | 0-9 | 10-14 | 15-19 | 20 |
| | | | | |
| **Total Grade (100)** | | | | |

# SE 471 Software Architecture

## Problems:

Given the following design (next page), implement it in Java. Note:

1. You may add more attributes or operations to a class if necessary. Specifically, you may use meaningful operations for FBI_Agent and CIA_Agent classes. Remember that your CIA_Agent and FBI_Agent class should implement runnable interface. Each agent object has its own thread for doing the assigned tasks.
2. Read textbook to see some example code snippets for the object pool pattern (pp. 170—174).
3. Some corrections:

```
private ObjectPool(ObjectCreation_IF c, int max){
  instanceCount=0;
  creator=c;
  maxInstances=max;
  pool = new Object[maxInstances];
}

public synchronized static ObjectPool getPoolInstance(ObjectCreation_IF c, int max){
   if (poolInstance==null)
      poolInstance = new ObjectPool(c, max);
  return poolInstance;
}
```

4. To demonstrate how a limited number of agents are requested to process tasks, your testing code (FBIAgentApp or CIAAgentAPP) should create a pool of 5 agents to service 10 task requesters. Each agent should leave a unique foot prints while it is serving a requester.

## Solution:

- First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named "Lab4CodeSubmission". One submission from each team.

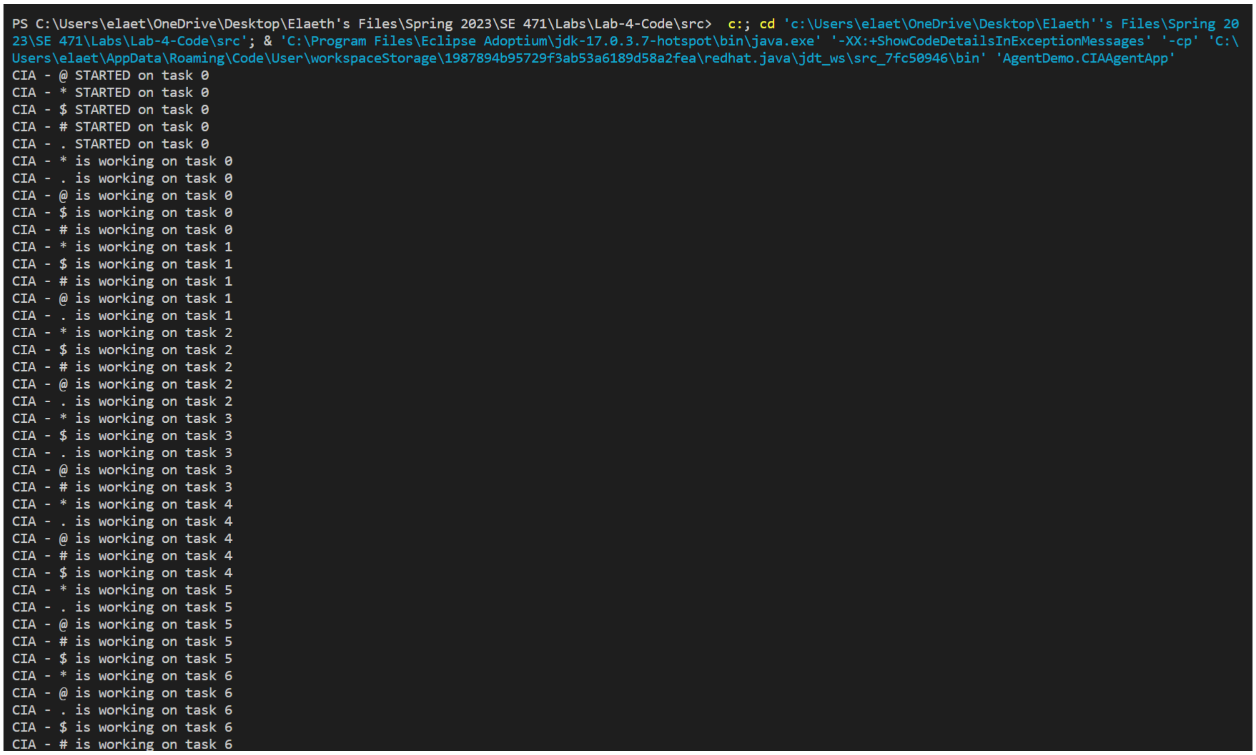
- Paste a screenshot of a run of your program here.

- Also paste all you source code here.

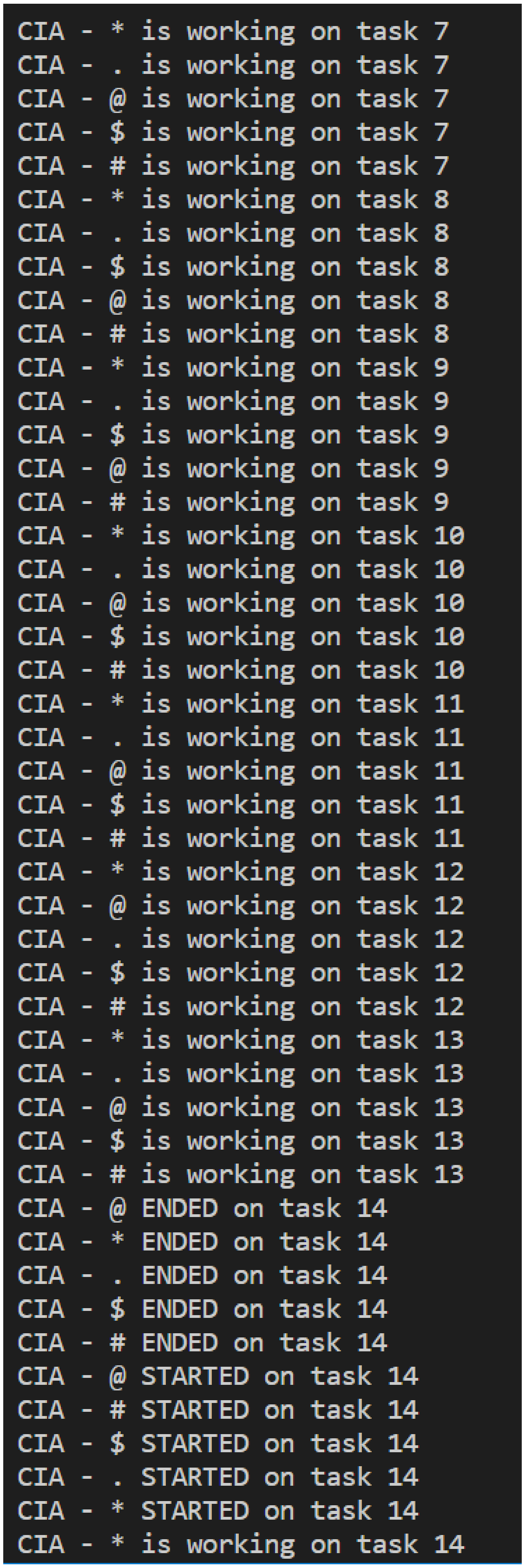
- Save this report in PDF, then submit the pdf report to the graded assignment named "Lab4ReportSubmission". One submission from each team

# ***Screenshots of Outputs for Lab 4:***

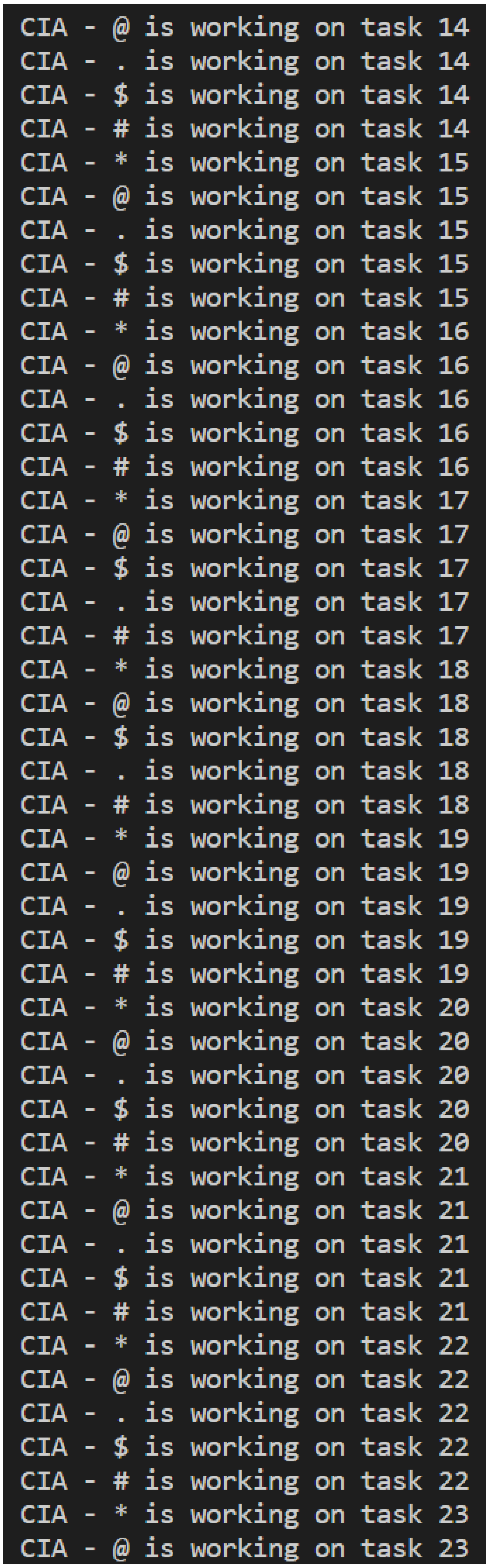
## CIA:



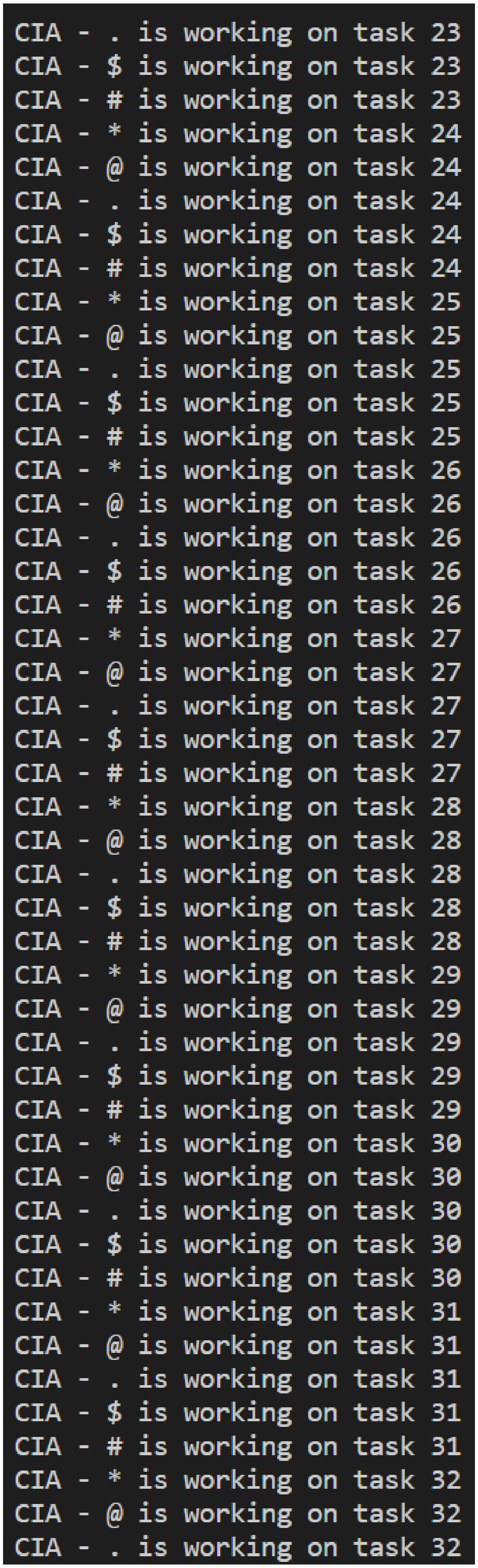
```
PS C:\Users\elaet\OneDrive\Desktop\Elaeth's Files\Spring 2023\SE 471\Labs\Lab-4-Code\src>  c:; cd 'c:\Users\elaet\OneDrive\Desktop\Elaeth''s Files\Spring 20
23\SE 471\Labs\Lab-4-Code\src'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\
Users\elaet\AppData\Roaming\Code\User\workspaceStorage\1987894b95729f3ab53a6189d58a2fea\redhat.java\jdt_ws\src_7fc50946\bin' 'AgentDemo.CIAAgentApp'
CIA - @ STARTED on task 0
CIA - * STARTED on task 0
CIA - $ STARTED on task 0
CIA - # STARTED on task 0
CIA - . STARTED on task 0
CIA - * is working on task 0
CIA - . is working on task 0
CIA - @ is working on task 0
CIA - $ is working on task 0
CIA - # is working on task 0
CIA - * is working on task 1
CIA - $ is working on task 1
CIA - # is working on task 1
CIA - @ is working on task 1
CIA - . is working on task 1
CIA - * is working on task 2
CIA - $ is working on task 2
CIA - # is working on task 2
CIA - @ is working on task 2
CIA - . is working on task 2
CIA - * is working on task 3
CIA - $ is working on task 3
CIA - . is working on task 3
CIA - @ is working on task 3
CIA - # is working on task 3
CIA - * is working on task 4
CIA - . is working on task 4
CIA - @ is working on task 4
CIA - # is working on task 4
CIA - $ is working on task 4
CIA - * is working on task 5
CIA - . is working on task 5
CIA - @ is working on task 5
CIA - # is working on task 5
CIA - $ is working on task 5
CIA - * is working on task 6
CIA - @ is working on task 6
CIA - . is working on task 6
CIA - $ is working on task 6
CIA - # is working on task 6
```



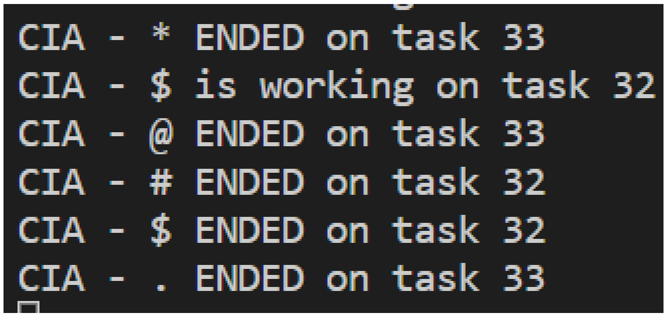
```
CIA - * is working on task 7
CIA - . is working on task 7
CIA - @ is working on task 7
CIA - $ is working on task 7
CIA - # is working on task 7
CIA - * is working on task 8
CIA - . is working on task 8
CIA - $ is working on task 8
CIA - @ is working on task 8
CIA - # is working on task 8
CIA - * is working on task 9
CIA - . is working on task 9
CIA - $ is working on task 9
CIA - @ is working on task 9
CIA - # is working on task 9
CIA - * is working on task 10
CIA - . is working on task 10
CIA - @ is working on task 10
CIA - $ is working on task 10
CIA - # is working on task 10
CIA - * is working on task 11
CIA - . is working on task 11
CIA - @ is working on task 11
CIA - $ is working on task 11
CIA - # is working on task 11
CIA - * is working on task 12
CIA - @ is working on task 12
CIA - . is working on task 12
CIA - $ is working on task 12
CIA - # is working on task 12
CIA - * is working on task 13
CIA - . is working on task 13
CIA - @ is working on task 13
CIA - $ is working on task 13
CIA - # is working on task 13
CIA - @ ENDED on task 14
CIA - * ENDED on task 14
CIA - . ENDED on task 14
CIA - $ ENDED on task 14
CIA - # ENDED on task 14
CIA - @ STARTED on task 14
CIA - # STARTED on task 14
CIA - $ STARTED on task 14
CIA - . STARTED on task 14
CIA - * STARTED on task 14
CIA - * is working on task 14
```



```
CIA - @ is working on task 14
CIA - . is working on task 14
CIA - $ is working on task 14
CIA - # is working on task 14
CIA - * is working on task 15
CIA - @ is working on task 15
CIA - . is working on task 15
CIA - $ is working on task 15
CIA - # is working on task 15
CIA - * is working on task 16
CIA - @ is working on task 16
CIA - . is working on task 16
CIA - $ is working on task 16
CIA - # is working on task 16
CIA - * is working on task 17
CIA - @ is working on task 17
CIA - $ is working on task 17
CIA - . is working on task 17
CIA - # is working on task 17
CIA - * is working on task 18
CIA - @ is working on task 18
CIA - $ is working on task 18
CIA - . is working on task 18
CIA - # is working on task 18
CIA - * is working on task 19
CIA - @ is working on task 19
CIA - . is working on task 19
CIA - $ is working on task 19
CIA - # is working on task 19
CIA - * is working on task 20
CIA - @ is working on task 20
CIA - . is working on task 20
CIA - $ is working on task 20
CIA - # is working on task 20
CIA - * is working on task 21
CIA - @ is working on task 21
CIA - . is working on task 21
CIA - $ is working on task 21
CIA - # is working on task 21
CIA - * is working on task 22
CIA - @ is working on task 22
CIA - . is working on task 22
CIA - $ is working on task 22
CIA - # is working on task 22
CIA - * is working on task 23
CIA - @ is working on task 23
```

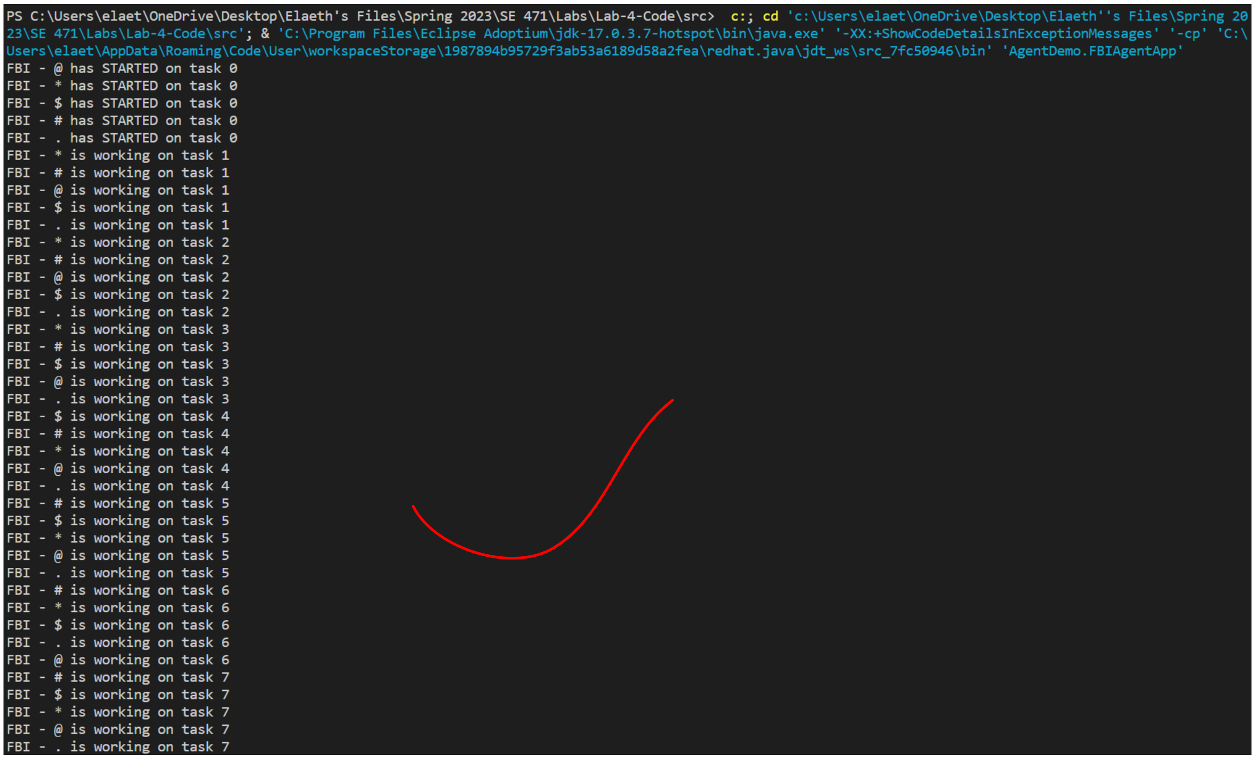


```
CIA - . is working on task 23
CIA - $ is working on task 23
CIA - # is working on task 23
CIA - * is working on task 24
CIA - @ is working on task 24
CIA - . is working on task 24
CIA - $ is working on task 24
CIA - # is working on task 24
CIA - * is working on task 25
CIA - @ is working on task 25
CIA - . is working on task 25
CIA - $ is working on task 25
CIA - # is working on task 25
CIA - * is working on task 26
CIA - @ is working on task 26
CIA - . is working on task 26
CIA - $ is working on task 26
CIA - # is working on task 26
CIA - * is working on task 27
CIA - @ is working on task 27
CIA - . is working on task 27
CIA - $ is working on task 27
CIA - # is working on task 27
CIA - * is working on task 28
CIA - @ is working on task 28
CIA - . is working on task 28
CIA - $ is working on task 28
CIA - # is working on task 28
CIA - * is working on task 29
CIA - @ is working on task 29
CIA - . is working on task 29
CIA - $ is working on task 29
CIA - # is working on task 29
CIA - * is working on task 30
CIA - @ is working on task 30
CIA - . is working on task 30
CIA - $ is working on task 30
CIA - # is working on task 30
CIA - * is working on task 31
CIA - @ is working on task 31
CIA - . is working on task 31
CIA - $ is working on task 31
CIA - # is working on task 31
CIA - * is working on task 32
CIA - @ is working on task 32
CIA - . is working on task 32
```

