
PROGRAMMAZIONE 2

6b. Le eccezioni, operazionalmente

```
class C {  
    public void via( ) {  
        primo( );  
        System.out.println("sei al via");  
    }  
  
    public void primo( ) {  
        secondo( );  
        System.out.println("sei al primo");  
    }  
  
    public void secondo( ) {  
        throw new Exception( );  
        System.out.println("sei al secondo");  
    }  
}
```

Cosa succede con `(new C()).via();`?

Abstract Stack Machine

Workspace

```
(new C()).via();
```

Stack

Heap

Abstract Stack Machine

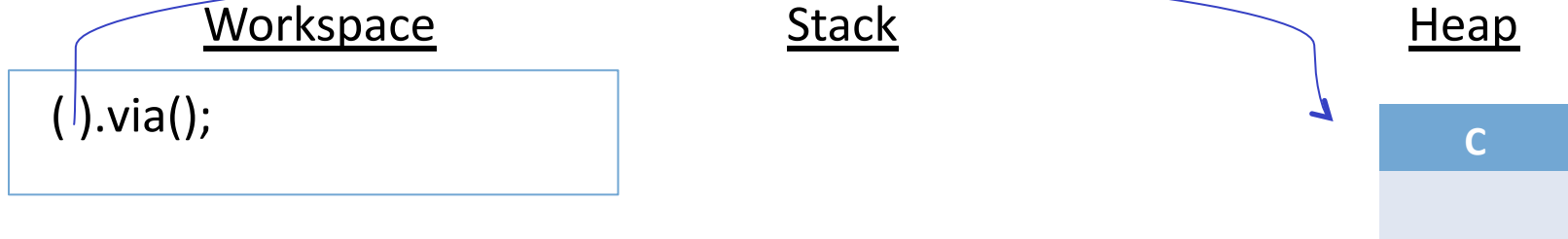
Workspace

```
(new C()).via();
```

Stack

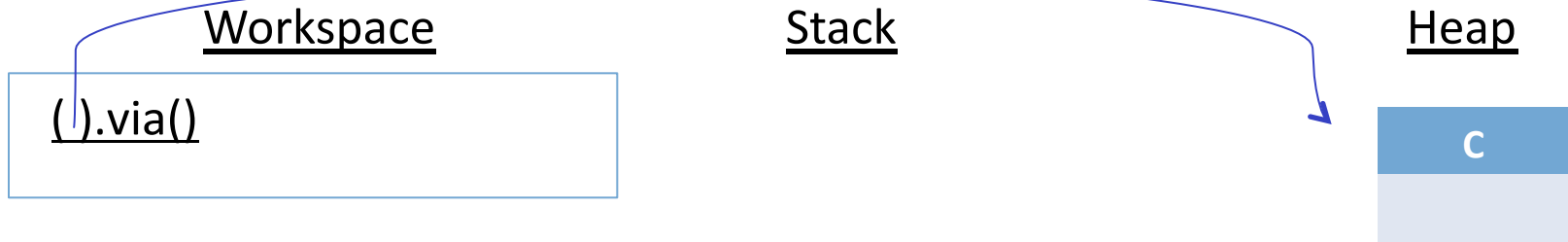
Heap

Abstract Stack Machine

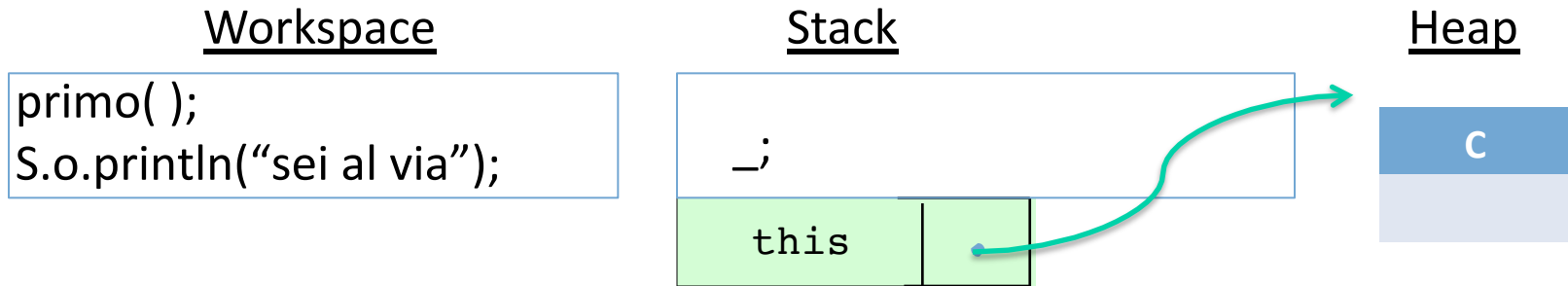


Allocata una istanza della classe C sullo heap

Abstract Stack Machine



Abstract Stack Machine



Viene salvato sullo stack la **continuazione** (cosa eseguire) dopo aver invocato "via"

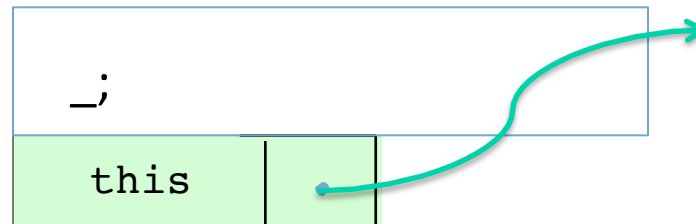
Viene salvato sullo stack anche il valore corrente di `this`

Abstract Stack Machine

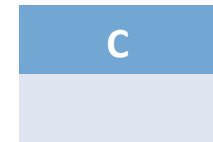
Workspace

```
primo();  
S.o.println("sei al via");
```

