

6 Лабораторная работа №6 Многопоточность в Java

6.1. Цель работы

Изучить основные приемы работы с потоками в Java и получить практические навыки написания многопоточных Java-приложений.приложений.

6.2. Постановка задачи

Был выдан вариант 2.

В главном классе реализовать статический метод сортировки массива (направление и метод сортировки по варианту) и вывода результата сортировки на экран с помощью `System.out.println(...)`, формат вывода: “имя_текущего_потока: [отсортированный_массив]”. Сигнатура функции сортировки: `printSorted(int[] array)`.

В методе `main` реализовать цикл генерации 3 массивов случайных чисел длины `N` и для каждого сгенерированного массива вызывать метод сортировки в новом потоке. Установить для потоков приоритеты в соответствии с вариантом и после этого запустит потоки в произвольном порядке. Запустить программу несколько раз и пронаблюдать за соблюдением приоритетов.

В методе сортировки заменить вывод всего массива на поэлементный вывод с помощью вызова `System.out.print(...)`, причем массив должен выводиться по 100 элементов в строке и с двух сторон отделяться строками, состоящими из символов «*». Перед строками содержимого массива выводить имя текущего потока. Проверить корректность вывода при запуске этой программы.

Реализовать синхронизацию потоков в момент записи значений в поток вывод. Запустить исправленную программу и проверить корректность вывода.

6.3. Ход выполнения работы

6.3.1. Текст программы

Программа составлена на языке Java.

Листинг 1 — Содержимое файла AbstractApplication.java

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Random;
import java.util.concurrent.CompletableFuture;
import java.util.stream.Collectors;

public class AbstractApplication
{
    protected static final int ARRAY_SIZE = 10000;
    protected static final int THREAD_COUNT = 3;
    protected static final int PRINT_STEP = 100;
    protected static final int[] THREAD_PRIORITIES = { Thread.MAX_PRIORITY, Thread.NORM_PRIORITY,
Thread.MIN_PRIORITY };
    private static final Random gen = new Random();

    public static void main (String... args) throws InterruptedException
    {
        System.out.println("Создание");
        Integer[][] arrays = generateArrays(THREAD_COUNT, ARRAY_SIZE);

        for (int i = 0; i < THREAD_COUNT; ++i)
        {
            var thread = new Thread(new MyTask(arrays[i]));
            thread.setName("Sorter_" + (i + 1));
            thread.setPriority(THREAD_PRIORITIES[i % THREAD_PRIORITIES.length]);
            thread.start();
        }
    }

    // public static void printArray (Class<?> who, Integer[] array)
    // {
    //     Logger.toLog(who, Arrays.stream(array).map(x -> x.toString()).collect(Collectors.joining("; ", "{",
    // "}")"));
    // }

    // public static void printArray (Class<?> who, Integer[] array)
    // {
    //     int i;
    //     for (i = 0; i + PRINT_STEP < array.length; i += PRINT_STEP + 1)
    //     {
    //         Logger
    //             .toLog(
    //                 who,
    //                 Arrays
    //                     .stream(Arrays.copyOfRange(array, i, i + PRINT_STEP))
    //                     .map(x -> x.toString())
    //                     .collect(Collectors.joining("; ", "* ", " *")));
    //     }
    //     Logger
    //         .toLog(
    //             who,
    //             Arrays
    //                 .stream(Arrays.copyOfRange(array, i, array.length))
    //                 .map(x -> x.toString())
    //                 .collect(Collectors.joining("; ", "* ", " *")));
    // }

    public synchronized static void printArray (Class<?> who, Integer[] array)
```

```

{
    int i;
    for (i = 0; i + PRINT_STEP < array.length; i += PRINT_STEP + 1)
    {
        Logger
            .toLog(
                who,
                Arrays
                    .stream(Arrays.copyOfRange(array, i, i + PRINT_STEP))
                    .map(x -> x.toString())
                    .collect(Collectors.joining("; ", "* ", " *")));
    }