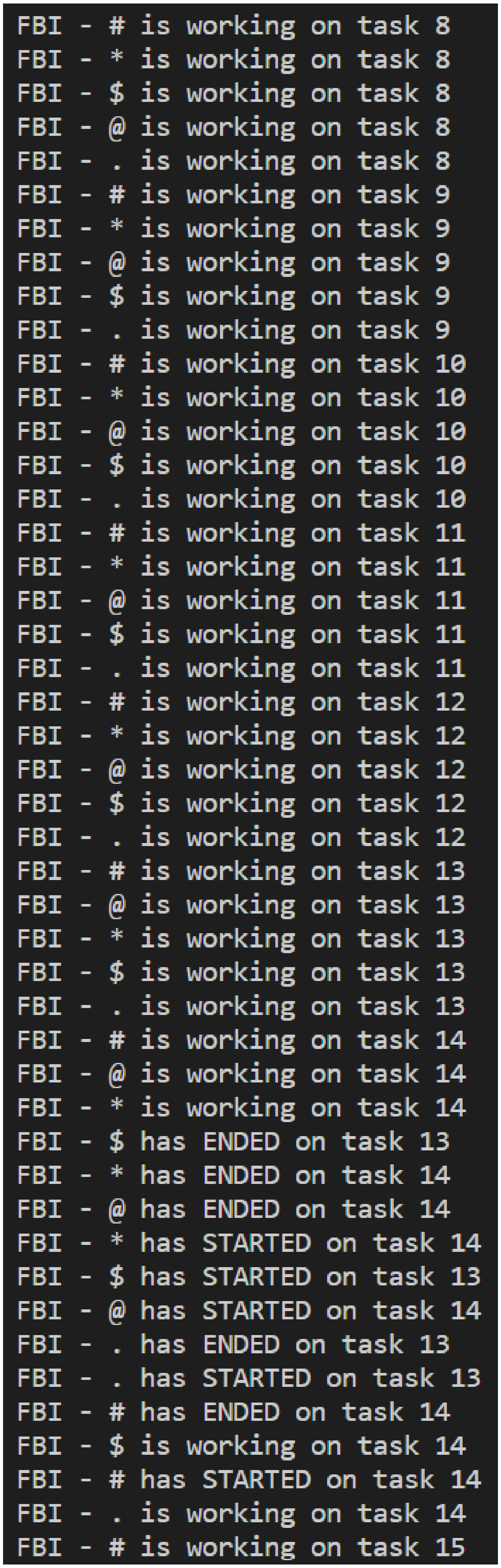


```
CIA - * ENDED on task 33
CIA - $ is working on task 32
CIA - @ ENDED on task 33
CIA - # ENDED on task 32
CIA - $ ENDED on task 32
CIA - . ENDED on task 33
```

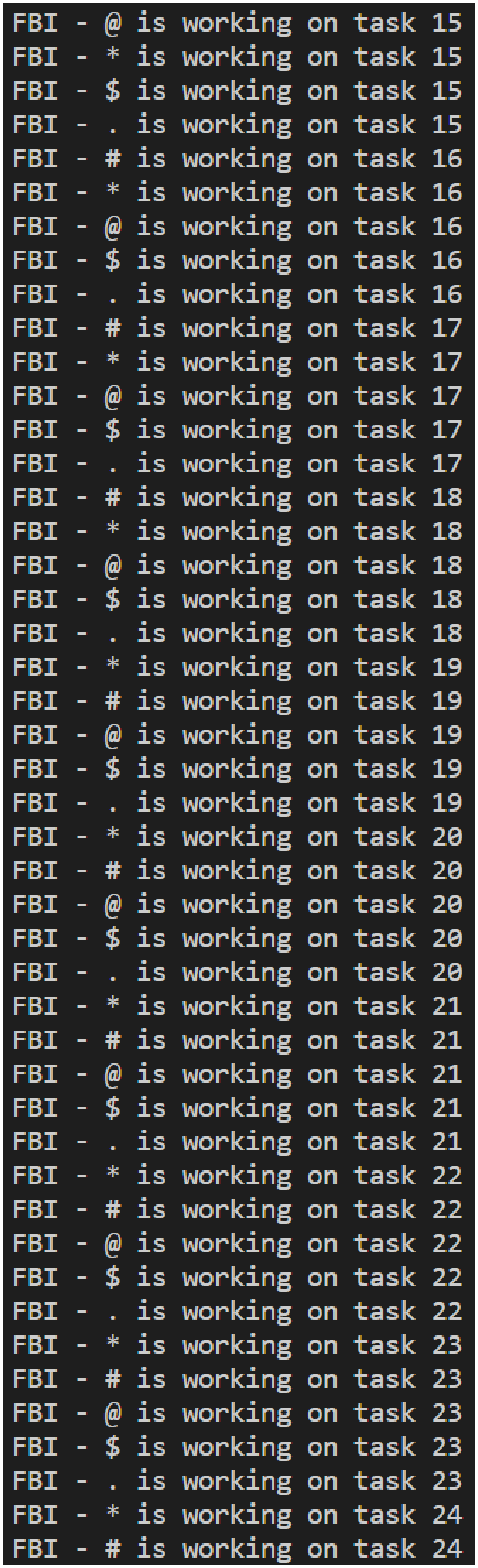
# FBI:



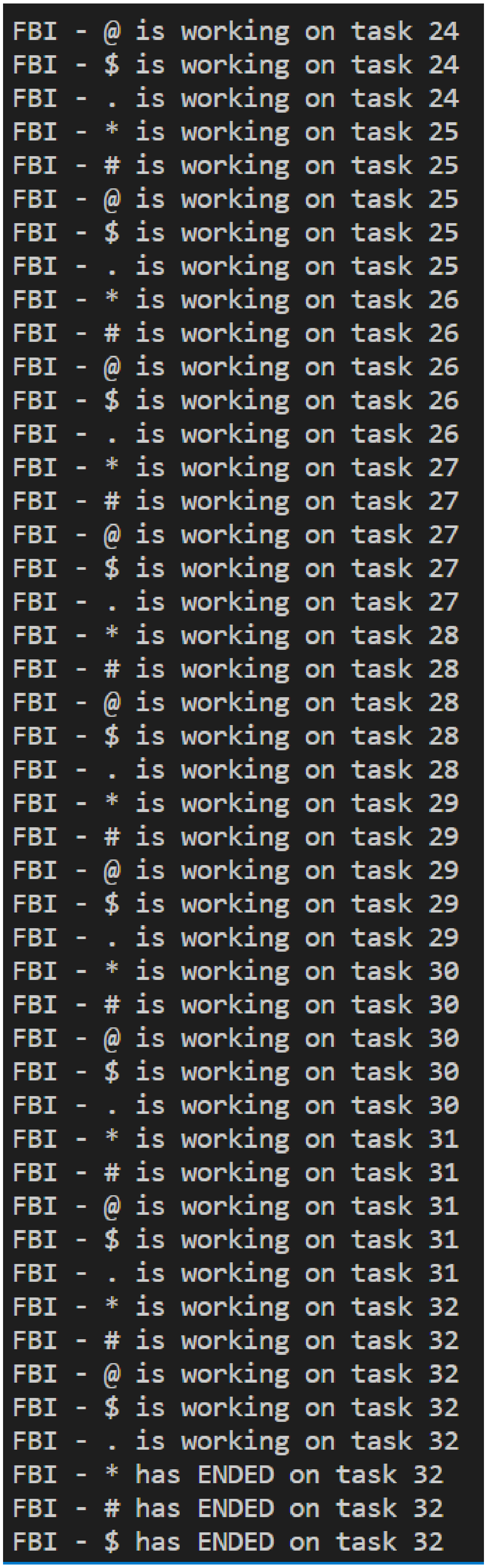
```
PS C:\Users\elaet\OneDrive\Desktop\Elaeth's Files\Spring 2023\SE 471\Labs\Lab-4-Code\src>  c:; cd 'c:\Users\elaet\OneDrive\Desktop\Elaeth''s Files\Spring 20
23\SE 471\Labs\Lab-4-Code\src'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\
Users\elaet\AppData\Roaming\Code\User\workspaceStorage\1987894b95729f3ab53a6189d58a2fea\redhat.java\jdt_ws\src_7fc50946\bin' 'AgentDemo.FBIAgentApp'
FBI - @ has STARTED on task 0
FBI - * has STARTED on task 0
FBI - $ has STARTED on task 0
FBI - # has STARTED on task 0
FBI - . has STARTED on task 0
FBI - * is working on task 1
FBI - # is working on task 1
FBI - @ is working on task 1
FBI - $ is working on task 1
FBI - . is working on task 1
FBI - * is working on task 2
FBI - # is working on task 2
FBI - @ is working on task 2
FBI - $ is working on task 2
FBI - . is working on task 2
FBI - * is working on task 3
FBI - # is working on task 3
FBI - $ is working on task 3
FBI - @ is working on task 3
FBI - . is working on task 3
FBI - $ is working on task 4
FBI - # is working on task 4
FBI - * is working on task 4
FBI - @ is working on task 4
FBI - . is working on task 4
FBI - # is working on task 5
FBI - $ is working on task 5
FBI - * is working on task 5
FBI - @ is working on task 5
FBI - . is working on task 5
FBI - # is working on task 6
FBI - * is working on task 6
FBI - $ is working on task 6
FBI - . is working on task 6
FBI - @ is working on task 6
FBI - # is working on task 7
FBI - $ is working on task 7
FBI - * is working on task 7
FBI - @ is working on task 7
FBI - . is working on task 7
```



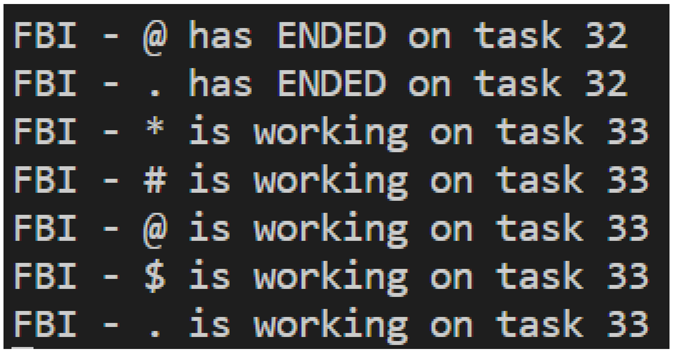
```
FBI - # is working on task 8
FBI - * is working on task 8
FBI - $ is working on task 8
FBI - @ is working on task 8
FBI - . is working on task 8
FBI - # is working on task 9
FBI - * is working on task 9
FBI - @ is working on task 9
FBI - $ is working on task 9
FBI - . is working on task 9
FBI - # is working on task 10
FBI - * is working on task 10
FBI - @ is working on task 10
FBI - $ is working on task 10
FBI - . is working on task 10
FBI - # is working on task 11
FBI - * is working on task 11
FBI - @ is working on task 11
FBI - $ is working on task 11
FBI - . is working on task 11
FBI - # is working on task 12
FBI - * is working on task 12
FBI - @ is working on task 12
FBI - $ is working on task 12
FBI - . is working on task 12
FBI - # is working on task 13
FBI - @ is working on task 13
FBI - * is working on task 13
FBI - $ is working on task 13
FBI - . is working on task 13
FBI - # is working on task 14
FBI - @ is working on task 14
FBI - * is working on task 14
FBI - $ has ENDED on task 13
FBI - * has ENDED on task 14
FBI - @ has ENDED on task 14
FBI - * has STARTED on task 14
FBI - $ has STARTED on task 13
FBI - @ has STARTED on task 14
FBI - . has ENDED on task 13
FBI - . has STARTED on task 13
FBI - # has ENDED on task 14
FBI - $ is working on task 14
FBI - # has STARTED on task 14
FBI - . is working on task 14
FBI - # is working on task 15
```



```
FBI - @ is working on task 15
FBI - * is working on task 15
FBI - $ is working on task 15
FBI - . is working on task 15
FBI - # is working on task 16
FBI - * is working on task 16
FBI - @ is working on task 16
FBI - $ is working on task 16
FBI - . is working on task 16
FBI - # is working on task 17
FBI - * is working on task 17
FBI - @ is working on task 17
FBI - $ is working on task 17
FBI - . is working on task 17
FBI - # is working on task 18
FBI - * is working on task 18
FBI - @ is working on task 18
FBI - $ is working on task 18
FBI - . is working on task 18
FBI - * is working on task 19
FBI - # is working on task 19
FBI - @ is working on task 19
FBI - $ is working on task 19
FBI - . is working on task 19
FBI - * is working on task 20
FBI - # is working on task 20
FBI - @ is working on task 20
FBI - $ is working on task 20
FBI - . is working on task 20
FBI - * is working on task 21
FBI - # is working on task 21
FBI - @ is working on task 21
FBI - $ is working on task 21
FBI - . is working on task 21
FBI - * is working on task 22
FBI - # is working on task 22
FBI - @ is working on task 22
FBI - $ is working on task 22
FBI - . is working on task 22
FBI - * is working on task 23
FBI - # is working on task 23
FBI - @ is working on task 23
FBI - $ is working on task 23
FBI - . is working on task 23
FBI - * is working on task 24
FBI - # is working on task 24
```



```
FBI - @ is working on task 24
FBI - $ is working on task 24
FBI - . is working on task 24
FBI - * is working on task 25
FBI - # is working on task 25
FBI - @ is working on task 25
FBI - $ is working on task 25
FBI - . is working on task 25
FBI - * is working on task 26
FBI - # is working on task 26
FBI - @ is working on task 26
FBI - $ is working on task 26
FBI - . is working on task 26
FBI - * is working on task 27
FBI - # is working on task 27
FBI - @ is working on task 27
FBI - $ is working on task 27
FBI - . is working on task 27
FBI - * is working on task 28
FBI - # is working on task 28
FBI - @ is working on task 28
FBI - $ is working on task 28
FBI - . is working on task 28
FBI - * is working on task 29
FBI - # is working on task 29
FBI - @ is working on task 29
FBI - $ is working on task 29
FBI - . is working on task 29
FBI - * is working on task 30
FBI - # is working on task 30
FBI - @ is working on task 30
FBI - $ is working on task 30
FBI - . is working on task 30
FBI - * is working on task 31
FBI - # is working on task 31
FBI - @ is working on task 31
FBI - $ is working on task 31
FBI - . is working on task 31
FBI - * is working on task 32
FBI - # is working on task 32
FBI - @ is working on task 32
FBI - $ is working on task 32
FBI - . is working on task 32
FBI - * has ENDED on task 32
FBI - # has ENDED on task 32
FBI - $ has ENDED on task 32
```



```
FBI - @ has ENDED on task 32
FBI - . has ENDED on task 32
FBI - * is working on task 33
FBI - # is working on task 33
FBI - @ is working on task 33
FBI - $ is working on task 33
FBI - . is working on task 33
```

