Draghici Andreea-Maria
CR 3.1B

Raport Laborator 11

1. Am creat un nou proiect Java si am implementat codul din platform de laborator. Rezultatul obtinut este atasat mai jos.

```java
package com.home.lab10;


public class Accepter implements Runnable {
    private int c = 0;
    private Entry<String, String> entry;

    public Accepter(Entry<String, String> e) {
        entry = e;
    }

    public void run() {
        try {
            do {
                String request = entry.accept();
                System.out.println("Accepter request" + ": " + request);
                String result = request + (++c);
                System.out.println("Accepter result" + ": " + result);
                entry.reply(result);
            } while (true);
        } catch (InterruptedException e) {
        }
    }

}
```

AddTask.java × MultTask.java × lab9\Main.java × lab10\Main.java × Accepter.java × Ent

```java
package com.home.lab10;

public class Caller implements Runnable {

    private Entry<String, String> entry;
    private String id;

    public Caller(Entry<String, String> e, String s) {
        entry = e;
        id = s;
    }

    public void run() {
        try {
            for (int i = 0; i < 10; i++) {
                System.out.println("Caller " + id + " parameter: " + id);
                String result = entry.call(id);
                System.out.println("Caller " + id + " result: " + result);
            }
        } catch (InterruptedException e) {
        }
    }

}
```

Draghici Andreea-Maria
CR 3.1B

AddTask.java × MultTask.java × lab9\Main.java × lab10\Main.java × Accepter.java

```java
package com.home.lab10;

import java.util.LinkedList;
import java.util.Queue;

public class Port<T> {
    Queue<T> queue = new LinkedList<T>();
    int ready = 0;

    public synchronized void send(T v) {
        queue.add(v);
        ++ready;
        notifyAll();
    }

    public synchronized T receive() throws InterruptedException {
        while (ready == 0) wait();
        --ready;
        return queue.remove();
    }

}
```

pratoare 〉 src 〉 com 〉 home 〉 lab10 〉 Entry

AddTask.java × MultTask.java × lab9\Main.java × lab10\Main.java × Accepte

```java
package com.home.lab10;

public class Entry<R, P> {
    private CallMsg<R, P> cm;
    private Port<CallMsg<R, P>> cp = new Port<CallMsg<R, P>>();

    public P call(R req) throws InterruptedException {
        Channel<P> clientChan = new Channel<P>();
        cp.send(new CallMsg<R, P>(req, clientChan));
        return clientChan.receive();
    }

    public R accept() throws InterruptedException {
        cm = cp.receive();
        return cm.request;
    }

    public void reply(P res) throws InterruptedException {
        cm.replychan.send(res);
    }

    private class CallMsg<R1, P1> {
        R1 request;
        Channel<P1> replychan;

        CallMsg(R1 m, Channel<P1> c) {
            request = m;
            replychan = c;
        }
    }

}
```

Draghici Andreea-Maria
CR 3.1B

AddTask.java × MultTask.java × lab9\Main.java × lab10\Main.java × Acc

```java
package com.home.lab10;

public class Channel<T> {
    private T chan = null;
    int ready = 0;

    public synchronized void send(T mes)
            throws InterruptedException {
        // Copiaza mesajul pe canal
        chan = mes;

        // Instiinteaza receptorii potentiali facand
        // ready = 1
        ++ready;
        notifyAll();

        // Asteapta ca mesajul sa fie preluat de receptor,
        // canalul devenind vid
        while (chan != null) wait();
    }

    public synchronized T receive()
            throws InterruptedException {
        // Asteapta ca un mesaj sa fie disponibil pe canal,
        // cand ready = 1
        while (ready == 0) wait();

        // Copiaza mesajul local, elibereaza canalul si
        // instiinteaza potentialii transmitatori
        --ready;
        T tmp = chan;
        chan = null;
        notifyAll();
        return (tmp);
    }

}
```