    Logger
        .toLog(
            who,
            Arrays
                .stream(Arrays.copyOfRange(array, i, array.length))
                .map(x -> x.toString())
                .collect(Collectors.joining("; ", "* ", " *")));
}

private static Integer[] generateArray (Integer size)
{
    var ret = new Integer[size];
    for (int i = 0; i < ret.length; ++i)
    {
        ret[i] = gen.nextInt();
    }
    return ret;
}

private static Integer[][] generateArrays (int count, int size)
{
    var ret = new Integer[count][];
    List<CompletableFuture<Integer[]>> futures = new ArrayList<>(count - 1);

    for (int i = 0; i < ret.length - 1; ++i)
    {
        futures.add(CompletableFuture.supplyAsync( () -> generateArray(size)));
    }

    ret[0] = generateArray(size);

    for (int i = 0; i < futures.size(); ++i)
    {
        ret[i + 1] = futures.get(i).join();
    }

    return ret;
}

public static Integer[] sort (Integer[] arr)
{
    for (int left = 0; left < arr.length; left++)
    {
        int value = arr[left];
        int i = left - 1;
        for (; i >= 0; i--)
        {
            if (value < arr[i])
            {
                arr[i + 1] = arr[i];
            }
            else
            {
                break;
            }
        }
        arr[i + 1] = value;
    }
    return arr;
}
}

class MyTask implements Runnable

```

```

{
    private final Integer[] array;

    public MyTask (Integer[] array)
    {
        this.array = array;
    }

    @Override
    public void run ()
    {
        AbstractApplication.printArray(this.getClass(), AbstractApplication.sort(array));
    }
}

class Logger
{
    public static void toLog (Class<?> who, String message)
    {
        System.out.println(String.format("[%s/%s] %s", Thread.currentThread().getName(), who.getName(), message));
    }
}

```

6.3.2. Результаты тестирования

Программа была скомпилирована и запущена. Рисунок 1 демонстрирует вывод всех элементов каждого массива на экран.

Рисунок 2 демонстрирует вывод всех элементов каждого массива по 10 элементов в строке на экран. После анализа сделан вывод, что выводимые данные перемешиваются вследствие отсутствия блокировок. Рисунок 3 демонстрирует вывод программы с исправленной функцией вывода данных.