# AgentDemo (package)

## Agent_IF.java

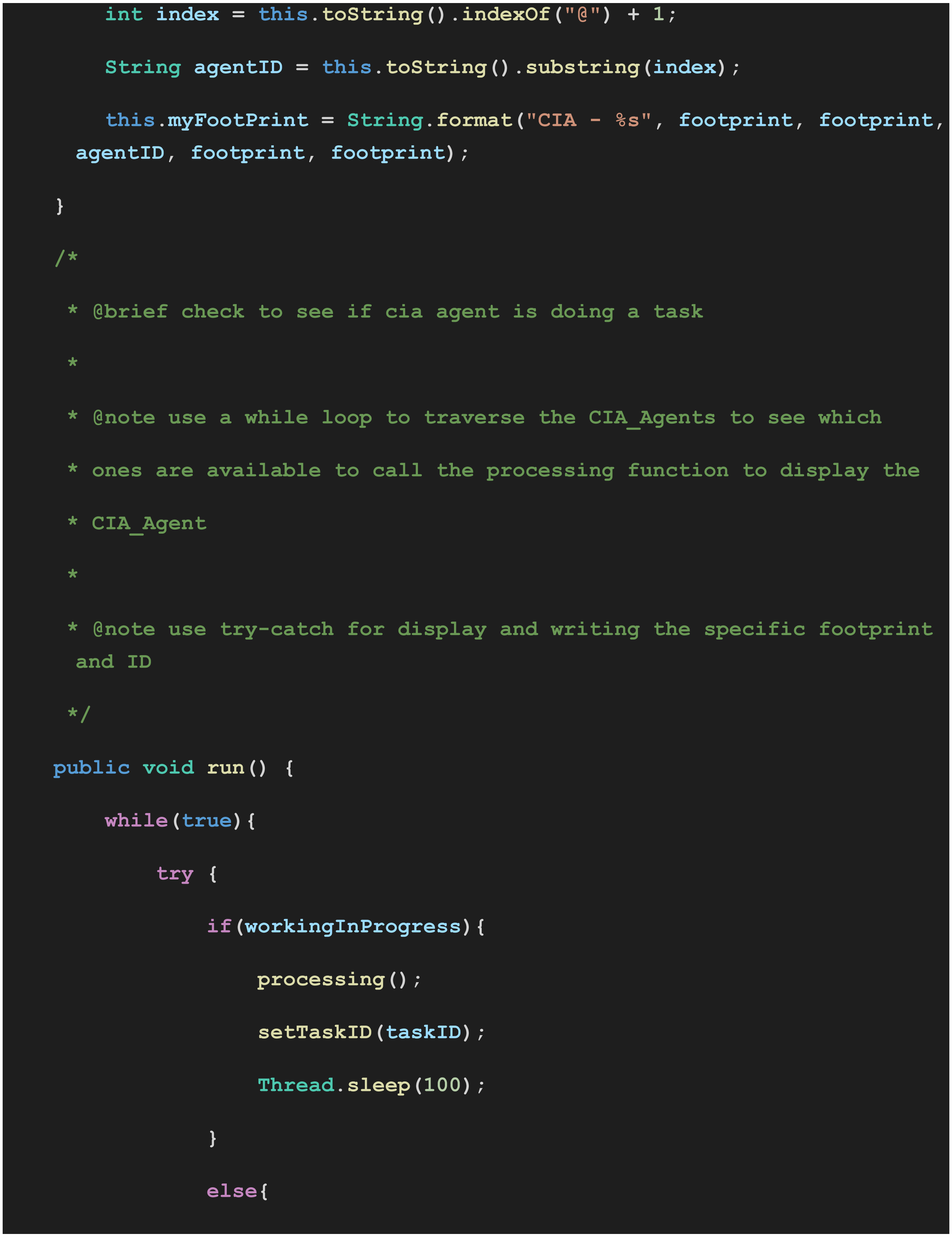
```
package AgentDemo;


/*
 * @brief: Following the contents of
 * Agent_IF from UML diagram
 */
public interface Agent_IF{
    public void startTask();
    public void stopTask();
    public void setTaskID(int id);
}
```

## CIA_Agent_Creator.java

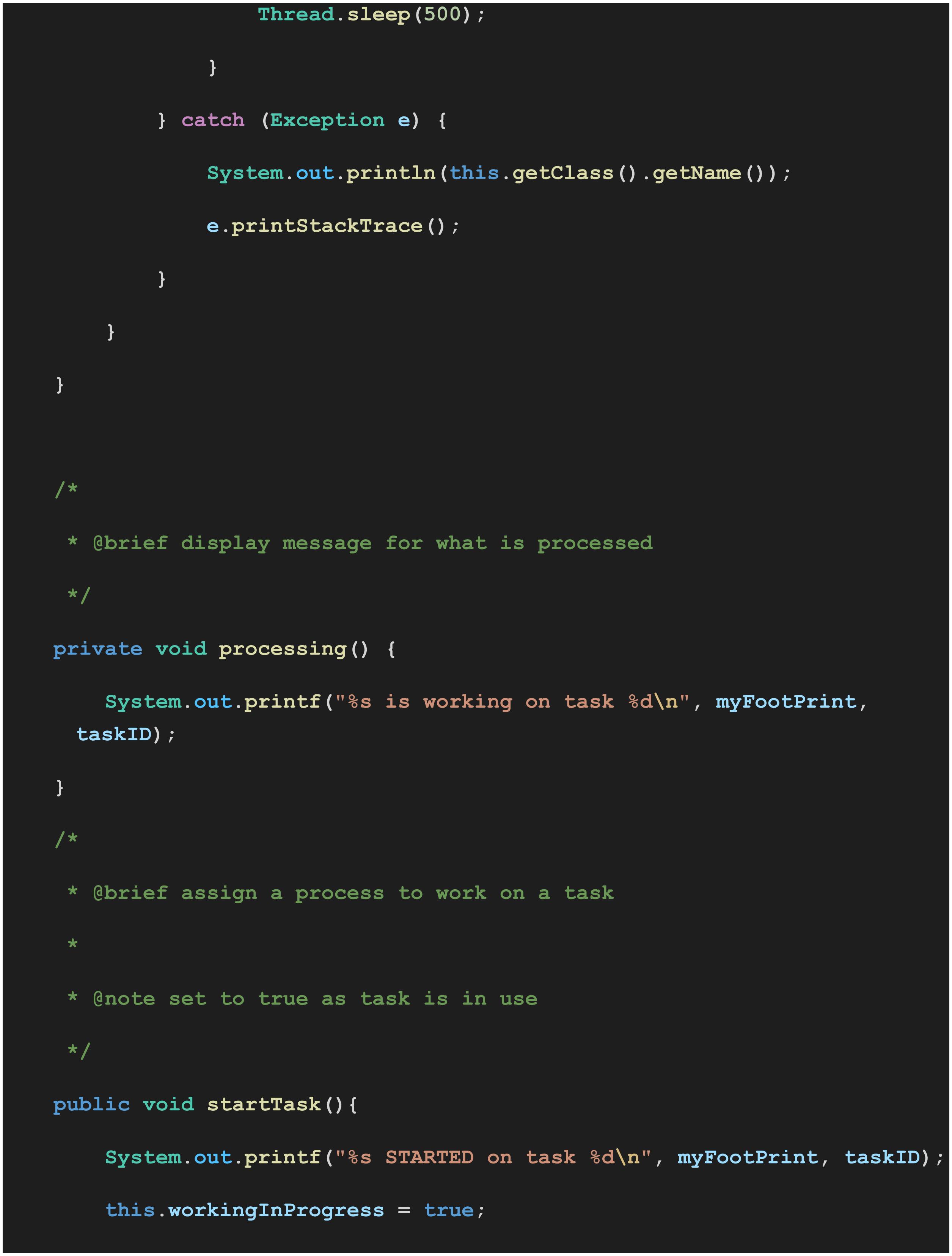
```
package AgentDemo;
import PoolPattern.ObjectCreation_IF;
/*
 * @brief CIA_Agent_Creator class that connects from the pool pattern
      folder
 * to obtain object creation interface
 *
 * @note footPrints string array to identify each CIA_Agent
 * @note index will be initalized

 */

public class CIA_Agent_Creator implements ObjectCreation_IF {

    private String[] footPrints = {"@", "#", "$", "*", ".", "?"};

    private int index = 0;

    /*

     * @brief creates and returns a CIA agent with

     * a specific footprint

     * @return the CIA Agent

     */

    public Object create() {

        CIA_Agent agent = new CIA_Agent(this.footPrints[(index++)]);

        new Thread(agent).start();

        return agent;

    }

}
```