AddTask.java × MultTask.java × lab9\Main.java × lab10\Main.java × Accepte

```java
package com.home.lab10;

public class Main {
    static Entry<String,String> e = new Entry<String,String>();
    static Caller ionel = new Caller(e, s: "Ionel");
    static Caller gigel = new Caller(e, s: "Gigel");
    static Accepter maria = new Accepter(e);

    public static void main(String[] args) {
        Thread tIonel = new Thread(ionel);
        Thread tGigel = new Thread(gigel);
        Thread tMaria = new Thread(maria);

        tIonel.start();
        tGigel.start();
        tMaria.start();
    }

}
```

Draghici Andreea-Maria
CR 3.1B

Output task 1:

```
Run: Main (4)
C:\Users\user\.jdks\corretto-16.0.2\bin\java.exe "
Caller Ionel parameter: Ionel
Caller Gigel parameter: Gigel
Accepter request: Ionel
Accepter result: Ionel1
Accepter request: Gigel
Accepter result: Gigel2
Caller Ionel result: Ionel1
Caller Ionel parameter: Ionel
Caller Gigel result: Gigel2
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel3
Caller Ionel result: Ionel3
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel4
Caller Gigel result: Gigel4
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel5
Accepter request: Gigel
Caller Ionel result: Ionel5
Accepter result: Gigel6
Caller Ionel parameter: Ionel
Caller Gigel result: Gigel6
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel7
Accepter request: Gigel
Caller Ionel result: Ionel7
Caller Ionel parameter: Ionel
Accepter result: Gigel8
Caller Gigel result: Gigel8
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel9
Caller Ionel result: Ionel9
Accepter request: Gigel
Caller Ionel parameter: Ionel
```

```
Project  AddTask.jav
Run: Main (4)
Caller Ionel parameter: Ionel
Accepter result: Gigel10
Caller Gigel result: Gigel10
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel11
Caller Ionel result: Ionel11
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel12
Caller Gigel result: Gigel12
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel13
Caller Ionel result: Ionel13
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel14
Caller Gigel result: Gigel14
Caller Gigel parameter: Gigel
Accepter request: Ionel
Accepter result: Ionel15
Caller Ionel result: Ionel15
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel16
Caller Gigel result: Gigel16
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel17
Caller Ionel result: Ionel17
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel18
Caller Gigel result: Gigel18
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel19
Caller Ionel result: Ionel19
Accepter request: Gigel
Git  Find  Run  TODO  Problems
Build completed successfully in 3 sec, 247 ms (13 minutes ago)
```

2. Mai jos am implementat problema producator-consumator utilizand channels.

```
task1\Main.java  Entry.java  task1\Port.java  ProducerConsumer.java  Producer.java
package com.home.lab10.task2;

public class Consumer implements Runnable {
    private int c = 0;
    private ProducerConsumer<String, String> pc;

    public Consumer(ProducerConsumer<String, String> pc) {
        this.pc = pc;
    }

    public void run() {
        try {
            while (true) {
                String request = pc.accept();
                System.out.println("Accepter request" + ": " + request);
                String result = request + (++c);
                System.out.println("Accepter result" + ": " + result);
                pc.reply(result);
            }
        } catch (InterruptedException e) {
        }
    }

}
```

Draghici Andreea-Maria
CR 3.1B

```java
package com.home.lab10.task2;

public class Producer implements Runnable {
    private ProducerConsumer<String, String> pc;
    private String id;

    public Producer(ProducerConsumer<String, String> pc, String s) {
        this.pc = pc;
        id = s;
    }

    public void run() {
        try {
            for (int i = 0; i < 10; i++) {
                System.out.println("Caller " + id + " parameter: " + id);
                String result = pc.call(id);
                System.out.println("Caller " + id + " result: " + result);
            }
        } catch (InterruptedException e) {
        }
    }

}
```

```java
package com.home.lab10.task2;


public class ProducerConsumer<R, P> {
    private CallMsg<R, P> cm;
    private Port<CallMsg<R, P>> cp = new Port<CallMsg<R, P>>();

    public P call(R req) throws InterruptedException {
        Channel<P> clientChan = new Channel<P>();
        cp.send(new CallMsg<R, P>(req, clientChan));
        return clientChan.receive();
    }

    public R accept() throws InterruptedException {
        cm = cp.receive();
        return cm.request;
    }

    public void reply(P res) throws InterruptedException {
        cm.replychan.send(res);
    }

    private class CallMsg<R1, P1> {
        R1 request;
        Channel<P1> replychan;

        CallMsg(R1 m, Channel<P1> c) {
            request = m;
            replychan = c;
        }
    }

}
```