```

less /tmp/output

Создание
[Sorter_1/MyTask] * -147325875; -1474282038; -146781584; -1466697783; -1466628213; -146445852; -146432173; -146181265; -1459937659; -145842538; -145784758; -145736462; -145602728; -145552491; -145521931
1; -1454519875; -1454074958; -1454073828; -144895627; -144876382; -1446808819; -144744417; -144647174; -144647054; -144618222; -144487489; -144429475; -144380898; -144382982; -1437121664; -143576288;
-1433514329; -143497693; -142497068; -1424835701; -1424769059; -1424560742; -1424489935; -142396953; -142365726; -142343415; -141767326; -141732908; -141674519; -141611547; -141381491; -141272463; -21
41257919; -142018154; -141052535; -1410478224; -140379382; -140289131; -140283898; -140077550; -139899753; -139819135; -139708441; -139636076; -139600879; -139533212; -139455643; -139108605; -1390
47787; -138908680; -138868429; -138839580; -138742489; -138725751; -138739137; -138725751; -138535724; -138454181; -138325699; -138082597; -137912611; -137847803; -137699670; -137643104; -137529181; -1375076
94; -137377935; -137322831; -137280884; -137280780; -137102143; -1369898562; -136807659; -136640627; -136560898; -136446878; -136218442; -135817891; -135637287; -135527178; -135381449; -135399541;
-135268923; -135223818; -13523283; -1351214795; -135068644; -135067963; -1350458016; -1347814127; -134701427; -134494745; -134465555; -134343649; -134004744; -133973613; -133939452; -133434628; -21
13379628; -133180943; -133176688; -133170857; -133083566; -133290685; -133293963; -133290173; -133247026; -133209445; -133206925; -133205357; -133190795; -133156738; -1331471541; -1331
41208; -133195551; -133176013; -133106968; -133106394; -133097773; -1330915259; -133085496; -133087783; -1330264820; -1330108975; -132981674; -1329704261; -1329321789; -1329306187; -1329215749; -1329135
681; -1328981387; -1328851460; -1328825937; -1328880856; -1328612488; -1328600158; -1328539958; -132851771; -1328287187; -1328263207; -132817201; -1328179832; -1328163167; -1328088143; -13281726452; -1327789347
; -1327836588; -1326679847; -1326279129; -1326275196; -1326233643; -1326233391; -1326096686; -1325674340; -1325663245; -1325637038; -1325322650; -1325284928; -1325152597; -1325197146; -1325104818; -1325434978; -21
1245931277; -1323966859; -1323689015; -1323687101; -1323639386; -13233419880; -1323359816; -1323253419; -1322924788; -1322779598; -1322725857; -1322666055; -1322804398; -1322082871; -1321938288; -1321
1904824; -1321869299; -1321521954; -1321467585; -1321326916; -1321055584; -1321030516; -1320459268; -1320330946; -1319896020; -1319802859; -13197799562; -1319748238; -1319666696; -1319409435; -1319445695; -131938
2970; -1319341656; -1319298103; -1319089559; -1318886474; -1318819888; -1318599436; -1318589365; -1318504460; -1318612736; -1318042937; -1318039352; -1317923713; -1317727877; -1317746095; -1317481084; -131743766
2; -131737986; -1317347862; -131687760; -131675316; -1316744056; -1316514992; -1316351896; -1316228169; -1316202546; -1316122173; -1316084955; -1315878588; -1315787833; -1315544943; -1315415632; -21