CIA_Agent.java

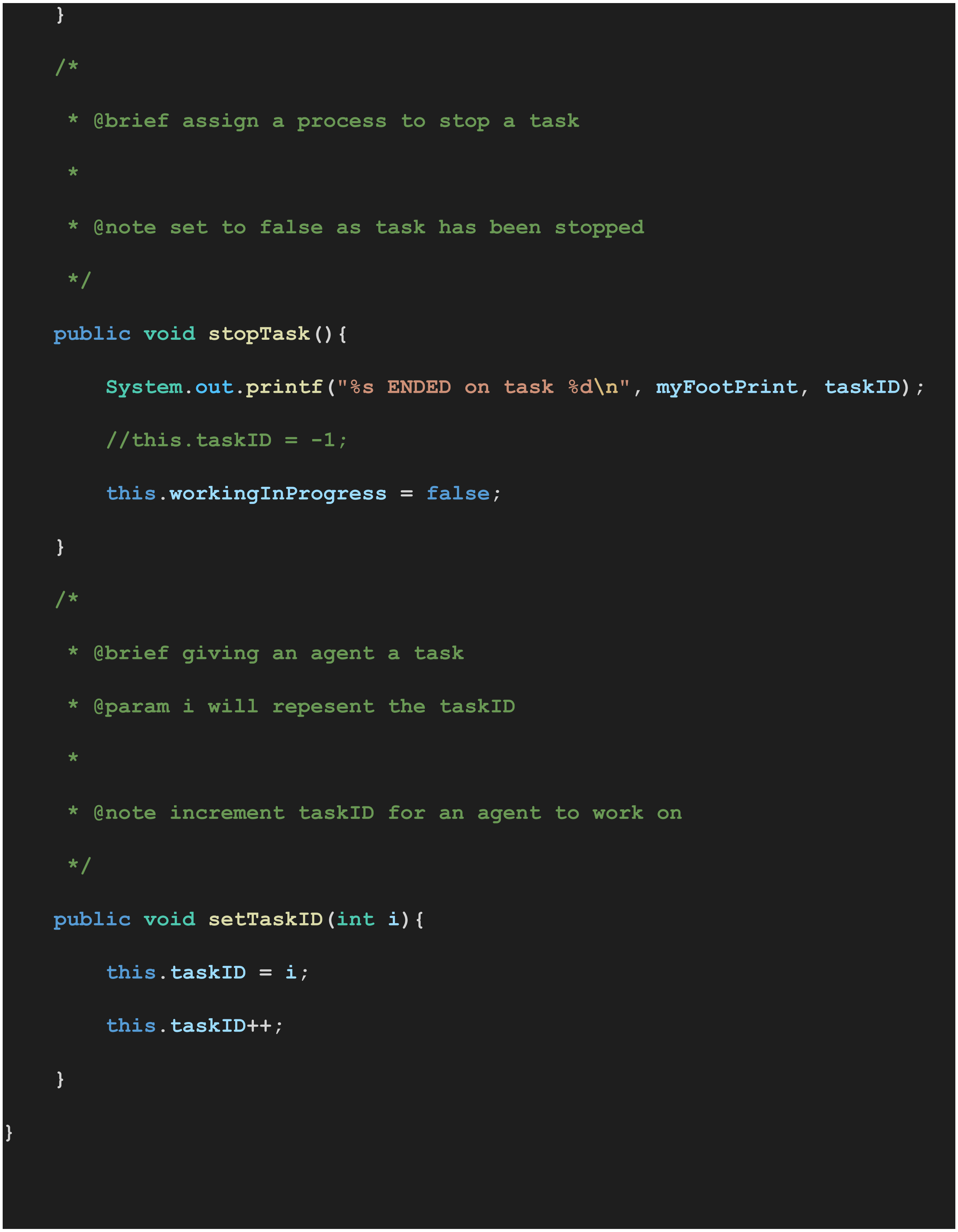
```
package AgentDemo;


/*

 * @brief CIA_Agent class that implements Agent_IF and Runnable
 *
 * @note Agent_IF is an interface that implements from CIA_Agent
 * @note Runnable is an interface from runnable that implements from CIA_Agent
 *
 * @note workingInProgress (boolean) is what the agent is working on currently
 * @note myFootPrint (String) is the special character that is based from the foot print
 *
 * @note taskID (added) to see which agent is doing which task
 */
public class CIA_Agent implements Runnable, Agent_IF{
    private boolean workingInProgress;
    private int taskID;
    private String myFootPrint;

    /*
    * @brief Constructor
    * @param footprint is the special character to indicate the agent
    *
    * @note index is to gain access of getting the first element within the footPrints array
    */
    public CIA_Agent(String footprint) {
```



```
        int index = this.toString().indexOf("@") + 1;
        String agentID = this.toString().substring(index);
        this.myFootPrint = String.format("CIA - %s", footprint, footprint,
      agentID, footprint, footprint);
    }
    /*
     * @brief check to see if cia agent is doing a task
     *
     * @note use a while loop to traverse the CIA_Agents to see which
     * ones are available to call the processing function to display the
     * CIA_Agent
     *
     * @note use try-catch for display and writing the specific footprint
      and ID
     */
    public void run() {
        while(true){
            try {
                if(workingInProgress){
                    processing();
                    setTaskID(taskID);
                    Thread.sleep(100);
                }
                else{
```



```
                Thread.sleep(500);
            }
        } catch (Exception e) {
            System.out.println(this.getClass().getName());
            e.printStackTrace();
        }
    }
}

/*
 * @brief display message for what is processed
 */
private void processing() {
    System.out.printf("%s is working on task %d\n", myFootPrint, taskID);
}
/*
 * @brief assign a process to work on a task
 *
 * @note set to true as task is in use
 */
public void startTask(){
    System.out.printf("%s STARTED on task %d\n", myFootPrint, taskID);
    this.workingInProgress = true;
```



```
    }
    /*
     * @brief assign a process to stop a task
     *
     * @note set to false as task has been stopped
     */
    public void stopTask(){
        System.out.printf("%s ENDED on task %d\n", myFootPrint, taskID);
        //this.taskID = -1;
        this.workingInProgress = false;
    }
    /*
     * @brief giving an agent a task
     * @param i will repesent the taskID
     *
     * @note increment taskID for an agent to work on
     */
    public void setTaskID(int i){
        this.taskID = i;
        this.taskID++;
    }
}
```

**CIAAgentApp.java**

```java
package AgentDemo;


import PoolPattern.ObjectPool;
/*
 * @brief CIAAgentApp class that keeps track of the CIA_Agent
 *
 */
public class CIAAgentApp{
    public static void main(String[] args) {
        ObjectPool server = ObjectPool.getPoolInstance(new
      CIA_Agent_Creator(), 5);
        for(int i = 0; i < 10; i++){
            Thread client = new Thread(new TaskRequester(server));
            client.start();
        }
    }
}
```

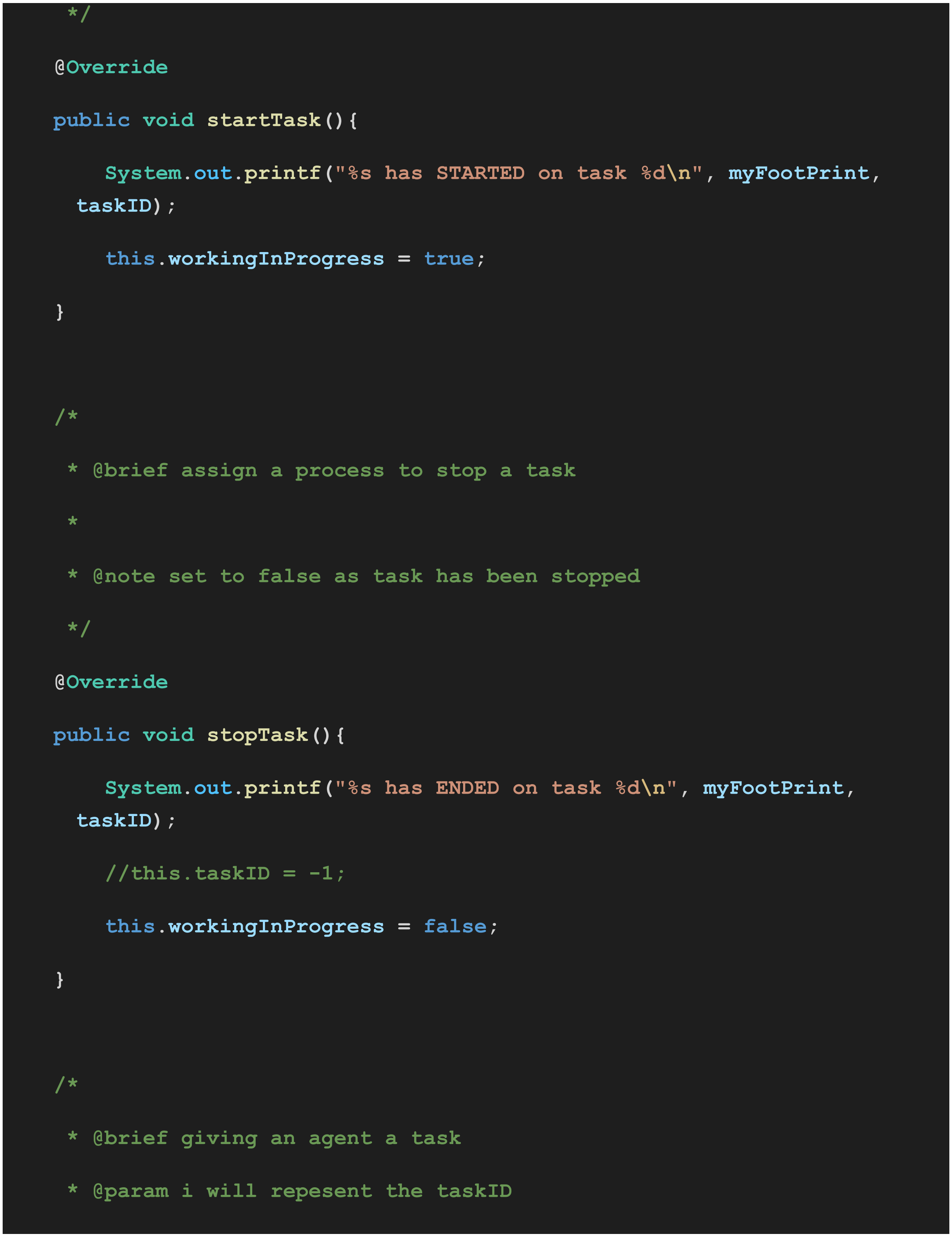
## FBI_Agent_Creator.java

```java
package AgentDemo;
import PoolPattern.ObjectCreation_IF;
/*
 * @brief FBI_Agent_Creator class that connects from the pool pattern folder
 * to obtain object creation interface
 *
 * @note footPrints string array to identify each FBI_Agent
 * @note index will be initalized
 */
public class FBI_Agent_Creator implements ObjectCreation_IF{
    private String[] footPrints = {"@","#","$","*",".","?"};
    private int index = 0;

    /*
     * @brief creates and returns a FBI agent with
     * a specific footprint
     * @return the FBI Agent
     *
     * @note copied the same format with the CIA_Agent_Creator java file
     */
    @Override
    public Object create(){
        FBI_Agent agent = new FBI_Agent(footPrints[(index++)]);

        new Thread(agent).start();

        return agent;

    }

}
```

