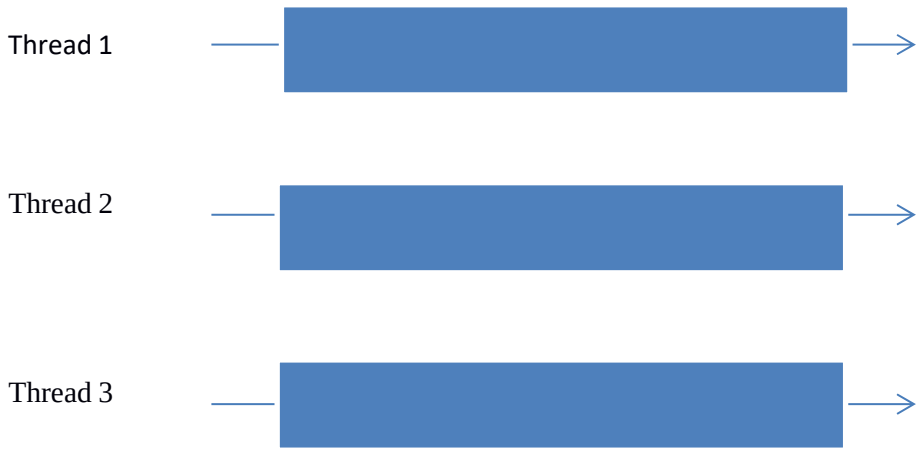
Програм хангамжийн бүтээлт SENG332

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
Хэрэглээний Шинжлэх Ухаан Инженерчлэлийн Сургууль
Лекц, лаборатори: Б.Батням, Д.Энхзол

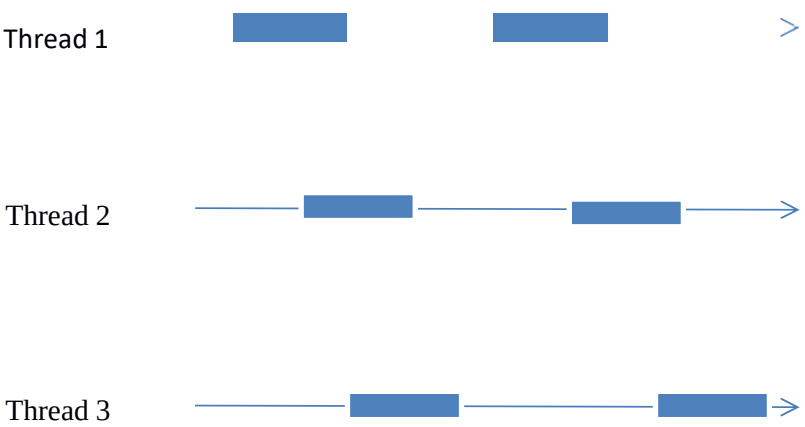
Дасгал ажил 10

“Компьютерийн шинжлэх ухаанд, гүйцэтгэлийн thread гэдэг нь үйлдлийн системийн хуваарьлагчын дангаар нь зохицуулж чадах хамгийн жижиг программчлагдсан зааврууд дараалал юм. Хуваарьлагч нь өөрөө хөнгөн хэмжээний процесс юм. Thread болон процессын хэрэгжүүлэлтүүд үйлдлийн системээс хамааранд ялгаатай байдаг ч ихэнхи тохиолдолд thread процессд агуулагддаг. Нэг процессд олон thread-үүд зэрэг оршиж болох бөгөөд санах ой гэх мэт нөөцүүдийг хуваж ашигладаг. Харин өөр процессууд уг нөөцүүдийг хуваж ашиглаж чадахгүй”.

Зураг 1 дээр олон цөмт процессор олон thread-үүдийг ажиллуулж байгааг харуулав



Зураг 2 нэг цөмт процессор олон thread-үүдийг ажиллуулж байгааг харуулав



testthread.java өөрчлөн гурван thread үүсгэ , thread 1 нь ‘a’ тэмдэгтийг 100 удаа хэвлэнэ, thread 2 нь ‘b’ тэмдэгтийг 100 удаа хэвлэнэ thread 3 нь 1-ээс 100 хүртэлх тоог хэвлэнэ.

Q1: Гурван thread-г үүсгээд эхлүүлэх кодыг бичнэ үү.

```
1  package thread;
2
3  public class TestThread {
4      public static void main(String[] args) {
5          // Create threads
6          PrintLetter thread1 = new PrintLetter('a', 100);
7          PrintLetter thread2 = new PrintLetter('b', 100);
8          PrintNumbers thread3 = new PrintNumbers(100);
9
10         // Start threads
11         thread1.start();
12         thread2.start();
13         thread3.start();
14     }
15 }
```

Q2: Дараах мөр кодуудыг гүйцээнэ үү.

class PrintNum extends

Thread

```
public class PrintNumbers extends Thread {
    private int n;
```

class PrintChar extends

Thread

```
public class PrintChar extends Thread {
    private char letter;
```

Q3: PrintChar ба PrintNum классуудын run() функцүүдийг тодорхойлно уу.