158080034; -1314966460; -1314733281; -1314441557; -1314326578; -1313917214; -1313619879; -1313522299; -1313433633; -1313422133; -1313239886; -1313197263; -1312818875; -1312556411; -1312261652; -1312197548; -21
12605186; -1311977394; -1311747820; -1311593335; -1311392159; -1311376255; -1311852157; -1310749858; -1310486240; -1310413727; -1310379771; -1310248361; -1310215997; -1310152685; -1310076818; -1309938768; -13097
00516; -1309553552; -1309517452; -1309433242; -1309325872; -1308889632; -1308638134; -1308608790; -1308559578; -1308506437; -1308362511; -1308158807; -1307993498; -1307896768; -1307841757; -1307824797; -13077467
73; -1307733685; -1307261579; -1307802870; -1307833488; -1306675491; -1306621380; -1306534145; -1306431909; -1306405211; -1306363489; -1306028313; -1305909986; -1305872627; -1305871980; -1305817524; -1305795540;
-1305752823; -1305717639; -1305348141; -1305140183; -1305138083; -1305079403; -1304998783; -1304869580; -1304713538; -1304687978; -1304597744; -1304388870; -1304295392; -1304211886; -1303854479; -1303796363; -2
107442878; -1303554798; -1303133349; -1302961060; -1302892642; -1302778980; -1302343687; -130231857; -1302276782; -1302124394; -1302096187; -1301980669; -1301522481; -1301125164; -1300727075; -1300721240; -1300
714694; -1300602650; -1300518455; -1300483937; -1300280181; -1300038253; -2099616259; -2099591637; -2099448463; -2099312863; -2099198459; -2098798941; -2098650561; -2098456845; -2098449612; -2098270
838; -2098100412; -2098083518; -2098035858; -2097989852; -20979810887; -2097794147; -2097727183; -2097598738; -2096952233; -2096837933; -2096769815; -2096624546; -2096368547; -2096227558; -2096220198; -2095981072
-2095953808; -2095908012; -2095886664; -2095854348; -2095836462; -2094833693; -2094835882; -2094592637; -2094441586; -2094331858; -2094383968; -2094370713; -2093642918; -2093388724; -2093235553; -2093085114;
-2092949844; -20925904316; -2092889422; -2092653819; -2092454834; -2092380838; -2092335714; -2092241559; -2092226976; -2092217752; -2092067161; -2092025888; -2091984091; -2091880615; -2091626548; -2091608265; -209
1383660; -209152855; -2091611373; -2091045130; -2090779635; -2090719963; -2090618628; -2090504284; -2090497288; -2090380012; -2090368219; -2090350356; -2090319058; -2090232871; -209074835; -209068389; -2089957
5619; -2089552793; -2089383600; -2089195877; -2089151577; -2089015798; -2088986351; -2088861618; -2088832225; -20888188105; -2088829253; -20887545258; -2088751374; -20887184564; -2088712758; -2088709778; -20887
5; -2088528441; -20886171813; -20886096033; -20885871382; -20885695592; -20885407833; -20885232153; -20885192356; -2088488152; -20884722807; -20884580596; -20884514413; -20884505726; -20884164765; -20884101843; -20884666340;
-2088483959; -2088372147; -20883688922; -20883685911; -2088284951; -20882719354; -2088251596; -2088237542; -2088187754; -20881835374; -20881354264; -20881277034; -20881261887; -2088116566; -20880998537; -20880928975; -20
88015972; -2088029132; -208804318; -2088025131; -20880037248; -2079787669; -2079918487; -2079797672; -2079588895; -2079468138; -2079344615; -2079144933; -2079090882; -2079873193; -2079841216; -20797976842; -20789