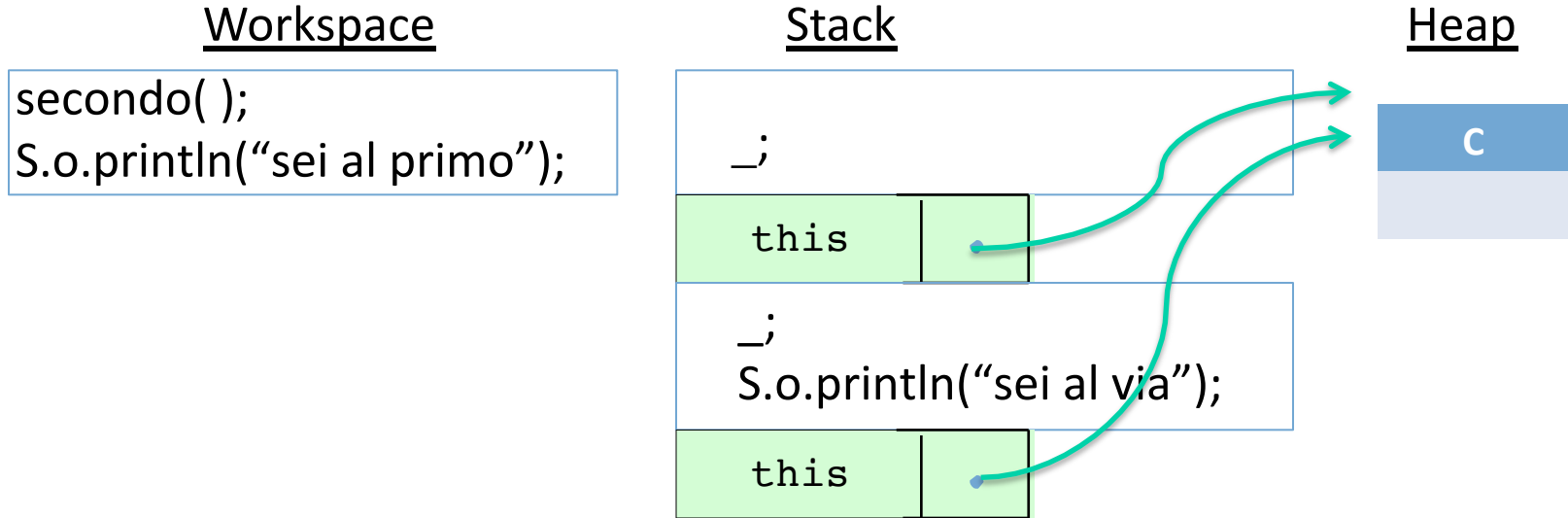
Stack



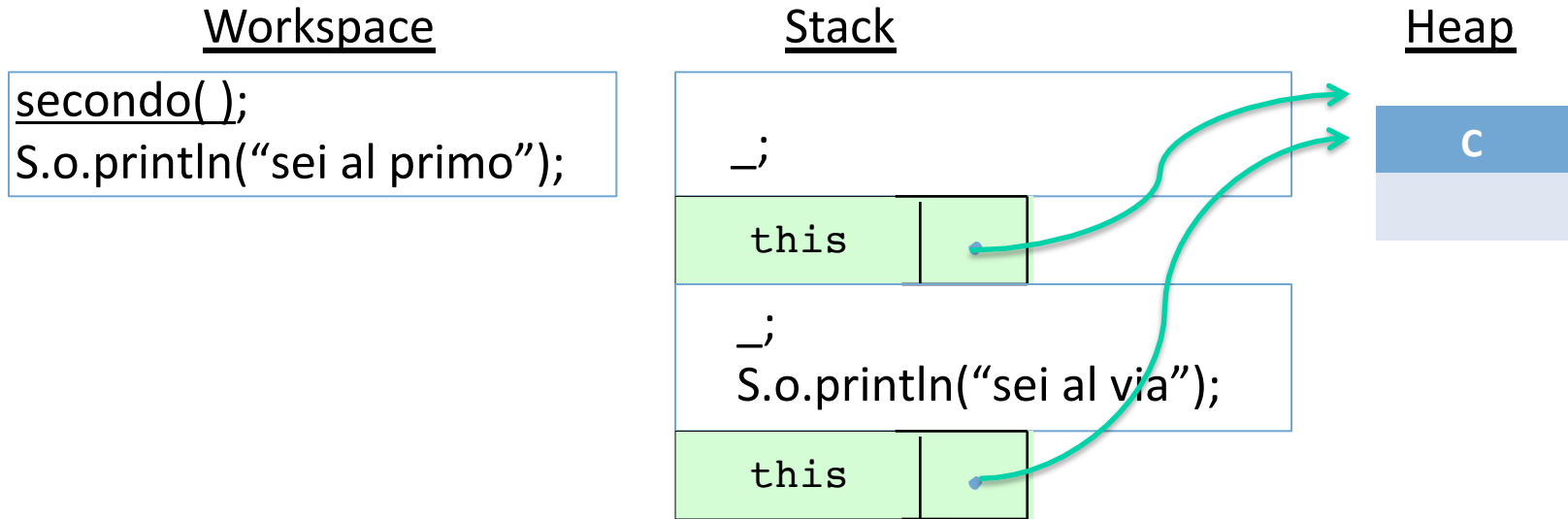
Heap



Abstract Stack Machine



Abstract Stack Machine

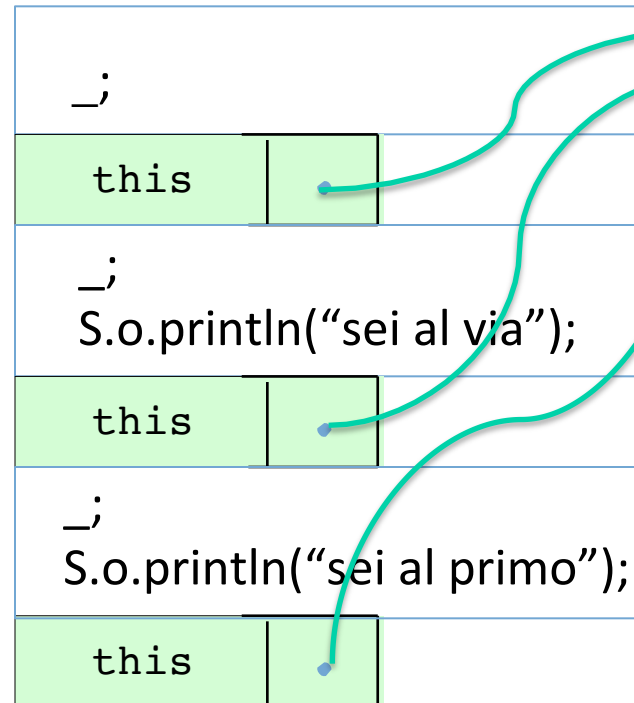


Abstract Stack Machine

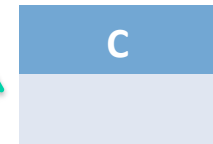
Workspace

```
throw new Exception();  
S.o.println("sei al secondo");
```

Stack



Heap

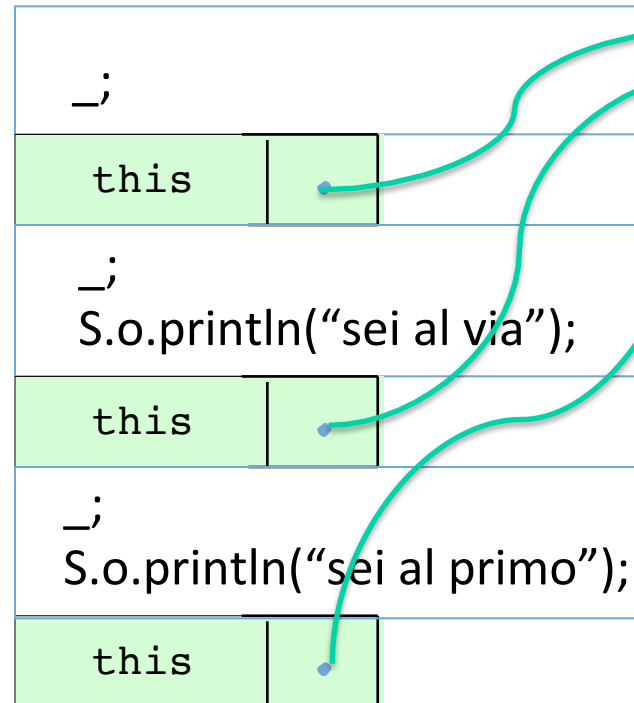


Abstract Stack Machine

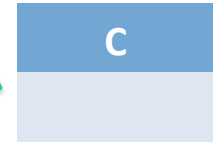
Workspace

```
throw new Exception();  
S.o.println("sei al secondo");
```