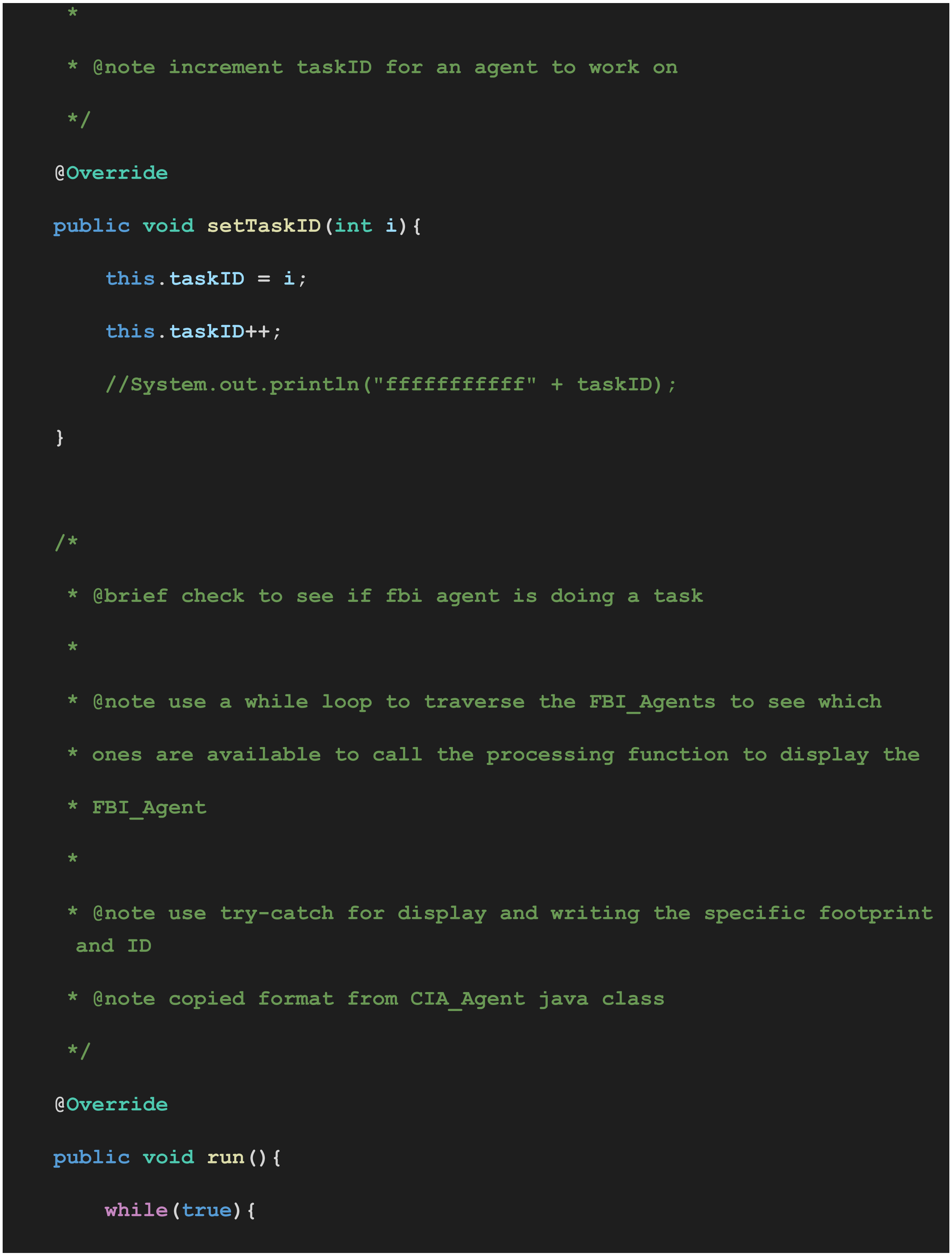
**FBI_Agent.java**

```
package AgentDemo;


/*

 * @brief FBI_Agent class that implements Agent_IF and Runnable

 *

 * @note Agent_IF is an interface that implements from FBI_Agent

 * @note Runnable is an interface from runnable that implements from FBI_Agent

 *

 * @note workingInProgress (boolean) is what the agent is working on currently

 * @note myFootPrint (String) is the special character that is based from the foot print

 *

 * @note taskID (added) to see which agent is doing which task

 */
```

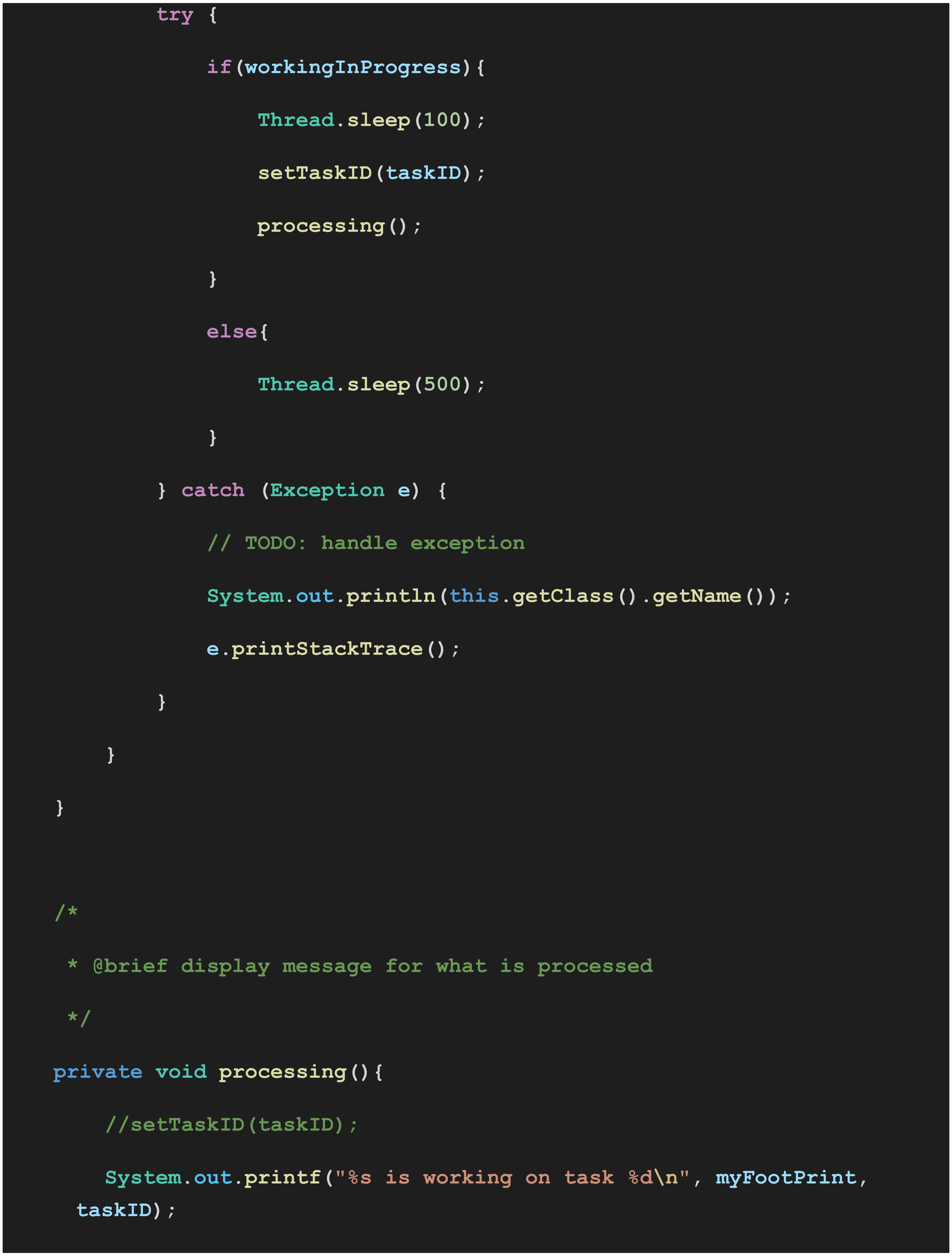
```
public class FBI_Agent implements Runnable, Agent_IF{

   private boolean workingInProgress;

   private String myFootPrint;


   private int taskID;


   /*

    * @brief Constructor

    * @param footprint is the special character to indicate the agent

    *

    * @note index is to gain access of getting the first element within
      the footPrints array

    */

    public FBI_Agent(String footprint){

       int index = this.toString().indexOf("@") + 1;

       String agentID = this.toString().substring(index);

       //workingInProgress = true;

       this.myFootPrint = String.format("FBI - %s", footprint, agentID);

    }


    /*

     * @brief assign a process to work on a task

     *

     * @note set to true as task is in use
```



```
  */
 @Override
 public void startTask(){
     System.out.printf("%s has STARTED on task %d\n", myFootPrint,
   taskID);
     this.workingInProgress = true;
 }


 /*
  * @brief assign a process to stop a task
  *
  * @note set to false as task has been stopped
  */
 @Override
 public void stopTask(){
     System.out.printf("%s has ENDED on task %d\n", myFootPrint,
   taskID);
     //this.taskID = -1;
     this.workingInProgress = false;
 }


 /*
  * @brief giving an agent a task
  * @param i will repesent the taskID
```



```
     *
     * @note increment taskID for an agent to work on
     */
    @Override
    public void setTaskID(int i){
        this.taskID = i;
        this.taskID++;
        //System.out.println("ffffffffff" + taskID);
    }

    /*
     * @brief check to see if fbi agent is doing a task
     *
     * @note use a while loop to traverse the FBI_Agents to see which
     * ones are available to call the processing function to display the
     * FBI_Agent
     *
     * @note use try-catch for display and writing the specific footprint
      and ID
     * @note copied format from CIA_Agent java class
     */
    @Override
    public void run(){
        while(true){
```



```
        try {
            if(workingInProgress){
                Thread.sleep(100);
                setTaskID(taskID);
                processing();
            }
            else{
                Thread.sleep(500);
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println(this.getClass().getName());
            e.printStackTrace();
        }
    }
}

/*
 * @brief display message for what is processed
 */
private void processing(){
    //setTaskID(taskID);
    System.out.printf("%s is working on task %d\n", myFootPrint,
  taskID);
```

```
    }

}
```

**FBIAgentApp.java**

```
package AgentDemo;


import PoolPattern.ObjectPool; //with uses relationship on uml


/*

 * @brief FBIAgentApp class that keeps track of the FBI_Agent

 *

 * @note copied same format for CIAAgentApp.java

 */

public class FBIAgentApp{

    public static void main(String[] args){

        ObjectPool server = ObjectPool.getPoolInstance(new FBI_Agent_Creator(), 5);

        for(int i = 0; i < 10; i++){

            Thread client = new Thread(new TaskRequester(server));

            client.start();

        }

    }

}
```

TaskRequester.java

```java
package AgentDemo;


import PoolPattern.ObjectPool;

/*

 * @brief TaskRequester will be a class that connects from

 * the ObjectPool java file and runnable

 *

 * @note since server is an ObjectPool object, it has to be

 * called from the import from the pool pattern folder

 */

public class TaskRequester implements Runnable{

    private ObjectPool server;


    /*

     * @brief Constructor

     * @param p to be used for task requesting services

     */

    public TaskRequester(ObjectPool p){

        this.server = p;

    }
```

```
/*
 * @brief given an available agent from the object pool,
 * it will search through any tasks given a period of time.
 *
 * @note this function will be using threads from runnable
 * @note try-catch will get the agent's record for tasks given.
 * To add on, as mentioned on the uml, we followed the format of the
  first and last two lines
 */
@Override
public void run(){
    //Agent_IF agent;
    try {
        Agent_IF agent = (Agent_IF)server.waitForObject();
        //agent.setTaskID();
        agent.startTask();
        Thread.sleep(2000);
        agent.stopTask();
        server.release(agent);
    } catch (InterruptedException e) {
        // TODO: handle exception
        e.printStackTrace();
    }
    }
}
```

