The "this" reference

- The reference "this" is a reserved word
- It can occur inside a normal (non-static) method
- It has a reference to the object receiving the message

Example

```
public class Movie
{
  String title, director;
  Movie(String t, String d)
  {
    title = t;
    director = d;
  }
  void print()
  {
    System.out.println(title);
    System.out.println(director);
  }
}
```

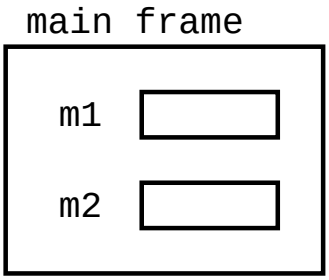
Is the same as...

```
public class Movie
{
  String title, director;
  Movie(String t, String d)
  {
    this.title = t;
    this.director = d;
  }
  void print()
  {
    System.out.println(this.title);
    System.out.println(this.director);
  }
}
```

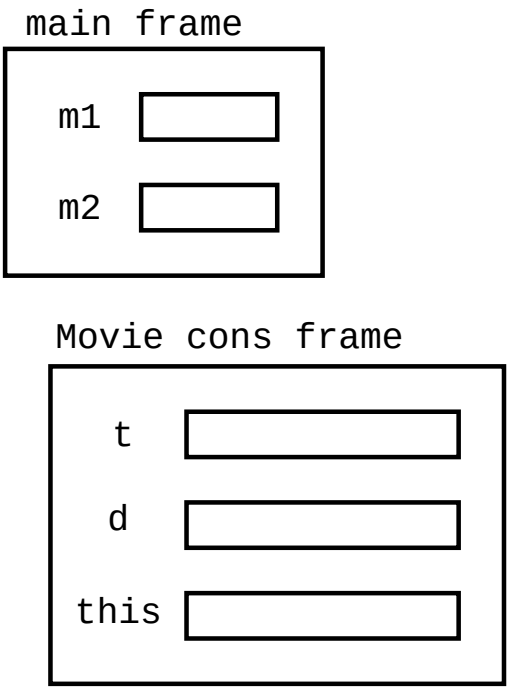
Example

```
public class MovieApplication {
  public static void main(String[] args)
  {
    Movie m1, m2;
    m1 = new Movie(“Trainspotting”, “Danny Boyle”)
    m2 = new Movie(“Pirates of the Caribbean”,
                   “Gore Verbinski”);
    m1.print();
    m2.print();
  }
}
```