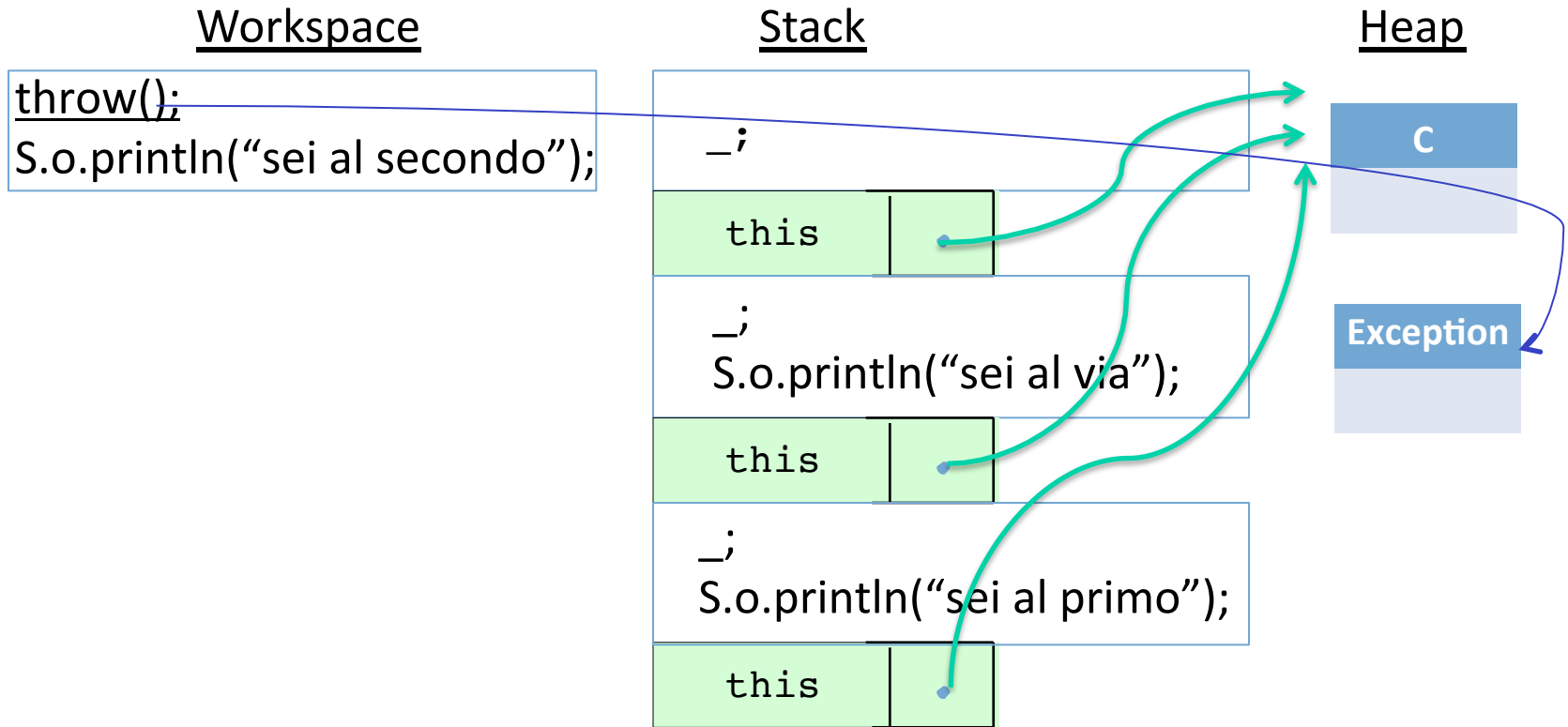
Stack



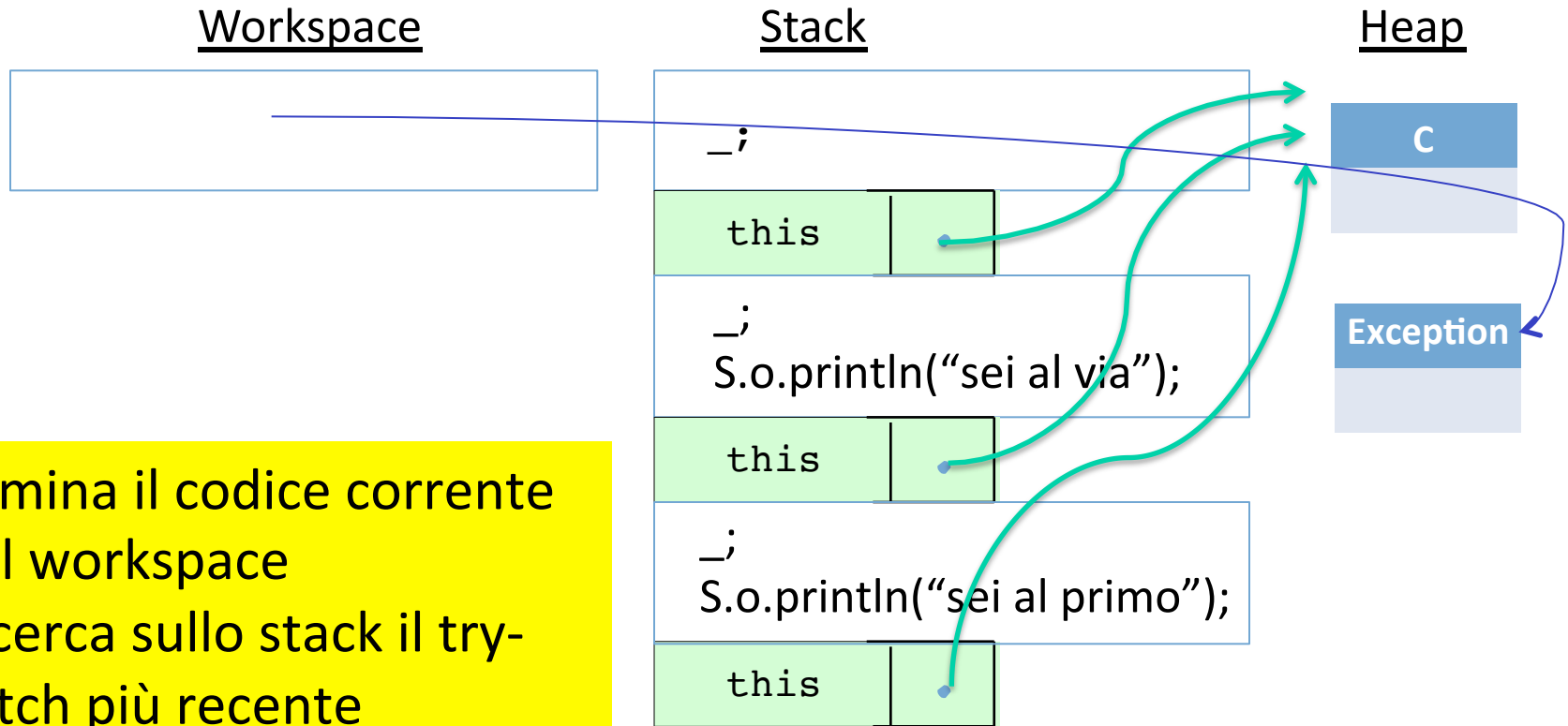
Heap



Abstract Stack Machine

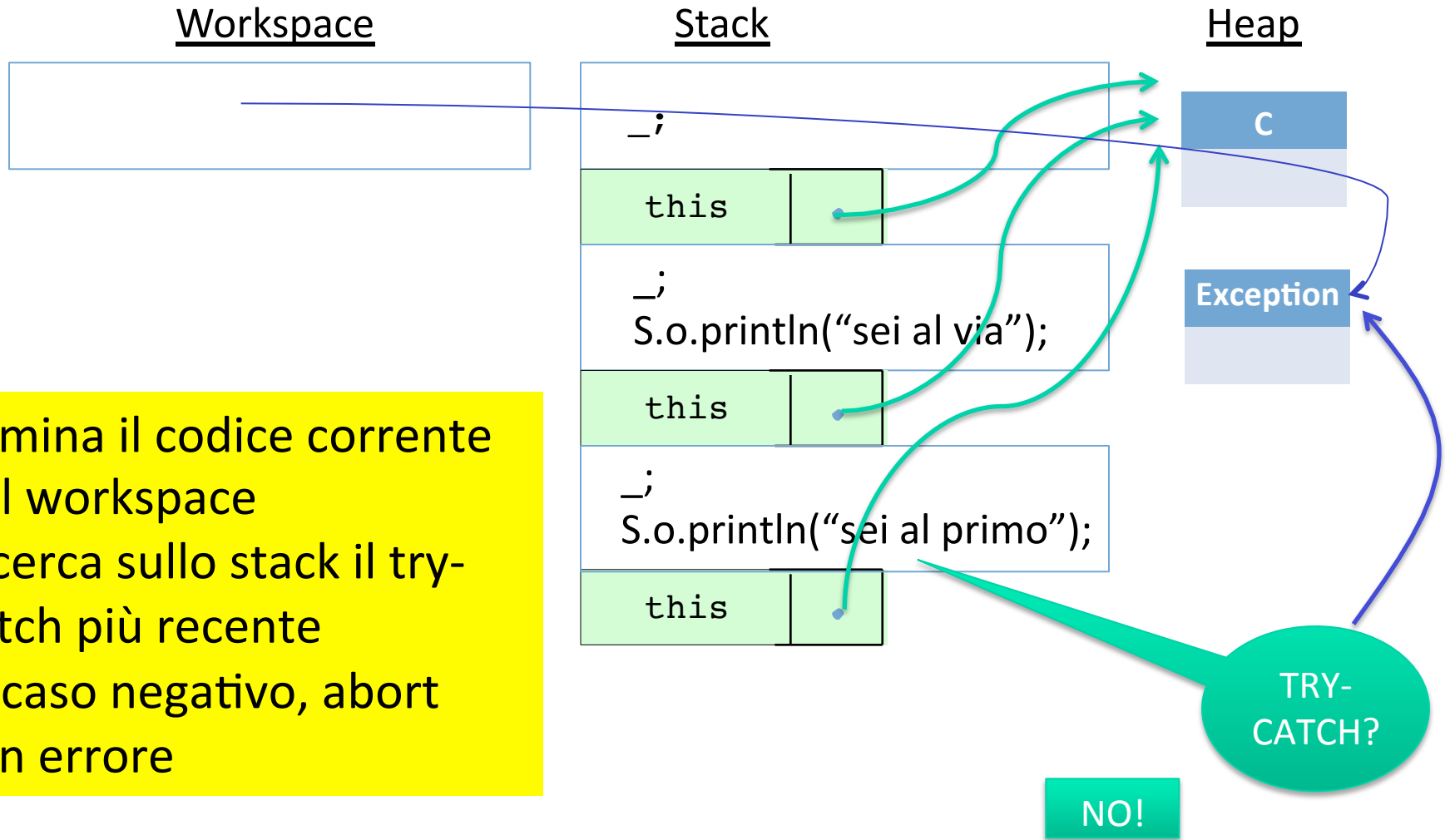