6855; -2078904944; -2078671144; -2078588985; -2078478252; -2078381845; -2078272828; -2078188984; -20780828528; -2077960737; -2077944052; -2077894083; -2077787598; -2077564761; -2077581527; -2077288263; -2077238
60; -2076987884; -2076757622; -2076604959; -2076541786; -2076517211; -2076116797; -2076038819; -2075994864; -2075959437; -2075727987; -2075551891; -2075544287; -2075443858; -2075387138; -2075284884; -2075282329
-2075065963; -2074889823; -2074877316; -2074799873; -20746595783; -2074492578; -2074145655; -2074081688; -20738399538; -2073783956; -2073437011; -2073258148; -2073201759; -20730859576; -2073047764; -2072828714; -20
07282781; -2072551207; -2072516161; -2072238999; -2072207719; -2072281322; -2072147875; -2072096771; -2072035653; -2071956463; -2071775894; -2071286136; -2071169306; -2071075892; -2070942543; -2070801422; -2070
832588; -2070799392; -2070634581; -2070629941; -2070458584; -2070433472; -2070271704; -20699957159; -20699735143; -2069709775; -20695938816; -2069236999; -2068979862; -2068797445; -2068655228; -20686559784; -2068486
340; -2068279248; -2068056817; -2068042725; -2067782228; -2067633578; -2067585801; -2067464825; -2067416557; -2067392594; -2067287958; -206686496; -2066729558; -206665083; -2066620537; -2066145897; -2066145897; -2066145897;
-2066088323; -2065875864; -2065822956; -2065810669; -2065435814; -2065266684; -2065157294; -2065142794; -20651131214; -2064939782; -2064758313; -2064405518; -2063932809; -2063871747; -2063828961; -21
2063697129; -2063655369; -2063637678; -2063327715; -2063280568; -2063146238; -2062816240; -2062784254; -2062467864; -2062093844; -2062008905; -2061680591; -2061681292; -2061650523; -2061270537; -2061193076; -206
9108282; -2060695758; -2060448498; -2060418077; -2060288049; -2060038128; -2059951639; -2059925445; -2059915447; -2059837828; -2058738855; -2058569868; -2058479853; -2058264884; -205822
2287; -2058061812; -2057445145; -2057244884; -2057233914; -2057088677; -2056985862; -2056823866; -2056763889; -2056642344; -20565926125; -2055796121; -2055390238; -2055351158; -2055260348; -205519662
1; -2054951067; -2054880881; -2054852945; -205453653; -2054375863; -2054154534; -2054082859; -2053990064; -2053844131; -2053671614; -20536347994; -2053427848; -2053259663; -2053170661;
-2053137434; -2052640874; -2052188544; -2052080958; -2051915818; -205195122; -2051615887; -2051530880; -2051280859; -205112554; -2051280201; -2051192758; -2050781952; -2050718462; -2050477785; -2050440854; -20
50361592; -2050332116; -20502879198; -2049899179; -2049860972; -2049765639; -2049624884; -2049619744; -2049554258; -2049318593; -20492448874; -2049241867; -2049114388; -2048824083; -2048431801; -2048384429; -20483
37882; -2048287698; -2047996094; -2047765481; -2047616592; -2047475158; -2047134197; -2047061786; -2046907786; -2046906283; -2046871288; -2046696283; -2046615627; -2046594627; -20459107
11; -2045643833; -2045613314; -2045549698; -2045293263; -2045187695; -2045112856; -2044538181; -2044526325; -2044454611; -2044426541; -2044413463; -2044167248; -2044067102; -2044081296; -2043916135; -2043843218;
/tmp/output

```

Рисунок 1 — Вывод всех элементов массива сразу

