

# OSHW1-实验报告-20301174-万兴全

---

## 1.实验要求&实验内容

---

Write a multithreaded Java, Pthreads, or Win32 program that outputs prime numbers.

This program should work as follows: the user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

Submit: report (in word or pdf) and source code of your program.

编写输出素数的多线程 Java、Pthreads 或 Win32 程序。

该程序应按如下方式工作：用户将运行该程序并在命令行上输入一个数字。然后程序将创建一个单独的线程，输出所有小于或等于用户输入的数字的素数。

## 2.交付物

---

报告（pdf 格式）和程序的源代码。

## 3.采用技术

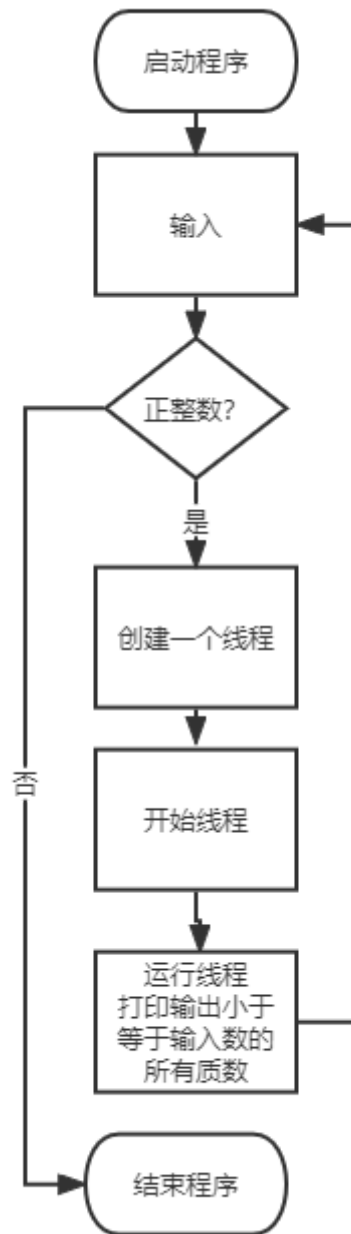
---

Java, c++ Pthread

## 4.程序介绍

---

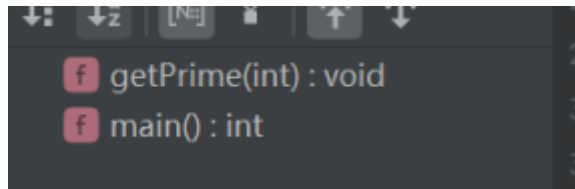
### 4.1程序流程



## 4.2程序结构

```
✓ C ◦ AllPrimeNumber
  > Runnable
    m ◦ AllPrimeNumber(String, int)
    m start(): void
    f t: Thread
    f threadName: String
    f number: int
```

```
✓ C ThreadDemo
  m main(String[]): void
```



## 5.关键代码

---

### 5.1求素数

#### Java

```
System.out.println(number+"以内的素数有: ");
try {
    for (int i = 2; i <= number; i++) {
        boolean isPrime = true;
        for (int j = 2; j < i; j++) {
            if (i % j == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            System.out.print(i + " ");
        }
    }
    Thread.sleep(0);
}
```

#### c++Pthread

```
void getPrime(int number) {
    sleep(1);
    int i, j, isPrime;
    for (i = 2; i <= number; i++) {
        isPrime = 0;
        for (j = 2; j < i; j++) {
            if (i % j == 0) {
                isPrime = 1;
                break;
            }
        }
        if (isPrime == 0) {
            std::cout << i << " ";
        }
    }
}
```

```
std::cout << std::endl;
}
```

## 5.2建立新的线程

### Java

```
if (a > 0) {
    AllPrimeNumber primeNumber = new AllPrimeNumber("Thread " +
i, a);
    primeNumber.start();
} else {
    System.out.println("非法输入，程序终止");
    return;
}
```

### c++ Pthread

```
if (a > 0) {
    pthread_create(&pid, nullptr, reinterpret_cast<void *(*)(void
*)>(getPrime), reinterpret_cast<void *>(a));
    std::cout << "线程-" << pid << "已建立\n" << a << "以内的素数有：
" << std::endl;
} else {
    printf("非法输入，程序终止");
    return 0;
}
```

## 6.程序运行截图

---

```
运行: ThreadDemo x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" -javaagent:C:\JetBrains\apps\IDEA-U\ch-0\222.41
请输入一个正整数, 非法输入将退出程序:
100
创建线程Thread 1
开始线程Thread 1
运行线程Thread 1
100以内的素数有:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
-----
请输入一个正整数, 非法输入将退出程序:
97
创建线程Thread 2
开始线程Thread 2
运行线程Thread 2
97以内的素数有:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
-----
请输入一个正整数, 非法输入将退出程序:
5
创建线程Thread 3
开始线程Thread 3
运行线程Thread 3
5以内的素数有:
2 3 5
-----
请输入一个正整数, 非法输入将退出程序:
10
创建线程Thread 4
开始线程Thread 4
运行线程Thread 4
10以内的素数有:
2 3 5 7
-----
请输入一个正整数, 非法输入将退出程序:
~3
非法输入, 程序终止

进程已结束,退出代码0
|
```

```
运行: OSHW1 x
D:\CodeWorkplace\CLionProject\OSHW1\cmake-build-debug\OSHW1.exe
请输入一个正整数, 非法输入将导致退出程序:100
线程-2已建立
100以内的素数有:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
请输入一个正整数, 非法输入将导致退出程序:30
线程-3已建立
30以内的素数有:
2 3 5 7 11 13 17 19 23 29
请输入一个正整数, 非法输入将导致退出程序:97
线程-4已建立
97以内的素数有:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
请输入一个正整数, 非法输入将导致退出程序:~3
非法输入, 程序终止

进程已结束,退出代码0
```