Abstract Stack Machine



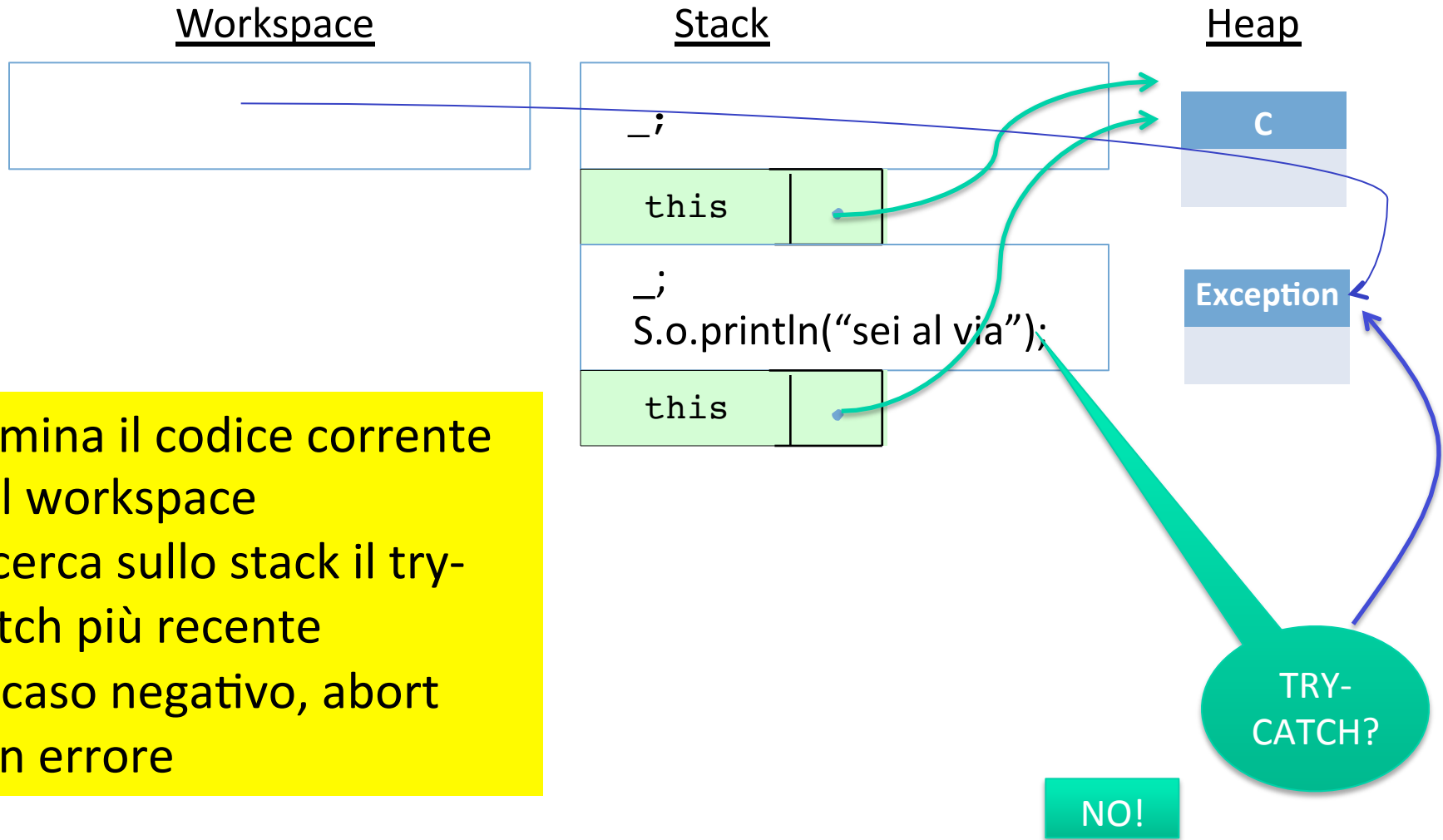
Elimina il codice corrente dal workspace
 Ricerca sullo stack il try-catch più recente
 In caso negativo, abort con errore