Draghici Andreea-Maria
CR 3.1B

```java
public class Consumer implements Runnable {
    private int c = 0;
    private ProducerConsumer<String, String> pc;

    public Consumer(ProducerConsumer<String, String> pc) {
        this.pc = pc;
    }

    public void run() {
        try {
            while (true) {
                String request = pc.accept();
                System.out.println("Accepter request" + ": " + request);
                String result = request + (++c);
                System.out.println("Accepter result" + ": " + result);
                pc.reply(result);
            }
        } catch (InterruptedException e) {
        }
    }

}
```

```java
import java.util.LinkedList;
import java.util.Queue;

public class Port<T> {
    Queue<T> queue = new LinkedList<T>();
    int ready = 0;

    public synchronized void send(T v) {
        queue.add(v);
        ++ready;
        notifyAll();
    }

    public synchronized T receive() throws InterruptedException {
        while (ready == 0) {
            wait();
        }
        --ready;
        return queue.remove();
    }

}
```

```java
package com.home.lab10.task2;

public class Main {
    static ProducerConsumer<String, String> e = new ProducerConsumer<String, String>();
    static Producer ionel = new Producer(e, s: "Ionel");
    static Producer gigel = new Producer(e, s: "Gigel");
    static Consumer maria = new Consumer(e);

    public static void main(String[] args) {
        Thread tIonel = new Thread(ionel);
        Thread tGigel = new Thread(gigel);
        Thread tMaria = new Thread(maria);

        tIonel.start();
        tGigel.start();
        tMaria.start();
    }

}
```

Draghici Andreea-Maria
CR 3.1B

Output task2:

```
C:\Users\user\.jdks\corretto-16.0.
Caller Gigel parameter: Gigel
Caller Ionel parameter: Ionel
Accepter request: Gigel
Accepter result: Gigel1
Accepter request: Ionel
Accepter result: Ionel2
Caller Gigel result: Gigel1
Caller Gigel parameter: Gigel
Caller Ionel result: Ionel2
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel3
Accepter request: Ionel
Caller Gigel result: Gigel3
Accepter result: Ionel4
Caller Gigel parameter: Gigel
Caller Ionel result: Ionel4
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel5
Accepter request: Ionel
Caller Gigel result: Gigel5
Accepter result: Ionel6
Caller Gigel parameter: Gigel
Caller Ionel result: Ionel6
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel7
Accepter request: Ionel
Caller Gigel result: Gigel7
Accepter result: Ionel8
Caller Gigel parameter: Gigel
Caller Ionel result: Ionel8
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel9
Caller Gigel result: Gigel9
Accepter request: Ionel
Caller Gigel parameter: Gigel
```

```
Caller Gigel parameter: Gigel
Accepter result: Ionel10
Caller Ionel result: Ionel10
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel11
Caller Gigel result: Gigel11
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel12
Caller Ionel result: Ionel12
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel13
Caller Gigel result: Gigel13
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel14
Caller Ionel result: Ionel14
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel15
Caller Gigel result: Gigel15
Caller Gigel parameter: Gigel
Accepter request: Ionel
Accepter result: Ionel16
Caller Ionel result: Ionel16
Caller Ionel parameter: Ionel
Accepter request: Gigel
Accepter result: Gigel17
Caller Gigel result: Gigel17
Accepter request: Ionel
Caller Gigel parameter: Gigel
Accepter result: Ionel18
Caller Ionel result: Ionel18
Accepter request: Gigel
Caller Ionel parameter: Ionel
Accepter result: Gigel19
Caller Gigel result: Gigel19
Accepter request: Ionel
```

Observatii:

Producatorul este emitatorul, iar consumatorul este receptorul, iar clasa Port creeaza o coada prioritara unde datele vor fi stocate si ulterior procesate .