Język programowania JAVA

Vincent Van GOGH: *Mężczyzna pijący filiżankę kawy*

Zadanie 1

Napisać program jednowątkowy, który informuje nas o tym, który wątek aktualnie jest wykonywany (wyświetla jego nazwę systemową). Następnie zmienia jego nazwę na nową, wyświetla reprezentację łańcuchową wątku. Następnie w pętli (10 razy): wyświetla wartość licznika i usypia wątek na 10 sekund. Ponieważ wątek jest usypiany należy wprowadzić blok *try* - *catch*, który w razie usiłowania przerwania uśpionego wątku powoduje wypisanie komunikatu "Przerwanie uśpionego wątku".

Zadanie 1 – przykładowe rozwiązanie

```
1  class CurrentThreadDemo
2  {
3    public static void main(String args[])
4    {
5      System.out.println(Thread.currentThread().getName());
6      Thread biezacy = Thread.currentThread();
7      biezacy.setName("Watek glowny");
8      System.out.println("Biezacy watek: " + biezacy);
9      try
10     {
11       for (int n =10; n > 0; n--)
12         {
13           System.out.println("" + n};
14           Thread.sleep(1000);
15         }
16     }
17     catch (InterruptedException e)
18     {
19       System.out.println("Przerwanie uspionego watku");
20     }
21   }
22 }
```

Zadanie 2

Napisać kod klasy MyThread wyprowadzonej z klasy (rozszerzającą) Thread, uruchamiającej dwa wątki, wypisując ich nazwy systemowe.

zadanie 2 – przykładowe rozwiązanie

```
1  public class MyThread extends Thread
2  {
3        public MyThread()
4        {
5            super();
6        }
7        public void run()
8        {
9            System.out.println(getName());
10       }
11       public static void main(String args[])
12       {
13          new MyThread().start();
14          new MyThread().start();
15
16       }
17 }
```

Zadanie 3

Wątkom z Zadania 2 nadać własne nazwy.

Zadanie 3 – przykładowe rozwiązanie

```
1  public class MyThread extends Thread
2  {
3
4        public MyThread(String name)
5        {
6            super();
7            setName(name);
8        }
9        public void run()
10       {
11           System.out.println(getName());
12       }
13       public static void main(String args[])
14       {
15          new MyThread("Watek_1").start();
16          new MyThread("Watek_2").start();
17
18       }
19 }
```

Zadanie 4

Kod z Zadania 3 zmodyfikuj tak, aby w pętli dziesięciokrotnie wypisywał nazwę każdego wątku, z różną częstotliwością.

Zadanie 4 – przykładowe rozwiązanie

```
1  public class MyThread extends Thread
2  {
3      int delay;
4      public MyThread(String name, int delay)
5      {
6          super();
7          setName(name);
8          this.delay = delay;
9      }
10     public void run()
11     {
12         for(int i = 0; i < 10; i++)
13         {
14             System.out.println(getName());
15             try
16             {
17               sleep(delay);
18             }
19             catch(InterruptedException e)
20             {
21             }
22         }
23     }
24     public static void main(String args[])
25     {
26        new MyThread("Pierwszy",2).start();
27        new MyThread("Drugi",1).start();
28     }
29 }
```

Zadanie 5

Zilustruj działanie dwóch wątków z zadania 4 wykorzystując interfejs Runnable

Zadanie 5 – przykładowe rozwiązanie

```
1  public class Main implements Runnable
2  {
3       int delay;
4       String word;
5       public Main(String word, int delay)
6       {
7
8            this.word = word;
9            this.delay = delay;
10      }
11      public void run()
12      {
13           for(int i = 0; i < 10; i++)
14           {
15               System.out.println(word);
16               try
17               {
18                 Thread.sleep(delay);
19               }
20               catch(InterruptedException e)
21               {
22               }
23           }
24      }
25      public static void main(String args[])
26      {
27         Runnable Pierwszy = new Main("Pierwszy", 2);
28         Runnable Drugi = new Main("Drugi", 1);
29         new Thread(Pierwszy).start();
30         new Thread(Drugi).start();
31      }
32 }
```

Zadanie 6

Napisać program, tworzący i uruchamiający dwa wątki o różnych priorytetach, jeden o priorytecie o 2 mniejszym od średniego, a drugi o 2 wiekszym od średniego. Oba wątki są uruchamiane na 10 sekund. W tym czasie zliczają liczbę przebiegów prostej pętli. Po ich zatrzymaniu (przez przypisanie wartości *false* zmiennej sterującej pętli), wątek główny wyświetla dla każdego z nich liczbę zliczanych przebiegów.