Abstract Stack Machine



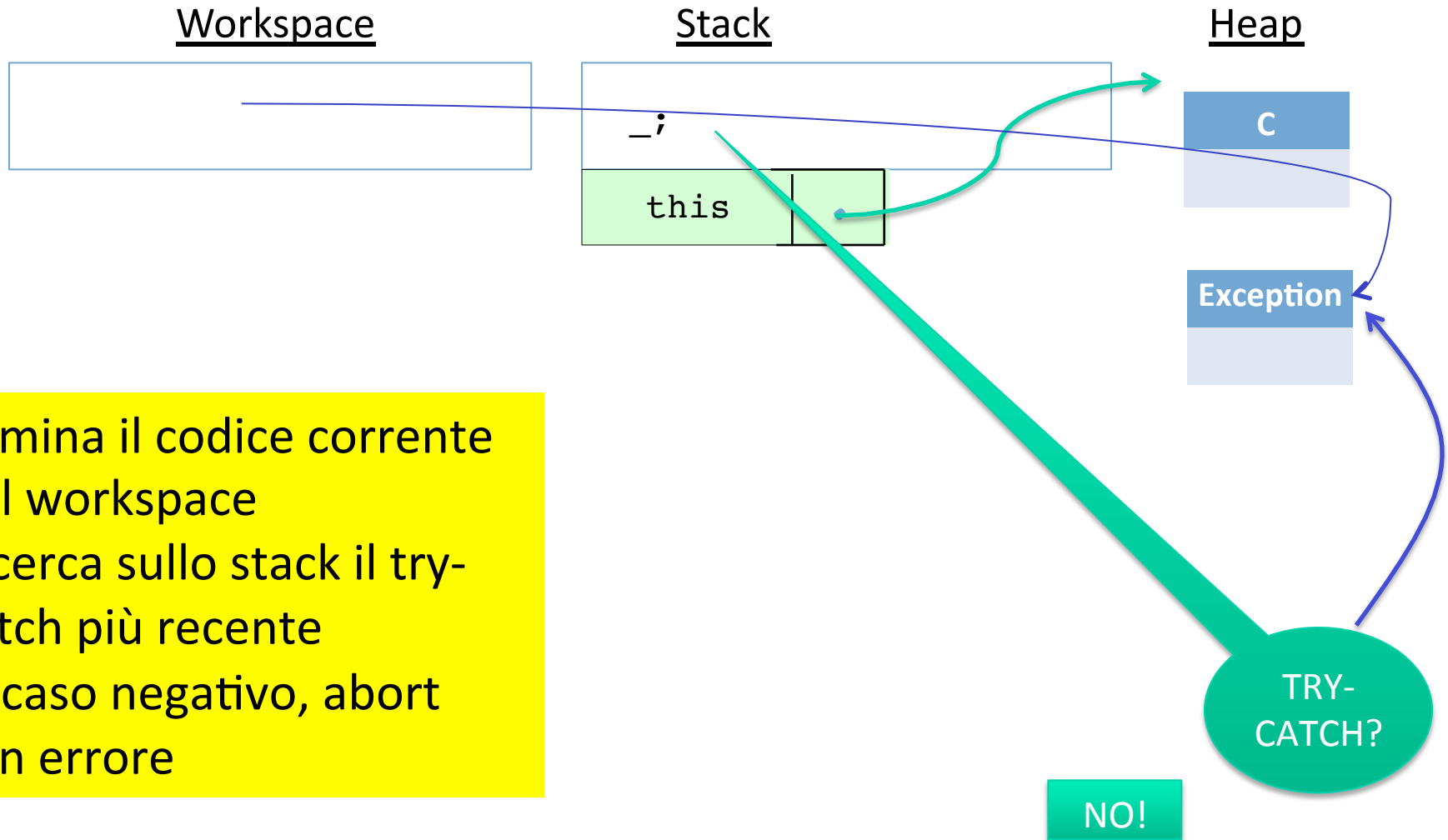
Elimina il codice corrente dal workspace
 Ricerca sullo stack il try-catch più recente
 In caso negativo, abort con errore

Abstract Stack Machine



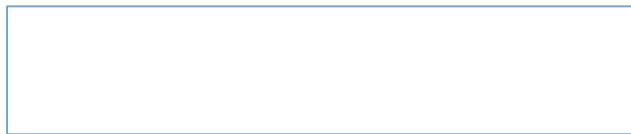
Elimina il codice corrente dal workspace
 Ricerca sullo stack il try-catch più recente
 In caso negativo, abort con errore

Abstract Stack Machine

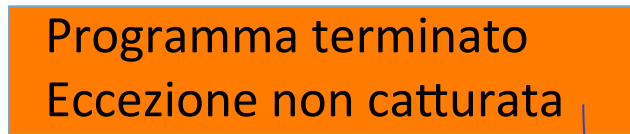


Abstract Stack Machine

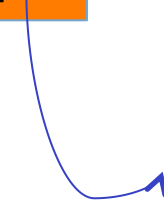
Workspace



Stack



Heap



```
class C {
    public void via( ) {
        primo( );
        System.out.println("sei al via");
    }

    public void primo( ) {
        try { secondo( ); }
        catch (Exception e) {
            System.out.println("catturata " + e)
        }
        System.out.println("sei al primo");
    }

    public void secondo( ) {
        throw new Exception( );
        System.out.println("sei al secondo");
    }
}
```

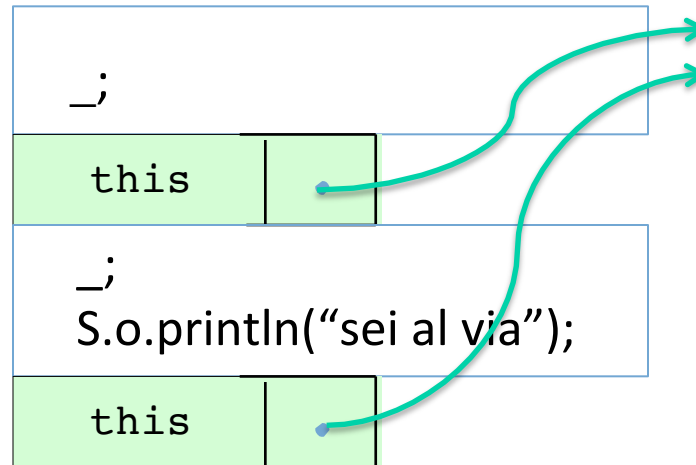
Cosa succede con (new C()).via();?

