Actor Pattern

. event-driven receive loop .

Dispatcher , Cluster .

AKKA .

- Active Object :
- :
- : . ( )
- : . ( )
- :
- :

. .

?

=> ,

=>

,

. (Design Patterns : Elements of Reusable Object-Oriented Software)

- GoF(4) .

Erich Gamma 3 . (.)

1.
2.
3.

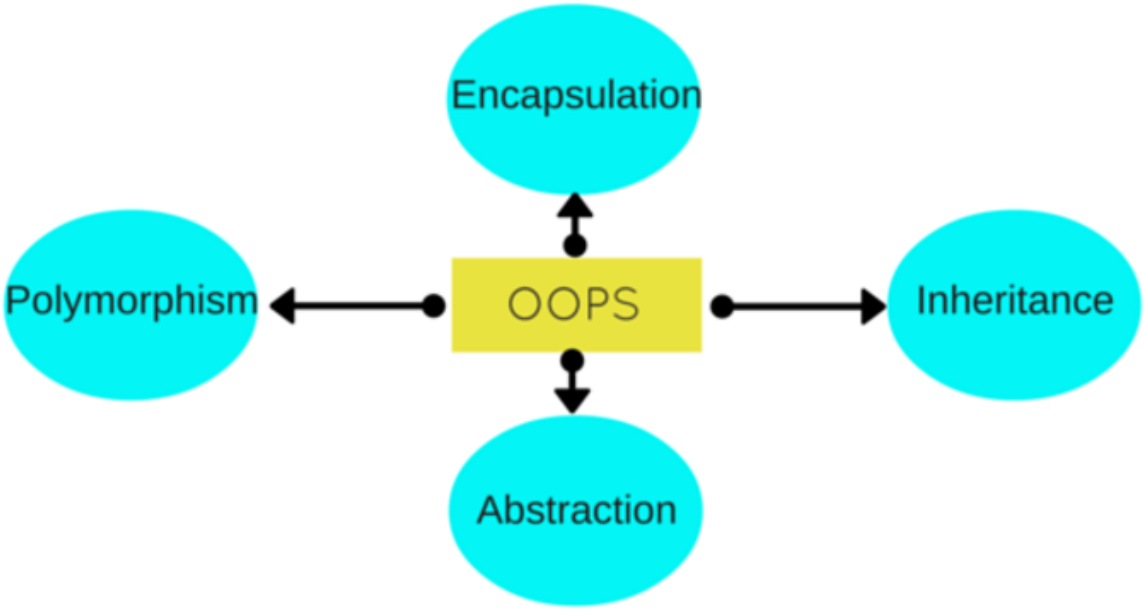
1.
2.
3.

Gof -C++

.

OOP Actor Model

OOP(Object-oriented programming)



