# Практические задания

**День 4**. **Многопоточность**

1. **Создать класс расширяющий Thread**

Создать класс NewThread расширяющий Thread.

Переопределить метод run(). В цикле for вывести на консоль символ 100 раз.

Создать экземпляр класса и запустить новый поток.

1. **Создать класс реализующий Runnable**

Создать класс, реализующий интерфейс Runnable.

Переопределить run() метод. Создать цикл for. В цикле распечатываем значения от 0 до 100 делящиеся на 10 без остатка.

Используем статический метод Thread.sleep(), чтобы сделать паузу.

Создать три потока, выполняющих задачу распечатки значений.

1. **Вывод символа в трех потоках**

Необходимо создать три потока, которые изменяют один и тот же объект. Каждый поток должен вывести на экран одну букву 100 раз, и затем увеличить значение символа на 1.

Создать класс расширяющий Thread.

Переопределить метод run() - здесь будет находиться синхронизированный блок кода.

Для того чтобы три объекта потока имели доступ к одному объекту, создаем конструктор принимающий на вход StringBuilder объект.

Синхронизированный блок кода будет получать монитор на объект StringBuilder из пункта 3.

Внутри синхронизированного блока кода выведите StringBuilder на экран 100 раз, а потом увеличьте значение символа на 1.

В методе main() создайте один объект класса StringBuilder, используя символ ‘a’.

Затем создайте три экземпляра объекта нашего класса и запустите потоки.

1. **Создать многопоточную программу вычисления количества простых чисел для диапазона [2, N].**
2. Создайте свою аннотацию Repeat с целочисленным параметром.  
   Расширьте класс ​ [ThreadPoolExecutor](https://docs.oracle.com/javase/10/docs/api/java/util/concurrent/ThreadPoolExecutor.html)​​ и переопределите метод ​*execute* следующим образом: если экземпляр **Runnable** имеет аннотацию **Repeat**, то его метод *run* выполняется несколько раз (количество задается параметром в **Repeat**).  
     
   То есть, написав такой класс:
3. @Repeat(3)
4. **class** **MyRunnable** **implements** **Runnable**{
5. @Override
6. **public** **void** **run**() {
7. System.out.println("Hello!");
8. }
9. }

и использовав его:

1. **public** **static** **void** **main**(String[] strings) {
2. CustomThreadPoolExecutor customThreadPoolExecutor =
3. **new** CustomThreadPoolExecutor(10);
4. customThreadPoolExecutor.execute(**new** MyRunnable());
5. }

Мы должны увидеть:

1. Hello!
2. Hello!
3. Hello!