Abstract Stack Machine

Workspace

```
try { secondo( ); }  
catch( ... ) { ... }  
S.o.println("sei al primo");
```

Stack



Heap

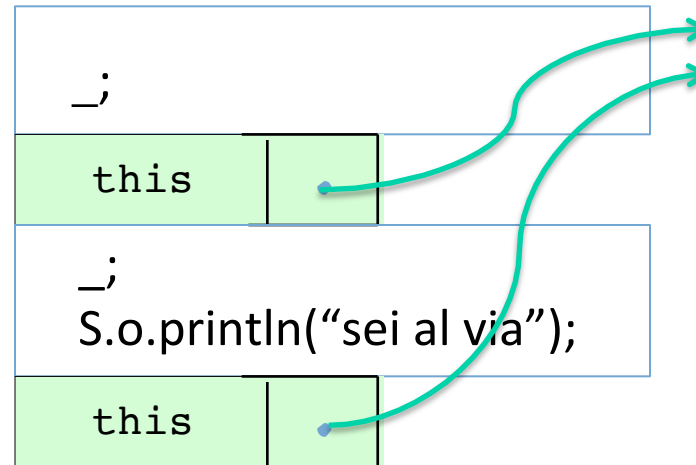


Abstract Stack Machine

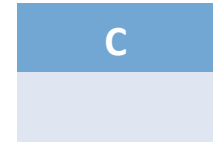
Workspace

```
try { secondo( ); }  
catch( ... ) { ... }  
S.o.println("sei al primo");
```

Stack



Heap

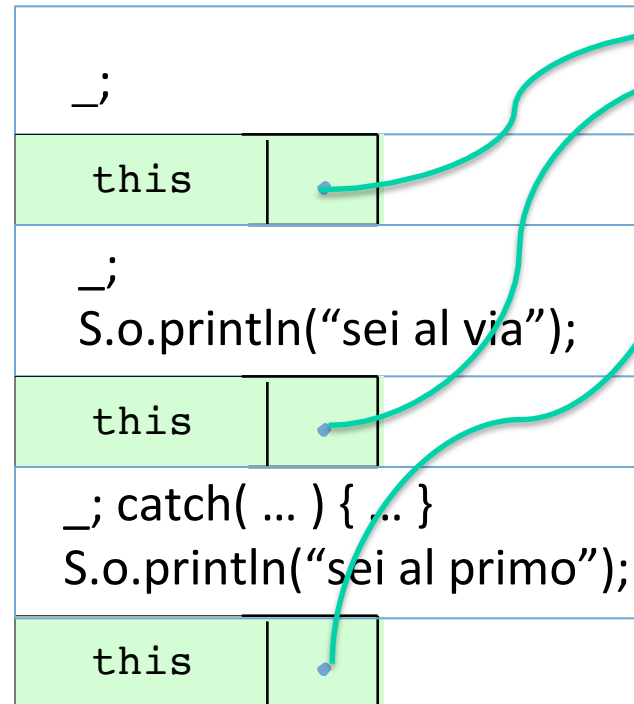


Abstract Stack Machine

Workspace

```
secondo( );
```

Stack



Heap



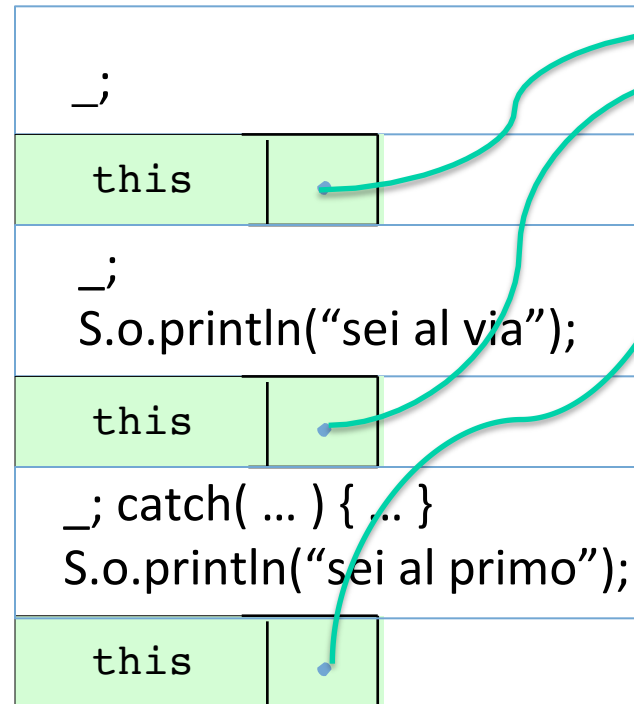
L'esecuzione del try-catch inserisce sullo stack il codice di gestione e lascia sul workspace il codice monitorato

Abstract Stack Machine

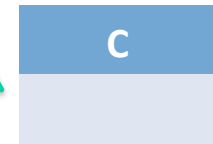
Workspace

```
secondo( );
```

Stack



Heap



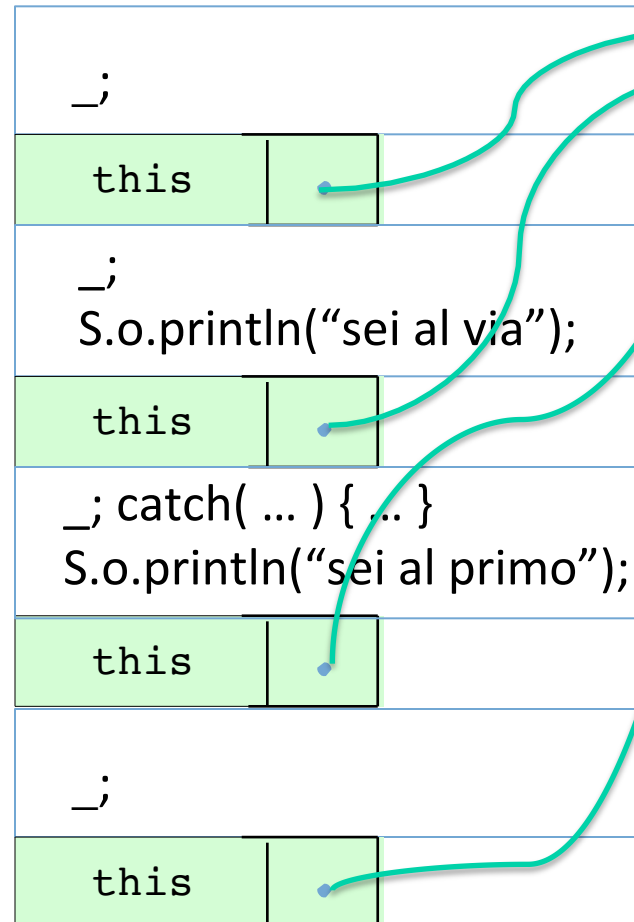
L'esecuzione del try-catch inserisce sullo stack il codice di gestione e lascia sul workspace il codice monitorato