```

less /tmp/output

Создание
[Sorter_1/MyTask] * -147075402; -146668861; -1454582844; -145377807; -1446001718; -144533285; -143817533; -143782861; -143432961; -143375803 *
[Sorter_2/MyTask] * -146689742; -145644115; -145459319; -144867622; -144638065; -143718832; -143131638; -1424740837; -142304424; -142012589 *
[Sorter_3/MyTask] * -147446460; -147404933; -146456838; -146190155; -145573876; -144663813; -1444342362; -143884937; -143510448 *
[Sorter_1/MyTask] * -142456580; -141636582; -141492619; -141348760; -141257957; -141067281; -140977695; -1408398940; -1408342243; -139765404 *
[Sorter_2/MyTask] * -2141124576; -2141061845; -214955272; -214088273; -2139853282; -2137973916; -2137712950; -2136393621; -2136359159 *
[Sorter_1/MyTask] * -138683166; -138651487; -138530437; -138096957; -1380993324; -137821981; -1373735778; -137168768; -136297303; -1358699329 *
[Sorter_2/MyTask] * -142485482; -141257358; -140999780; -139340418; -139071680; -138818534; -138152855; -137638213; -137598454 *
[Sorter_1/MyTask] * -213151529; -213477191; -213452576; -213446346; -213426790; -2133416977; -213315664; -2130223692; -2129285285; -2128029090 *
[Sorter_2/MyTask] * -131480899; -1313987875; -131333566; -1313836954; -1312496897; -131245680; -1312605763; -1313828925; -131364157; -1313124825 *
[Sorter_3/MyTask] * -136836679; -135754613; -134211677; -133768595; -131389728; -131329686; -1313266681; -1308335918; -130842487; -1308015540 *
[Sorter_1/MyTask] * -132868340; -132849670; -132848561; -1327790518; -132771678; -1327471294; -1326954255; -1326872431; -1326595847; -1326542214 *
[Sorter_2/MyTask] * -2129973161; -2129812928; -2129586462; -2129378658; -2128834395; -2127852393; -2127846956; -2127389809; -2126713148 *
[Sorter_1/MyTask] * -2125925485; -2125815525; -2125259157; -212527151; -2124494860; -2124453122; -2121926517; -2121477895; -2121314943; -2120644789 *
[Sorter_2/MyTask] * -2127653848; -2127571204; -212788206; -2127806958; -2126661601; -2124088363; -2123824580; -2122779107; -2120832281; -2120688889 *
[Sorter_1/MyTask] * -2119288595; -2118684995; -2118021766; -2118013465; -2117767223; -2117491864; -2117449398; -2116902381; -2116778285; -2116527835 *
[Sorter_2/MyTask] * -2126405751; -2126021964; -2125939753; -2125890338; -2125608583; -2124996889; -2124859253; -2124497119; -2124409283; -2124256833 *
[Sorter_3/MyTask] * -2128309126; -2128084344; -2119697896; -2119245513; -2118213886; -2118145804; -2118062524; -2117173557; -2116844853; -2116763057 *
[Sorter_1/MyTask] * -2123814987; -2123439818; -2122243938; -2121939275; -2120763337; -2120644843; -2120496482; -2119841922; -2118953571; -2118190478 *
[Sorter_2/MyTask] * -2114809296; -2114640764; -2114471699; -2113682584; -211377368; -2113568285; -2113051852; -2112934897; -2111779999; -2111236472 *
[Sorter_3/MyTask] * -2118046182; -2117735952; -2117586359; -2117468758; -2117081711; -2116803238; -2116663831; -2116519582; -2115312065; -2115248980 *
[Sorter_1/MyTask] * -2115948598; -211547716; -2115367998; -2114963380; -2114739455; -2114304144; -2114313617; -2114085195; -2113897510; -2113209994 *
[Sorter_2/MyTask] * -2118768598; -2118597892; -210959357; -2109505987; -2108881552; -2108808561; -2108582444; -2105992431; -2105990652; -2105384621 *
[Sorter_3/MyTask] * -2115159898; -2115034373; -2114021197; -2113508388; -2112246595; -2111563927; -2111509822; -2111259516; -2110516277 *
[Sorter_1/MyTask] * -2110381306; -2110080936; -2109881042; -2109561823; -2109412713; -2108194123; -2107992732; -2107227853; -2107210837; -2106513748 *
[Sorter_2/MyTask] * -2105160410; -2105065157; -2105283175; -210222567; -2102119362; -2101810891; -2101707831; -2101565224; -2101343885 *
[Sorter_3/MyTask] * -2110167762; -2109821066; -2109593820; -2108423663; -21077973613; -2107747487; -2107645238; -2106975138; -2106631943; -2106208276 *
[Sorter_1/MyTask] * -2108044805; -2106552685; -210552685; -2105488413; -2105437366; -2104789862; -2104325659; -2104115393; -2104101058; -2104057842 *
[Sorter_2/MyTask] * -2108968874; -2108758378; -2108551428; -2108250883; -2108027800; -2108028463; -2099943688; -2099596386; -2098981482; -2098342677 *
[Sorter_3/MyTask] * -2105861156; -2105658832; -2105575549; -2105447482; -2104851685; -2104402595; -2103831437; -210357
```