AccountWithoutSync.java | TestThread.java | PrintLetter.java | PrintN

```
1  package thread;
2
3  public class PrintNumbers extends Thread {
4      private int n;
5
6      public PrintNumbers(int number) {
7          n = number;
8      }
9
10     public void run() {
11         for (int i = 1; i <= n; i++)
12             System.out.print(" " + i);
13             System.out.println("\n");
14     }
15 }
16
```

AccountWithoutSync.java | TestThread.java | PrintChar.java | Pri

```java
package thread;

public class PrintChar extends Thread {
    private char letter;
    private int times;

    public PrintChar(char c, int t) {
        letter = c;
        times = t;
    }

    public void run() {
        for (int i = 0; i < times; i++)
            System.out.print(letter);
            System.out.println("\n");
    }
}
```

Q2: TestRunnable.java-д Runnable – интерфэйсийг хэрэгжүүлж гурван thread үүсгэ, thread 1 нь ‘a’ тэмдэгтийг 100 удаа хэвлэнэ, thread 2 нь ‘b’ тэмдэгтийг 100 удаа хэвлэнэ thread 3 нь 1-ээс 100 хүртэлх тоог хэвлэнэ.

TestRunnable.java | PrintChar.java | PrintNumbers.java

```java
package runnable;

public class TestRunnable {
    public static void main(String args[]) {
        // Create threads
        Thread thread1 = new Thread(new PrintChar('a', 100));
        Thread thread2 = new Thread(new PrintChar('b', 100));
        Thread thread3 = new Thread(new PrintNumbers(100));
        //start threads
        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```

TestRunnable.java | PrintChar.java | PrintNumbers.java

```java
package runnable;

class PrintChar implements Runnable {
    private char letter;
    private int times;

    public PrintChar(char c, int t) {
        letter = c;
        times = t;
    }

    public void run() {
        for (int i = 0; i <= times; i++) {
            System.out.print(" " + letter);
        }
    }
}
```

Програм хангамжийн бүтээлт SENG332

TestRunnable.java | *PrintChar.java | PrintNumbers.java

```
1  package runnable;
2
3  class PrintNumbers implements Runnable {
4      private int n;
5
6
7      public PrintNumbers(int number) {
8          n = number;
9      }
10
11     public void run() {
12         for (int i = 0; i <= n; i++) {
13             System.out.print(" " + i);
14         }
15     }
16 }
```

Q3: TestRunnableSleep.java-д PrintNum классыг өөрчлөн дараагийн хэвлэх тоо нь 50-аас их бол тус thread-ийг 1 секүнд унтдаг болго.

```
public class TestRunnableSleep {
        public static void main(String[] args) throws InterruptedException {
                public void run(){
                        for (int i=1; i <= 100; i++){
                                System.out.print(" " + i);
                                try {
                                        if (i>50)
                                                thread.sleep(1000);
                                }
                                catch (InterruptedException ex){
                        }
                        }
                }
        }
}
```

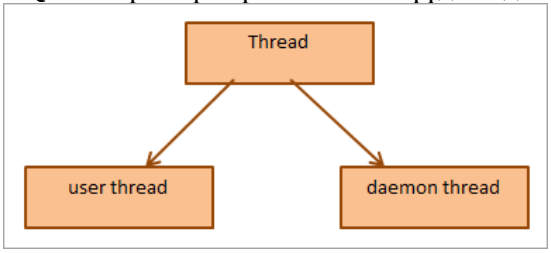
Q4: Q1, Q2, Q3 дах программуудыг ажиллуулахад үр дүн нь тодорхойгүй байна. Яагаад вэ?

Учир нь бид thread ямар дарааллаар execute хийж байгааг баталгаажуулж чадахгүй байгаа.

Q5: Runnable интерфэйсийг хэрэглэх нь Thread классыг ашигласнаас ямар давуу талтай вэ?

Thread бол класс юм. Тиймээс thread-ийг эх класс болгон ашиглавал жавагийн хязгаарлалтын улмаас дахин удамшуулах боломжгүй болно. Харин runnable нь интерфейс учраас дахин удамшуулах боломжтойгоороо давуу талтай.

Q6: Ямар хоёр төрлийн thread-үүд байдаг вэ? Тэдгээрийг ялгаа юу вэ?



Q7: Event-dispatch thread дотор яагаад их цаг зарцуулах үйлдэл гүйцэтгэх ёсгүй вэ?

Учир нь GUI thread-ийг нарийн төвөгтэй ажлуудаар туршиж үзэх нь GUI өөрөө хариу үйлдэл үзүүлэхгүй болоход хүргэдэг.

Програм хангамжийн бүтээлт SENG332
Q4 пакэйж дах MyForm.java-д SwingWorker классыг ашиглан программын ард хийгдэх үйлдлүүдийг гүйцэтгэдэг программ байна. Бид өөрсдийн программын ард ажиллаж цагаан толгойн үсэгнүүдээс санамсаргүйгээр 10 тэмдэгт сонгон тэмдэгтэн цуваа үүсгэх үйлдлийг зохионо.

Q8: SwingAlphabet гэсэн шинэ классыг SwingWorker классаас удамшуулан үүсгэ. Энэ нь private хандалттай цагаан толгойг хадгалах тэмдэгтэн цуваа төрөлтэй гишүүн өгөгдөлтэй байна. doInBackground функц, done функц, process функцүүдийг дахин программчилна. Мөн сонгосон үсэг тус бүрийг хэвлэдэг ба эцэст нь тэмдэгтэн цувааг буцаадаг байна.