Abstract Stack Machine

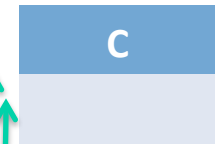
Workspace

```
throw new Exception();
S.o.println("sei al secondo");
```

Stack



Heap

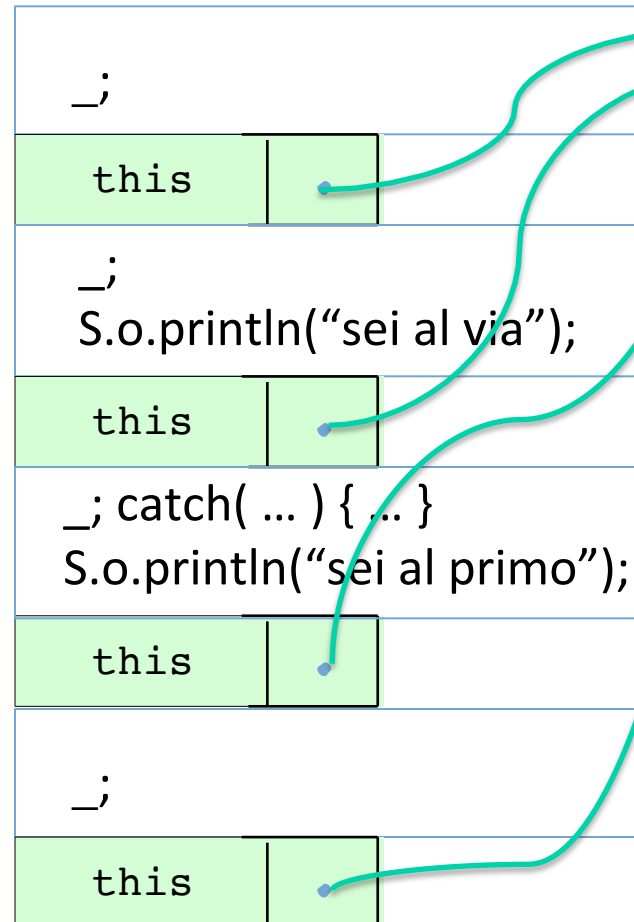


Abstract Stack Machine

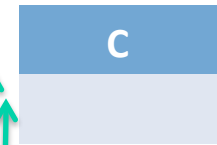
Workspace

```
throw new Exception();
S.o.println("sei al secondo");
```