```

less /tmp/output
[Sorter_1/MyTask] * 2050947812; 2051091000; 2051500340; 2051601677; 2051700085; 2053220257; 2053381797; 2053893080; 2053895295; 2054187176 *
[Sorter_1/MyTask] * 2054592377; 2054961985; 2055101251; 2055668382; 2056438517; 2056844205; 2056849068; 2057130833; 2058070719; 2059344310 *
[Sorter_1/MyTask] * 2059849942; 2060033063; 2060084507; 2060533866; 2060987595; 2061263427; 2061496636; 2061769121; 2062775797; 2063071613 *
[Sorter_1/MyTask] * 2063480615; 2063547846; 2063720405; 2064052888; 2064521273; 2064579356; 2064664282; 2064980601; 2065527041; 2065845615 *
[Sorter_1/MyTask] * 2065202269; 2065855863; 2066945899; 2069053149; 2069787907; 2070449324; 2070509636; 2072400254; 2072560621; 2073669469 *
[Sorter_1/MyTask] * 2074304498; 2075815902; 2076270055; 2076386087; 2076541121; 2076732835; 2076814796; 2076118149; 2078946133; 2079442965 *
[Sorter_1/MyTask] * 2079832561; 2080401249; 2080570285; 2080697609; 2081070129; 2081653466; 2081692034; 2081839495; 2082056065; 2083019924 *
[Sorter_1/MyTask] * 2084777309; 2085218832; 2085221365; 2085267898; 2085391040; 2085486421; 2087146398; 2087248202; 2088249264; 2088447267 *
[Sorter_1/MyTask] * 2088741981; 2088765624; 2089316900; 2089762878; 2090806050; 2091150117; 2091678584; 2091895133; 2091904630; 2092666589 *
[Sorter_1/MyTask] * 2093005498; 2093111820; 2093433196; 2093443108; 2094088939; 2094757190; 2095066518; 2095846915; 2096826625; 2097225059 *
[Sorter_1/MyTask] * 2098979946; 2099244153; 2099353910; 2099584650; 2100442257; 2100573638; 2100961231; 2101399612; 2101915809; 2102282182 *
[Sorter_1/MyTask] * 2103617691; 2103713261; 2103946825; 2104893352; 2105112943; 2105748611; 2105973078; 2106299767; 2106964748; 2107672702 *
[Sorter_1/MyTask] * 2108274505; 2109111929; 2109122712; 2109148947; 2109618027; 2109667949; 2111009952; 2111666467; 2111992193; 2112217110 *
[Sorter_1/MyTask] * 2113120319; 2114067536; 2115055509; 2115237857; 2115941150; 2116532693; 2117537651; 2118047939; 2119343144; 2119390888 *
[Sorter_1/MyTask] * 2119832527; 2120094305; 2120338538; 2121263487; 2121839292; 2121997240; 2122605993; 2122967608; 2123002350; 2123716295 *
[Sorter_1/MyTask] * 2124140444; 2125757309; 2126383040; 2127265205; 2127704461; 2128018060; 2128282331; 2129132514; 2129240209; 2129595505 *
[Sorter_1/MyTask] * 2130424820; 2131167667; 2131220694; 2132447685; 2132691377; 2133024251; 2133281218; 2133420768; 2134457258; 2135268572 *
[Sorter_1/MyTask] * 2135852497; 2136934166; 2137221228; 2137301266; 2137385527; 2137702950; 2138587524; 2138795327; 2140089675; 2140621528 *
[Sorter_1/MyTask] * 2140992928; 2143196135; 2143223179; 2144337728; 2145355510; 2145866150; 2145930500; 2146156538; 2146261848; 2146488531 *
[Sorter_1/MyTask] * 2147192821 *
[Sorter_2/MyTask] * -2147470838; -2147427082; -2146906598; -2146743161; -2146398551; -2145540470; -2144729428; -2144663911; -2144181807; -2144154072 *
[Sorter_2/MyTask] * -2143500590; -2143300018; -2142843519; -2142293883; -2142214389; -2142080527; -2141712878; -2141667512; -2141552190; -2141280615 *
[Sorter_2/MyTask] * -2140720539; -2140681537; -2140293020; -2139990593; -2137102888; -2136765760; -2136590008; -2136585572; -2135944486; -2135463758 *
[Sorter_2/MyTask] * -2134978593; -2134464256; -2134406226; -2134104108; -2133636208; -2133223916; -2133045695; -2132852866; -2132740756; -2132650004 *
[Sorter_2/MyTask] * -2132013800; -2131450649; -2129473079; -2128432952; -2127776664; -2127697051; -2127451117; -2127114085; -2127094494 *
