

SIMD
MIMD
SISD → typical program
MISD

```

class MyThread1
{
    run()
}
class MyThread2
{
    run()
}

myThread1 t1 = new MyThread1
           t2 = new MyThread2

t1.start();
t2.start();
;
    
```

MIMD

mutually exclusive
↳
Mutex mu

```

class Data {
public:
    Data(int a, int b) {
        x = a;
        y = b;
    }
    void f() {
        mu.lock();
        int r = rand() % 100;
        x += r;
        y -= r;
        mu.unlock();
    }
    int sum() {
        return (x+y);
    }
private:
    int x;
    int y;
}
    
```

Can't be atomic in JAVA
by the time thread starts constructor has already finished

not atomic
can stop anywhere in the thread

mu.lock(); int r = x+y; mu.unlock(); return r;
not atomic
can be split into multiple steps

```

Data *d = new Data(50,50);
for (int i = 0; i < 100; i++) {
    d->f();
    cout << d->sum() << endl;
}
    
```

always 100 on single-thread

```

class MyThread : public QThread {
public:
    Data *d; Data d;
    MyThread(Data *x) { d = x; };
    void run() {
        for (int i = 0; i < 100; i++) {
            d->f(); d.f();
            cout << d->sum() << endl;
            d.sum()
        }
    }
}
    
```

will always get 100 each thread has copy of data

```

Data d(50,50);
Data *d = new Data(50,50);
MyThread *t1 = new MyThread(d);
           *t2
           *t3

t1->start();
t3
t1->wait();
t3
    
```

will not always get 100