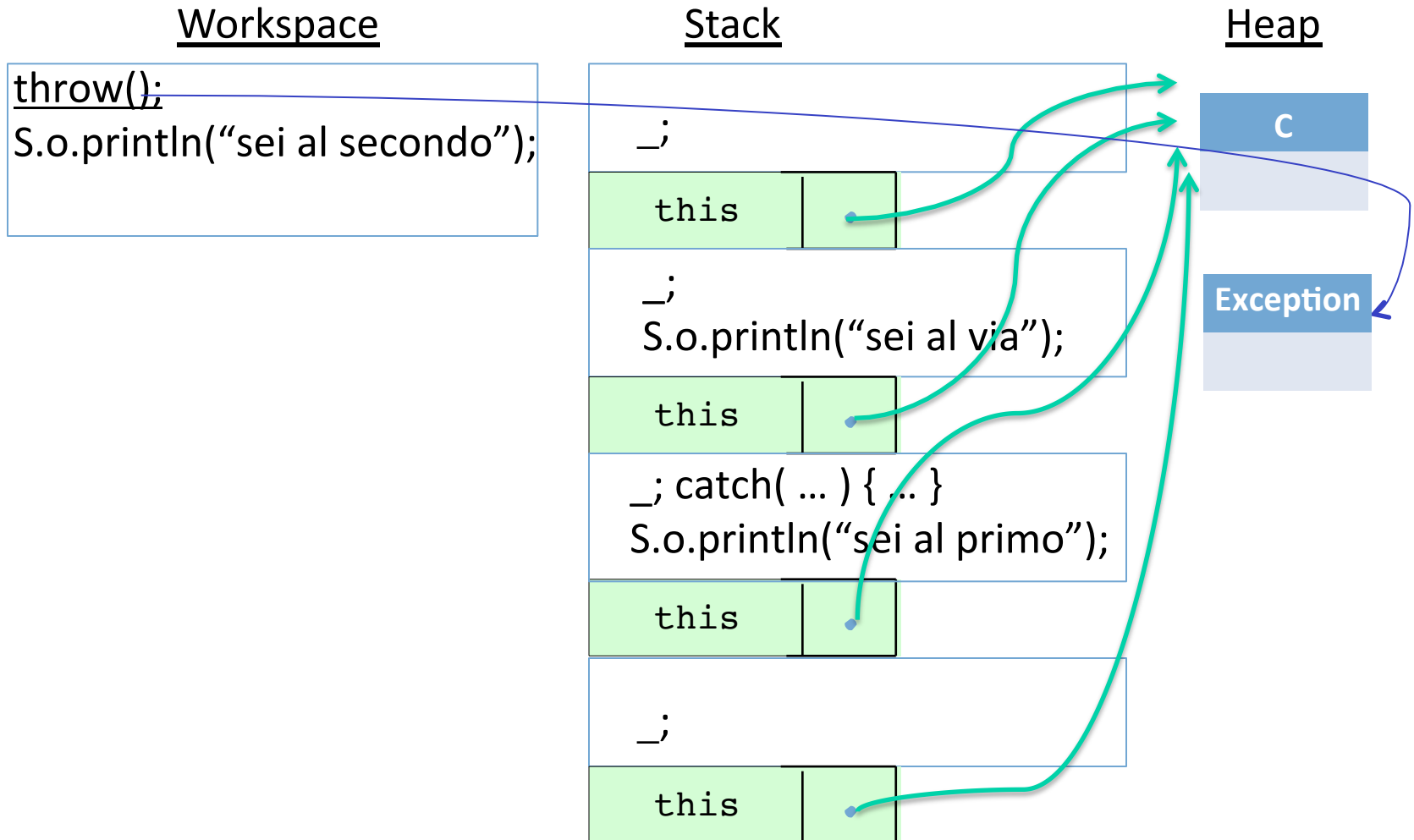
Stack



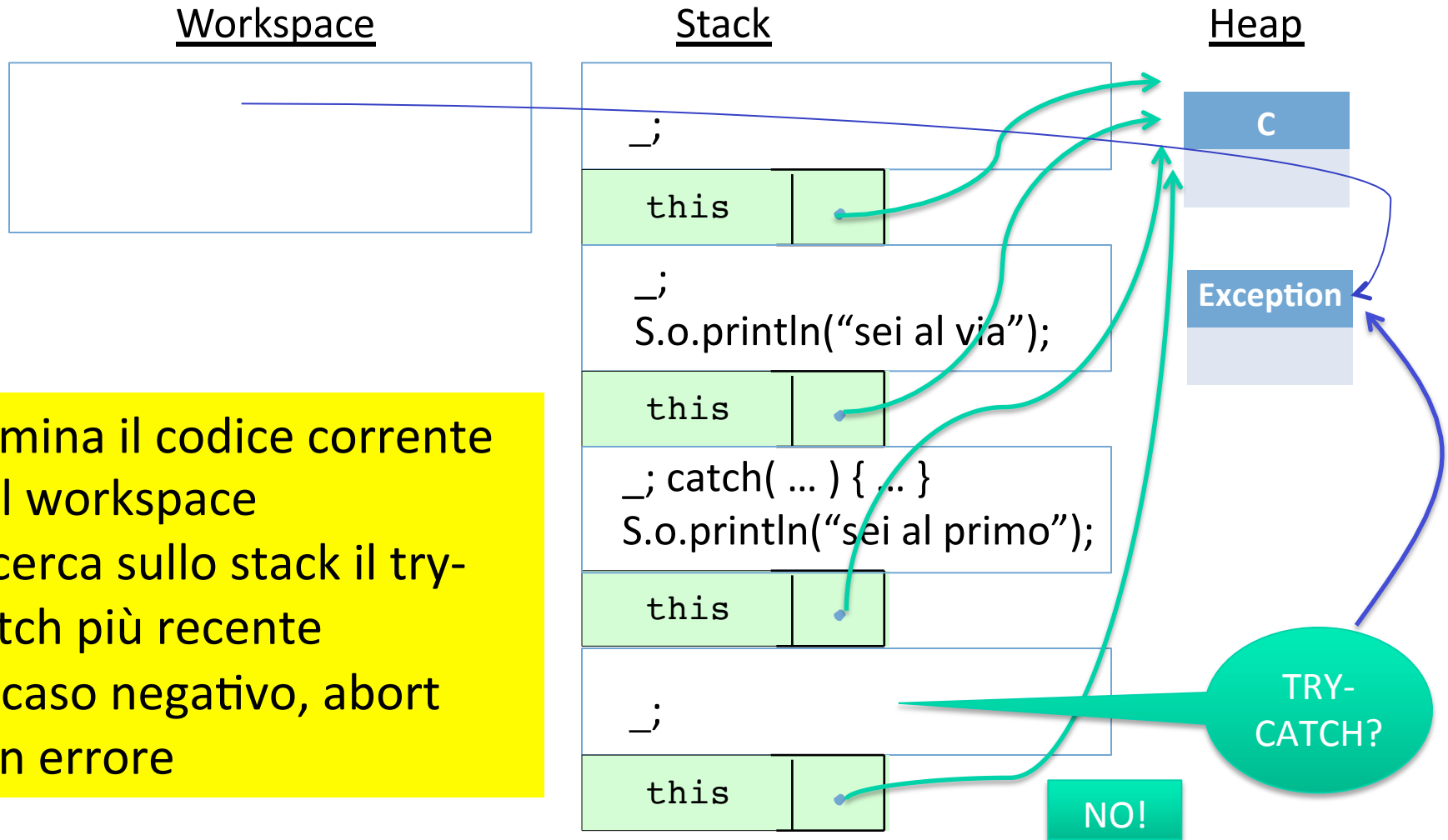
Heap



Abstract Stack Machine

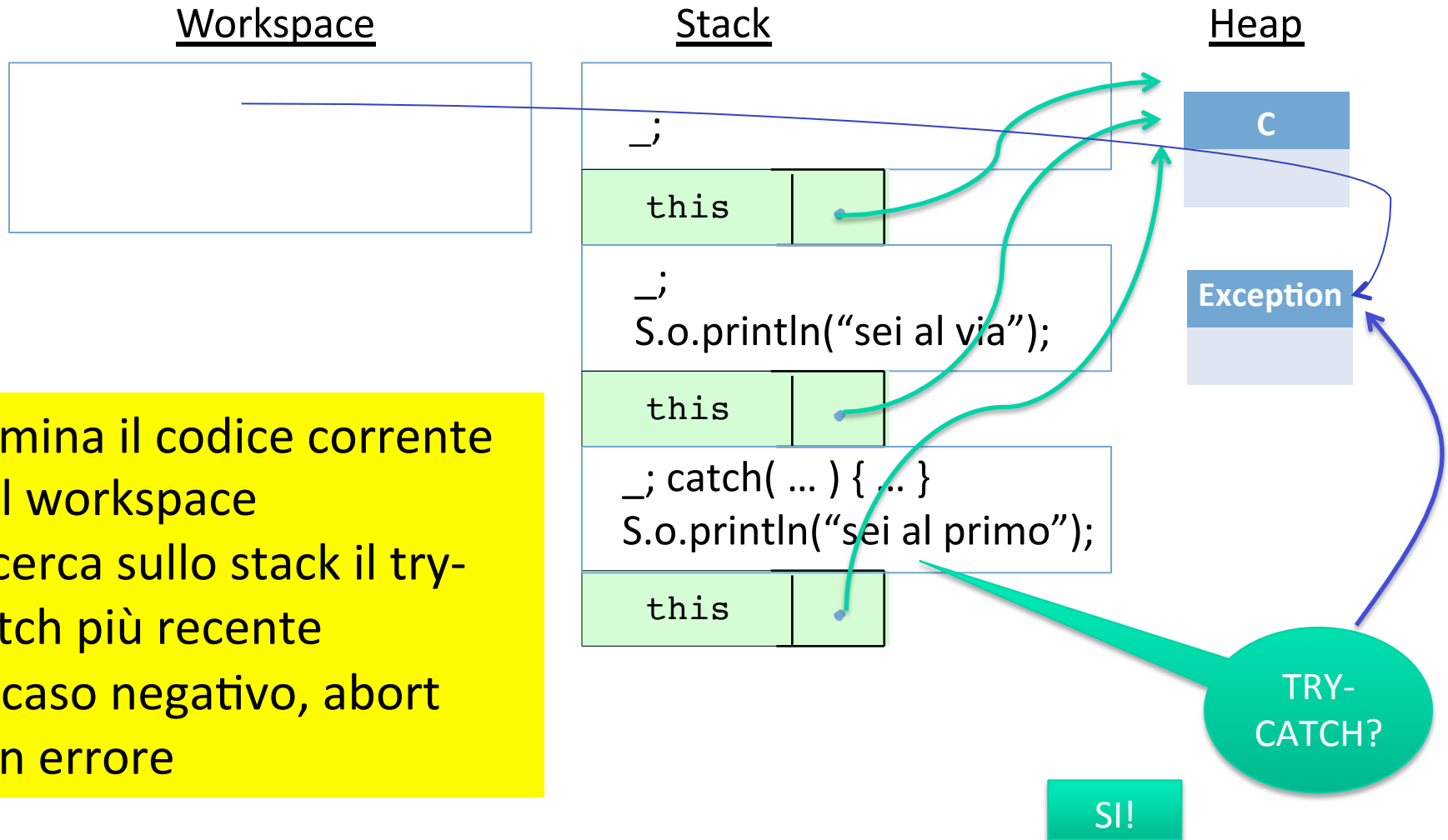


Abstract Stack Machine



Elimina il codice corrente dal workspace
 Ricerca sullo stack il try-catch più recente
 In caso negativo, abort con errore

Abstract Stack Machine



Elimina il codice corrente dal workspace
 Ricerca sullo stack il try-catch più recente
 In caso negativo, abort con errore

Abstract Stack Machine