[Sorter_2/MyTask] * -2126750440; -2125669917; -2125603805; -2124954123; -2124593398; -2124207052; -2123995714; -2123693608; -2123573127; -2122774226 *
[Sorter_2/MyTask] * -2122024246; -2122001645; -2121837615; -2121508016; -2121137864; -2121072809; -2120782389; -2120246789; -2120130202; -2119706282 *
[Sorter_2/MyTask] * -2118892664; -2118500492; -2118198587; -2118031304; -2117658477; -2117224759; -2116516610; -2116139099; -211589310; -2115707678 *
[Sorter_2/MyTask] * -2114534931; -2114272727; -2113873854; -2113111779; -2112927198; -2112838523; -2112968608; -2112176520; -2111813456; -2111625790 *
[Sorter_2/MyTask] * -2110176570; -2109298269; -2108519778; -2108300431; -2108206149; -2107273067; -2107201044; -2106817218; -2106718921; -2105857476 *
[Sorter_2/MyTask] * -2105435089; -2105169669; -2104970787; -2104540826; -2104460189; -2104343644; -2103391846; -2103352200; -2103194268; -2102533897 *
[Sorter_2/MyTask] * -2101867014; -2101580892; -2101319839; -2100111427; -2099358268; -2098607553; -2098423794; -2098044253; -2097754003; -2097587475 *
[Sorter_2/MyTask] * -2096799018; -2094409202; -2094297190; -2094059949; -2093352127; -2093023540; -2092904044; -2092324359; -2091798809; -2089701484 *
[Sorter_2/MyTask] * -2088390564; -2088248719; -2087771486; -2087553995; -2087258763; -2086931078; -2086593745; -2086084727; -2085952879; -2085950591 *
[Sorter_2/MyTask] * -2085455381; -2083624329; -2083305108; -2082915561; -2081989425; -2081208813; -2081165641; -2081081387; -2079733153; -2078640017 *
[Sorter_2/MyTask] * -2077037067; -2076955868; -2076557587; -2075364183; -2075294471; -2074746430; -2074504207; -2074355186; -2073764458; -2073379832 *
[Sorter_2/MyTask] * -2072847409; -2071945337; -2071820275; -2071350100; -2071289689; -2070535129; -2070000149; -2069355837; -2069314497; -2068905420 *
[Sorter_2/MyTask] * -2067976735; -2067909390; -2067722726; -2067342056; -2066630416; -2066587940; -2066364947; -2066094295; -2066070298; -2065751571 *
[Sorter_2/MyTask] * -2064179715; -2063754859; -2062836459; -2062326768; -2062253164; -2062140758; -2061484148; -2061264069; -2061167065 *
[Sorter_2/MyTask] * -2059685673; -2059148713; -2059039411; -2059037059; -2058993979; -2058669434; -2058658695; -2058470384; -2058025572; -2057533186 *
[Sorter_2/MyTask] * -2056690977; -2056933908; -2056378642; -2055311533; -2055295353; -2055090012; -2054183619; -2053990946; -2053395505; -2053392328 *
[Sorter_2/MyTask] * -2053249627; -2052614220; -2052414150; -2052183412; -2052032532; -2051526969; -2051402171; -2050678992; -2049701954; -2049640196 *
[Sorter_2/MyTask] * -2049631214; -2049516139; -2049088026; -2048493796; -2048315795; -2048304577; -2048240201; -2048159906; -2046947386; -2046750949 *
[Sorter_2/MyTask] * -2046332523; -2045953696; -2045440474; -2044725304; -20446112934; -2044555098; -2044213055; -2044141513; -2043580987; -2043555030 *
[Sorter_2/MyTask] * -2043061376; -2041840856; -2041798455; -2040850210; -2039671673; -2039223077; -2038976656; -2038777574; -2037779787; -2037691816 *
[Sorter_2/MyTask] * -2036634186; -2036452941; -2035814311; -2035729108; -2035225790; -2035092965; -2034143655; -2034002974; -2033245345; -2032675670 *

```

Рисунок 3 — Исправленный вывод элементов массива по 10 на строчку

Вывод

При выполнении данной работы были получанны навыки созданий многопоточных приложений с помощью средств языка Java. В процессе работы были использованы следующие средства: конструкторы потоков, создание блоков-блокировок с помощью ключевого слова `synchronized`.