OOP

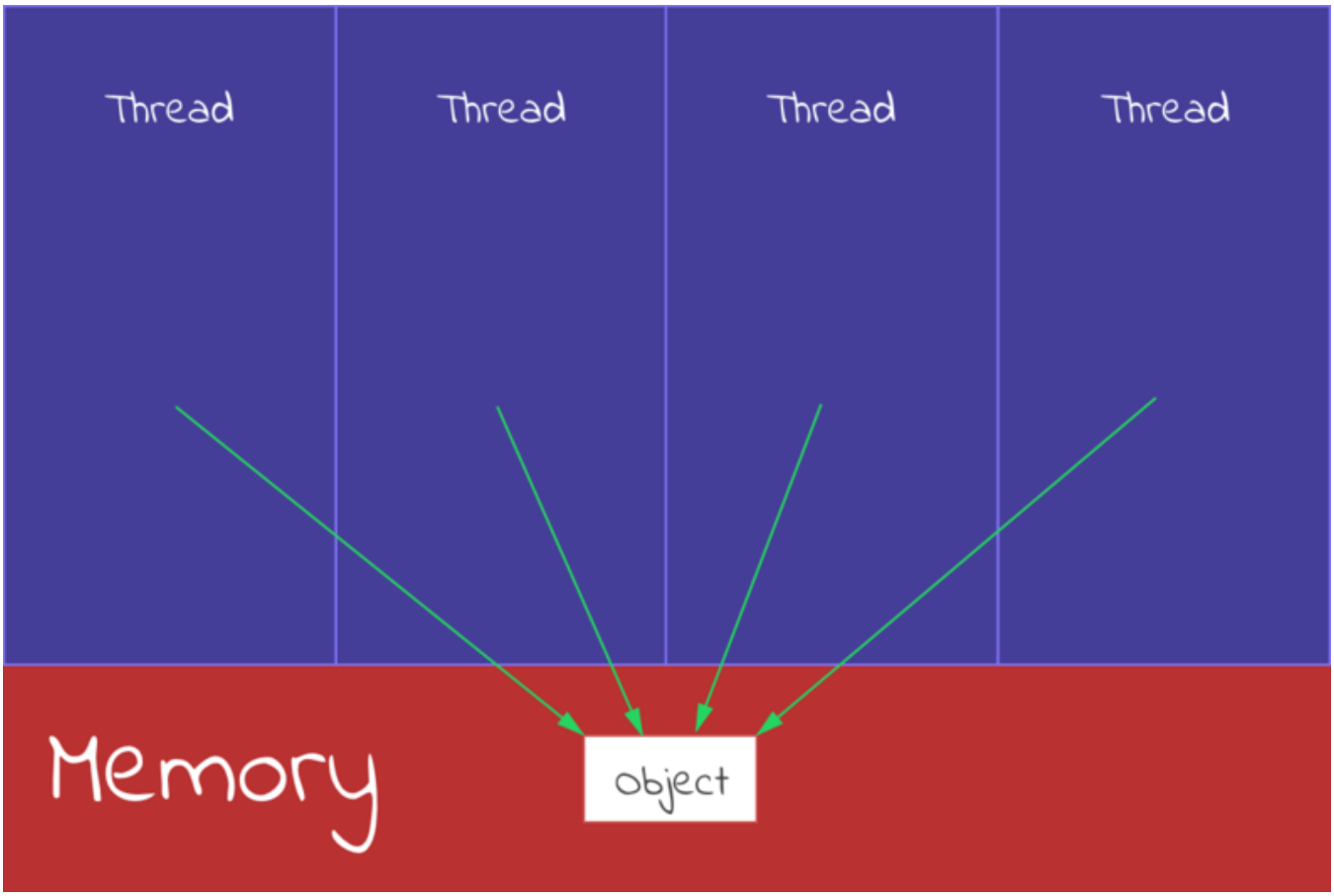
```
    public class DeepThought
    {
        private int state;

        public int State
        {
            get
            {
                lock (this)
                {
                    Thread.Sleep(3000); //    ..? ?
                    return state;
                }
            }

            set
            {
                lock (this)
                {
                    //      ,
                    state = value;
                }
            }
        }
    }
var dt = new DeepThought()

Thread.new { console.log( dt.State ) }
Thread.new { dt.State=1 }
```

OOP

, ' Lock(metex) , , ()' .

OOP Lock ?

, . .



==> Object1 Object2 Lock, Object3 Object1 A   Object3

, (CriticalSection,SpinLock,Mutex,Semaphore)

, .

Actor

, .

(Protected OOP .)

" .

, 'lock' 'synchronized' .

.