```
73    class SwingAlphabet extends SwingWorker<String, Character> {
74        String alphabet = "abcdefghijklmnopqrstuvwxyz";
75
76        @Override
77        protected String doInBackground() {
78            return alphabet;
79        }
80
81        @Override
82        protected void process(List<Character> chunks) {
83        }
84
85        @Override
86        protected void done() {
87        }
88    }
```

Q9: doInBackground() функцийг хэрэгжүүл: alpha гэсэн хувьсагчаас санамсаргүйгээр нэг тэмдэгт сонгож авна, түүнийгээ publish хийж GUI-д дэлгэцлэдэг байна, сонгосон тэмдэгтээ тэмдэгтэн цувааны төгсгөлд залгадаг байна, эцэст нь 1 секүнд унтдаг байна. Энэ үйлдлүүдийг 10 удаа давтаж санамсаргүйгээр үүсгэсэн шинэ тэмдэгтэн цуваг буцаана.

```
        protected String doInBackground() {
            System.err.format("doInBackground isDaemon=%b%n", Thread.currentThread().isDaemon());
            String fakeString = "";
            Random r = new Random();
            for (int i = 0; i < 10; i++) {
                char t = alphabet.charAt(r.nextInt(24));
                super.publish(new Character(t));
                fakeString += t;
                try {
                    Thread.sleep(500);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            return fakeString;

        }
```

Q10: done() функцийг хэрэгжүүлж боловсруулагдсан тэмдэгтэн цуваг TraceBox-д залгадаг болго.

```
        protected void done() {
            try {
                TraceBox.append("done: str="+super.get()+"\r\n");
                } catch (InterruptedException | ExecutionException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                }
                AlphaButton.setEnabled(true);

        }
    }
```

Q11: process() функцийг хэрэгжүүлж сонгогдсон тэмдэгтийг Tracebox-д залгадаг болго.

```
96
97⊖         @Override
98          protected void process(List<Character> chunks) {
99              TraceBox.append("process: ");
00              for (Character i : chunks) {
01                  TraceBox.append(i + " ");
02              }
03              TraceBox.append("\r\n");
04          }
05
```

Q12: Эцэст нь, шинэ SwingAlphabet-г эхлүүлэх товчлуурыг нэмж JFrame –н хэмжээг 800,520 болго.

```
JButton AlphaButton = new JButton("Start SwingAlpha");
              AlphaButton.setFont(font);
              AlphaButton.setPreferredSize(new Dimension(230,30));
              AlphaButton.addActionListener(new ActionListener() {
              @Override
             public void actionPerformed(ActionEvent arg0) {
             BoredButton_Click();
             }
             });
             this.setSize(800,520);
             public void AlphaButton_Click(){
                    SwingAlphabet m = new SwingAlphabet();
                    m.execute();
                    AlphaButton.setEnabled(false)
}
```

Програм хангамжийн бүтээлт SENG332

Q13: Q5 пакэйж дэх AccountWithoutSync.java программыг ажигла, хэрэглэгч тус бүр нь 1 цент нэмдэг 100 thread үүсгэхийг хүсч байна. 100 thread ажиллаж дууссаны дараа үлдэгдэл 100 болсон байх ёстой.

```
package Q1;

public class AccountWithoutSync {
    private Account account = new Account();     private
Thread[] thread = new Thread[100];

   public static void main(String[] args) {
    AccountWithoutSync test = new AccountWithoutSync();
      System.out.println("What is balance ? " +  test.account.getBalance());     }
  public AccountWithoutSync(){
                    ThreadGroup g = new ThreadGroup("group");
                  boolean done = false;
                    for(int i =0 ; i< 100; i++)
              {
                      thread[i] = new Thread(g,new AddAPennyThread(), "t");                    thread[i].start();
              }

            while (!done)                  if(g.activeCount()
== 0)
                                    done = true;
  }
```

```
    // An inner class of task for adding a penny to the account     class AddAPennyThread
  extends Thread {
        public void run() {
  account.deposit(1);
       }    }
     class Account {
        private int balance = 0;
        public int getBalance() {
  return balance;
       }

       public void deposit(int amount) {

          int newBalance = balance + amount;

          try {
             Thread.sleep(5);
          } catch (InterruptedException ex) {
             // do nothing
          }

          balance = newBalance;
       }
    }
  }
```

Q13-a: AccountWithoutSync.java программыг ажиллуулна уу, үлдэгдэл хэд байна вэ?

```
<terminated> AccountWithoutSync [Java Appli
What is balance ? 1
```

Q13-b: Асуудал юунд байна вэ?

Объект нь өөр өөр thread-ээр гэмтсэн олон хэлхээний дунд хуваалцагдсан.

Q13-c: Үүнийг хэрхэн засах вэ?

```
public synchronized void deposit(int amount)
```