## – break –

## PoolPattern (package)

### ObjectCreation_IF.java

```
package PoolPattern;

/*
 * @brief: Following the contents of
 * ObjectCreation_IF from UML diagram
 */
public interface ObjectCreation_IF {
    public Object create();
}
```

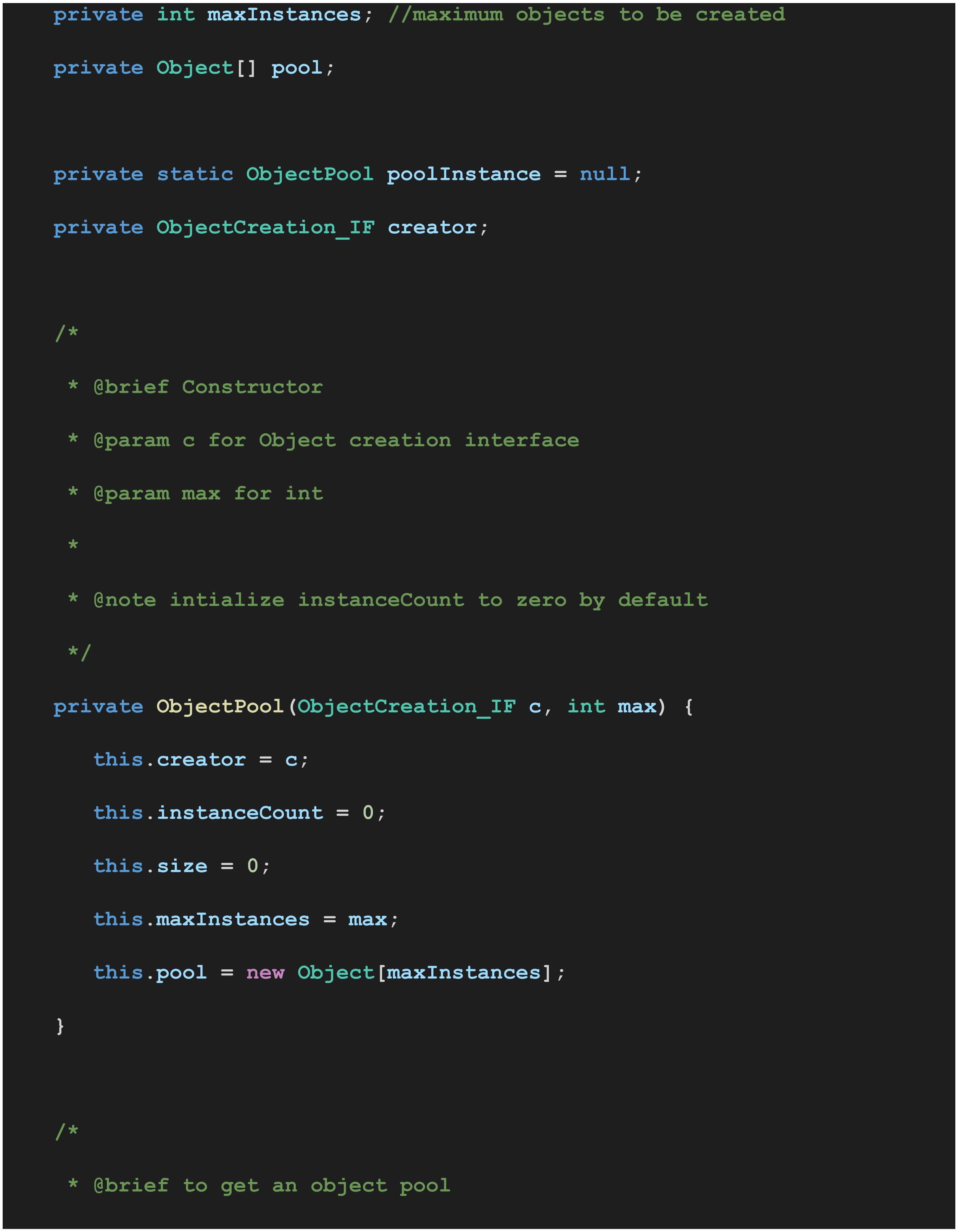
### ObjectPool_IF.java

```
package PoolPattern;

/*
 * @brief: Following the contents of
 * ObjectPool_IF from UML diagram
 */
```

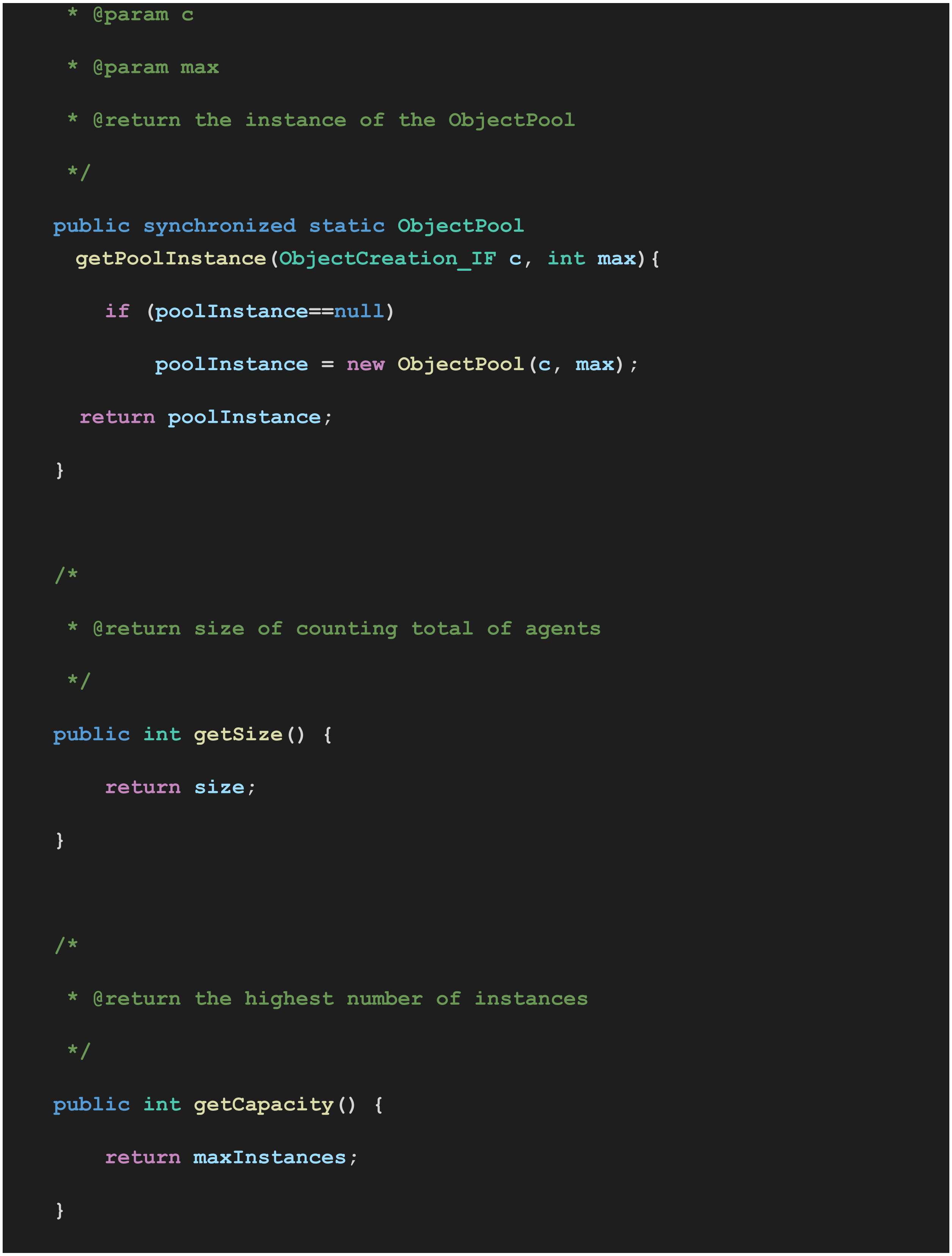
```
public interface ObjectPool_IF {
    int getSize();
    int getCapacity();
    void setCapacity(int c);
    Object getObject();
    Object waitForObject() throws InterruptedException;
    void release(Object o);
}
```

**ObjectPool.java**

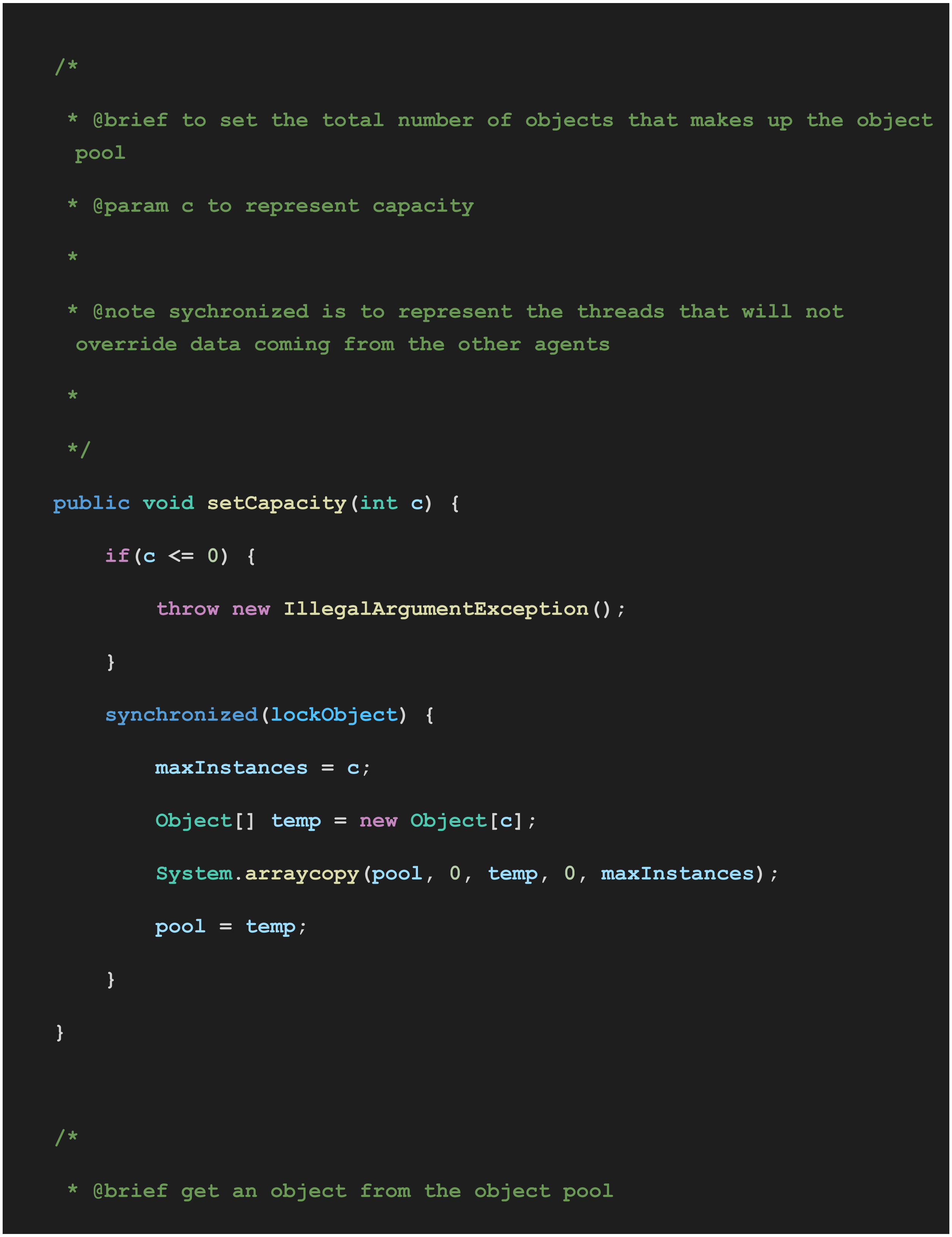
```
package PoolPattern;


/*
 * @brief use the object pool class to implement with the interface
 * as shown in the uml diagram
 *
 * @note size to keep track of free objects
 */
public class ObjectPool implements ObjectPool_IF{
    private final Object lockObject = new Object();

    private int size; //how many free objects
    private int instanceCount; //how many objects have been created
```



```
    private int maxInstances; //maximum objects to be created
    private Object[] pool;

    private static ObjectPool poolInstance = null;
    private ObjectCreation_IF creator;

    /*
     * @brief Constructor
     * @param c for Object creation interface
     * @param max for int
     *
     * @note intialize instanceCount to zero by default
     */
    private ObjectPool(ObjectCreation_IF c, int max) {
        this.creator = c;
        this.instanceCount = 0;
        this.size = 0;
        this.maxInstances = max;
        this.pool = new Object[maxInstances];
    }

    /*
     * @brief to get an object pool
```