Zadanie 6 – przykładowe rozwiązanie (1/2)

```
1  public class clicker implements Runnable
2  {
3        int click = 0;
4        private Thread t;
5        private boolean running = true;
6
7        public clicker (int p)
8        {
9          t = new Thread(this);
10         t.setPriority(p);
11       }
12       public void run()
13       {
14          while (running)
15          {
16             click++;
17          }
18       }
19       public void stop()
20       {
21          running = false;
22       }
23       public void start()
24       {
25         t.start();
26       }
27 }
```

Zadanie 6 – przykładowe rozwiązanie (2/2)

```
1  class HiLoPriority
2  {
3      public static void main(String args[])
4      {
5          Thread.currentThread().setPriority(Thread.MAX_PRIORITY);
6          clicker hi = new clicker(Thread.NORM_PRIORITY + 2);
7          clicker lo = new clicker(Thread.NORM_PRIORITY - 2);
8          lo.start();
9          hi.start();
10         try
11         {
12             Thread.sleep(10000);
13         }
14         catch (Exception e)
15         {
16         }
17         lo.stop();
18         hi.stop();
19         System.out.println(lo.click + "---" + hi.click);
20     }
21 }
```

Zadanie 7

Napisać program, w którym dwa wątki będą niezależnie od siebie modyfikowały wartość jednej zmiennej typu int. Niech bedzie to zwiększanie wartości zmiennej o 1 w każdym przebiegu. Pętla dla każdego wątku wykona się 10 razy.

Zadanie 7 – przykładowe rozwiązanie (1/2)

```
1  public class MyThread extends Thread
2  {
3      private int whichThread;
4      private int delay;
5      private static int account = 0;
6      public MyThread(int whichThread, int delay)
7      {
8          super();
9          this.delay = delay;
10         this.whichThread = whichThread;
11     }
12     public void run()
13     {
14         switch(whichThread)
15         {
16             case 1: thread1(); break;
17             case 2: thread2(); break;
18             case 3: thread3(); break;
19         }
20     }
21     public static void main(String args[])
22     {
23         new MyThread(1, 1).start();
24         new MyThread(2, 2).start();
25         new MyThread(3, 0).start();
26     }
```

Zadanie 7 – przykładowe rozwiązanie (2/2)

```
1   public void thread1()
2   {
3       for (int i = 0; i<10; i++){
4       try{
5             sleep(delay);
6         }
7         catch(InterruptedException e){
8         }
9         account++;
10      }
11  }
12  public void thread2()
13  {
14      for (int i = 0; i<10; i++){
15      try{
16            sleep(delay);
17        }
18        catch(InterruptedException e){
19        }
20        account++;
21      }
22  }
23   public synchronized void thread3()
24  {
25      try{
26            wait(1000);
27      }
28      catch(InterruptedException e){
29          System.out.println(e);
30      }
31      System.out.println(getName() + " " + account);
32  }
33  }
```

Zadanie 8

Napisz program wymuszający wzajemne przerywanie pracy wątków przy modyfikacji wspólnej zmiennej z zadania 7. Jaki jest wynik działania tego programu?

Zadanie 8 – przykładowe rozwiązanie – fragment

```
1   int temp;
2   for (int i=0; i<10; i++){
3           temp= account;
4           try{
5                   sleep(delay);
6               }
7           catch(InterruptedException e){
8               }
9           temp++;
10          account= temp;
11          System.out.println(getName() + "␣" + account);
12      }
```

Zadanie 9

Dokonaj synchronizacji do zmiennej account z zadania 8. Jaki jest wynik działania programu?

Zadanie 9 – przykładowe rozwiązanie (1/3)

```
1  public class MyThread extends Thread
2  {
3      private int whichThread;
4      private int delay;
5      private static int account = 0;
6      private static Object semaphore;
7      public MyThread(int whichThread, int delay)
8      {
9          super();
10         this.delay = delay;
11         this.whichThread = whichThread;
12     }
13     public void run()
14     {
15         switch(whichThread)
16         {
17             case 1: thread1(); break;
18             case 2: thread2(); break;
19             case 3: thread3(); break;
20         }
21     }
22     public static void main(String args[])
23     {
24         semaphore = new Object();
25         new MyThread(1, 1).start();
26         new MyThread(2, 2).start();
27         new MyThread(3, 0).start();
28     }
```

Zadanie 9 – przykładowe rozwiązanie (2/3)

```
1   public void thread1()
2   {  int temp;
3      for (int i = 0; i<10; i++){
4        synchronized(semaphore){
5         temp= account;
6         try{
7                 sleep(delay);
8            }
9         catch(InterruptedException e){
10           }
11        temp++;
12        account= temp;
13       }
14       System.out.println(getName + "␣" + account);
15     }
16  }
17  public void thread2()
18  {  int temp;
19     for (int i = 0; i<10; i++){
20       synchronized(semaphore){
21        temp= account;
22        try{
23                sleep(delay);
24           }
25        catch(InterruptedException e){
26           }
27        temp++;
28        account= temp;
29        }
30       System.out.println(getName + "␣" + account);
31     }
32  }
```

Zadanie 9 – przykładowe rozwiązanie (3/3)

```
1     public synchronized void thread3()
2     {
3         try{
4               wait(1000);
5         }
6         catch(InterruptedException e){
7            System.out.println(e);
8         }
9         System.out.println(getName + "␣" + account);
10    }
11 }
```