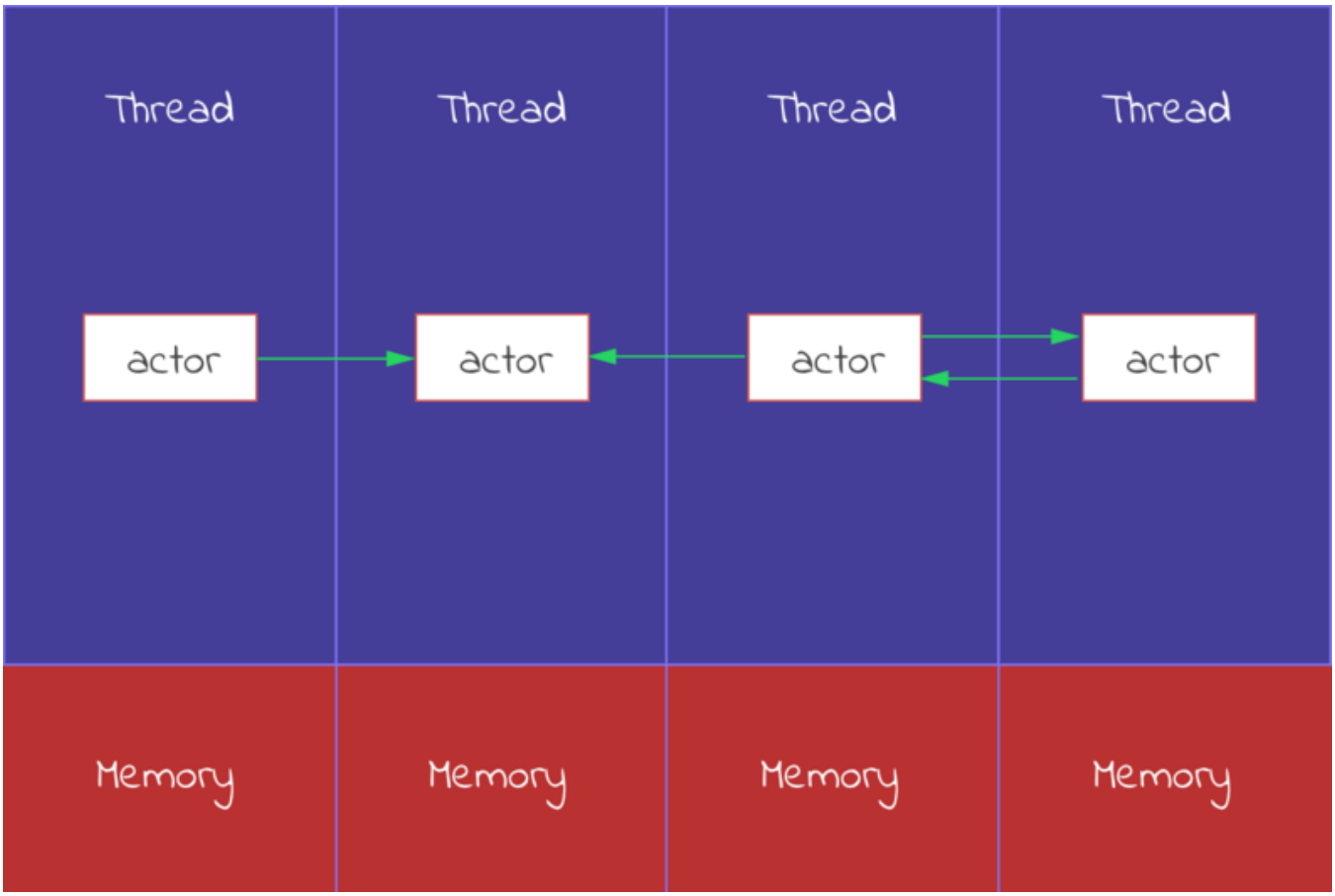
.

Actor

```
public class MyActor: ReceiveActor
{
  //Actor      ()  ,   priave .
  private readonly ILoggingAdapter log = Context.GetLogger();
  private int state =0;

  public MyActor()
  {
    Receive<string>(message => {
      log.Info("Received String message: {0}", message);
      Sender.Tell(state );
    });
    Receive<SomeMessage>(message => {...});
  }
}

//    ,,    .
var remoteActor = system.ActorSelection("akka.tcp://MyServer@127.0.0.1:8001/user/someActor");

//  ?(Ask),  ?(Tell),  (Result)?
//          .
// Tell   ,
//        ,    .
var state = remoteActor.Ask("Hello").Result;
```

Actor , .

, .


Thread
Thread
Thread
Thread
actor
actor
actor
actor
Memory
Memory
Memory
Memory

, .

,

, , .

Actor2 Lock .

OOP

Dispatcher , OOP .

OOP ?

, 3

, .

- Future and Promise
- Routing strategy
- MessageDeliveryReliability