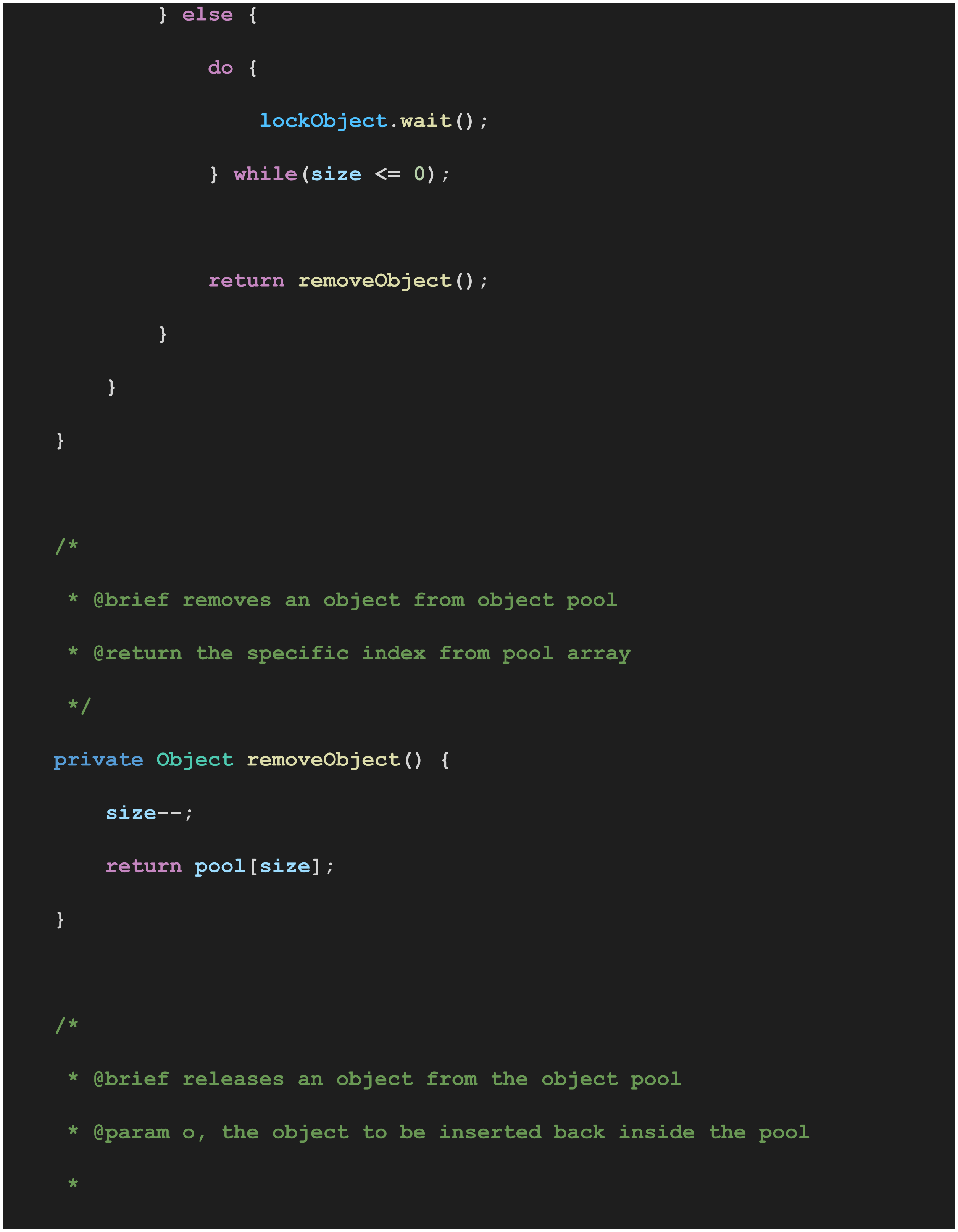


```
     * @param c

     * @param max

     * @return the instance of the ObjectPool

     */

    public synchronized static ObjectPool
      getPoolInstance(ObjectCreation_IF c, int max){

        if (poolInstance==null)

            poolInstance = new ObjectPool(c, max);

      return poolInstance;

    }


    /*

     * @return size of counting total of agents

     */

    public int getSize() {

        return size;

    }


    /*

     * @return the highest number of instances

     */

    public int getCapacity() {

        return maxInstances;

    }
```

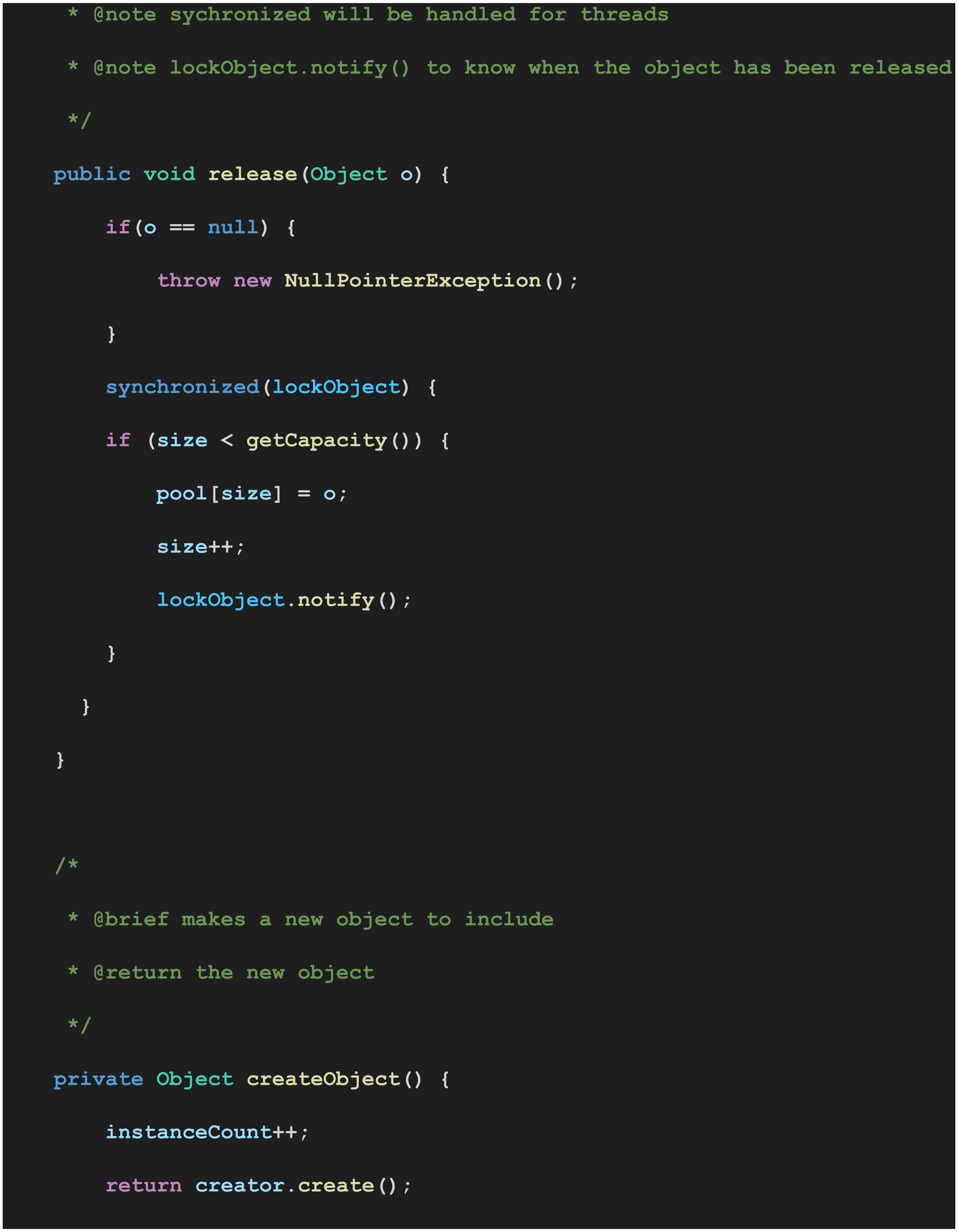


```
/*
 * @brief to set the total number of objects that makes up the object pool
 * @param c to represent capacity
 *
 * @note sychronized is to represent the threads that will not override data coming from the other agents
 *
 */
public void setCapacity(int c) {
    if(c <= 0) {
        throw new IllegalArgumentException();
    }
    synchronized(lockObject) {
        maxInstances = c;
        Object[] temp = new Object[c];
        System.arraycopy(pool, 0, temp, 0, maxInstances);
        pool = temp;
    }
}

/*
 * @brief get an object from the object pool
```

```
     * @return the object by calling the functions or NULL
     */
    public Object getObject() {
        synchronized(lockObject) {
            if(size > 0) {
                return removeObject();
            } else if (instanceCount < maxInstances) {
                return createObject();
            }
            return null;
        }
    }

    /*
     * @brief obtains an object when it becomes available from the object pool
     * @return the object
     */
    public Object waitForObject() throws InterruptedException {
        synchronized(lockObject) {
            if(size > 0) {
                return removeObject();
            } else if(instanceCount < maxInstances) {
                return createObject();
```



```
        } else {
            do {
                lockObject.wait();
            } while(size <= 0);

            return removeObject();
        }
    }
}

/*
 * @brief removes an object from object pool
 * @return the specific index from pool array
 */
private Object removeObject() {
    size--;
    return pool[size];
}

/*
 * @brief releases an object from the object pool
 * @param o, the object to be inserted back inside the pool
 *
```



```
     * @note sychronized will be handled for threads
     * @note lockObject.notify() to know when the object has been released
     */
    public void release(Object o) {
        if(o == null) {
            throw new NullPointerException();
        }
        synchronized(lockObject) {
        if (size < getCapacity()) {
            pool[size] = o;
            size++;
            lockObject.notify();
        }
      }
    }

    /*
     * @brief makes a new object to include
     * @return the new object
     */
    private Object createObject() {
        instanceCount++;
        return creator.create();
```

```
        }




}
```