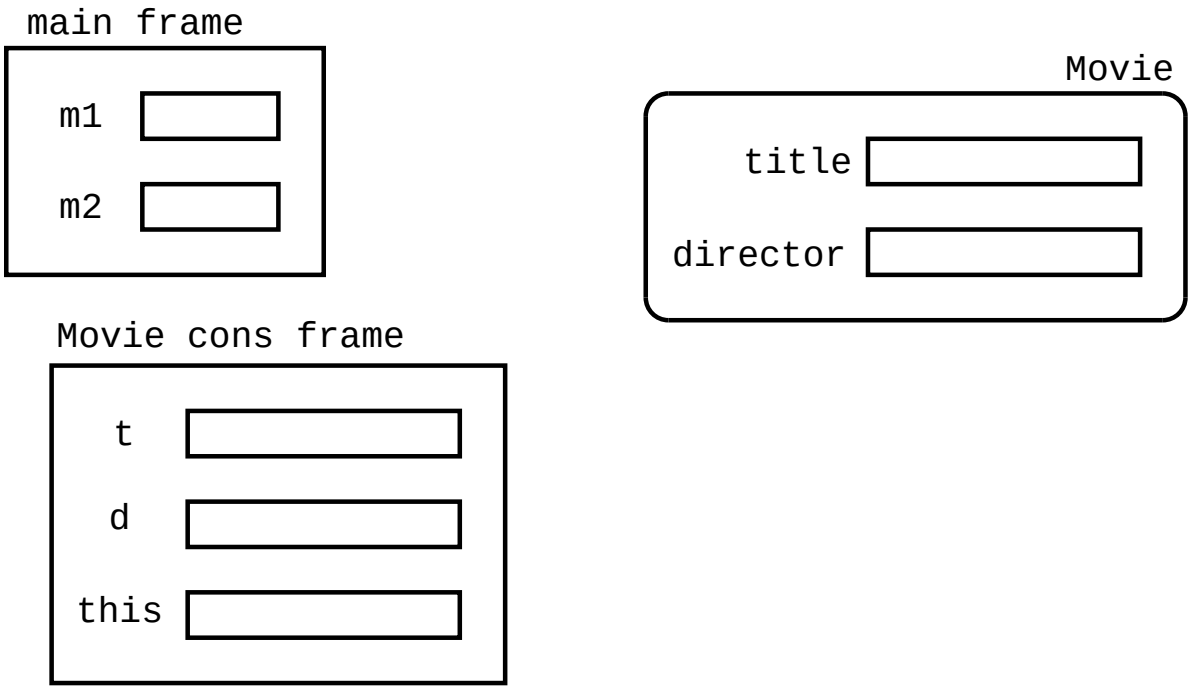
Execution



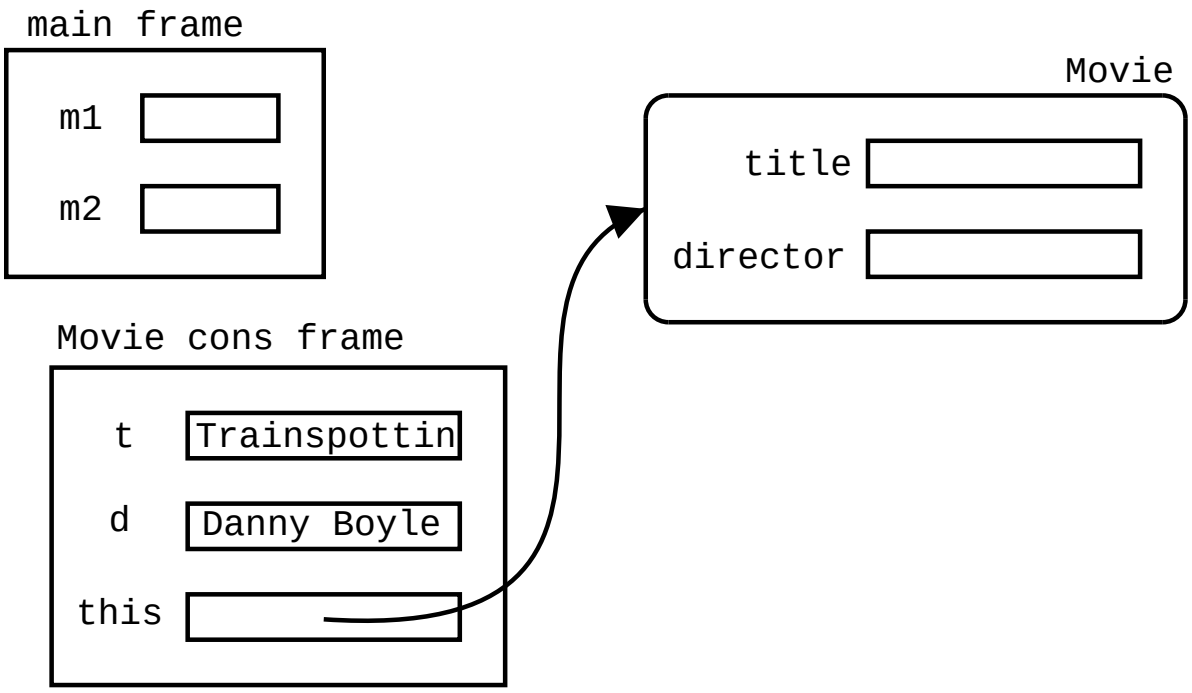
Execution



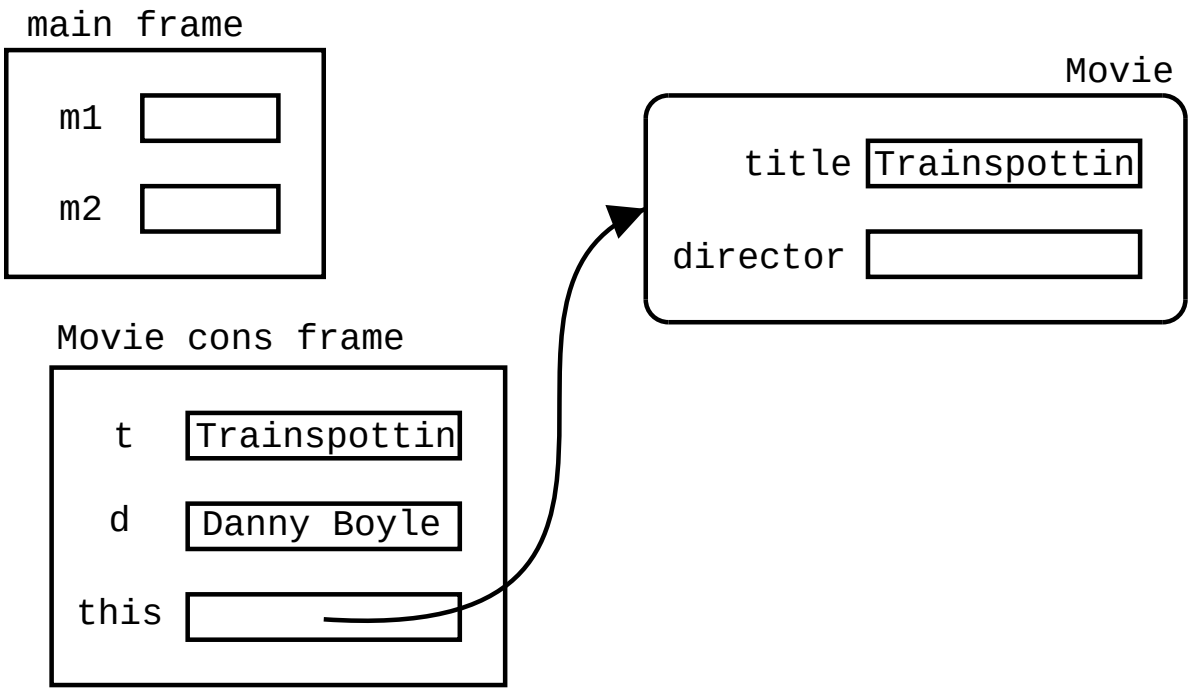
Execution


main frame
m1
m2
Movie
title
director
Movie cons frame
t
d
this

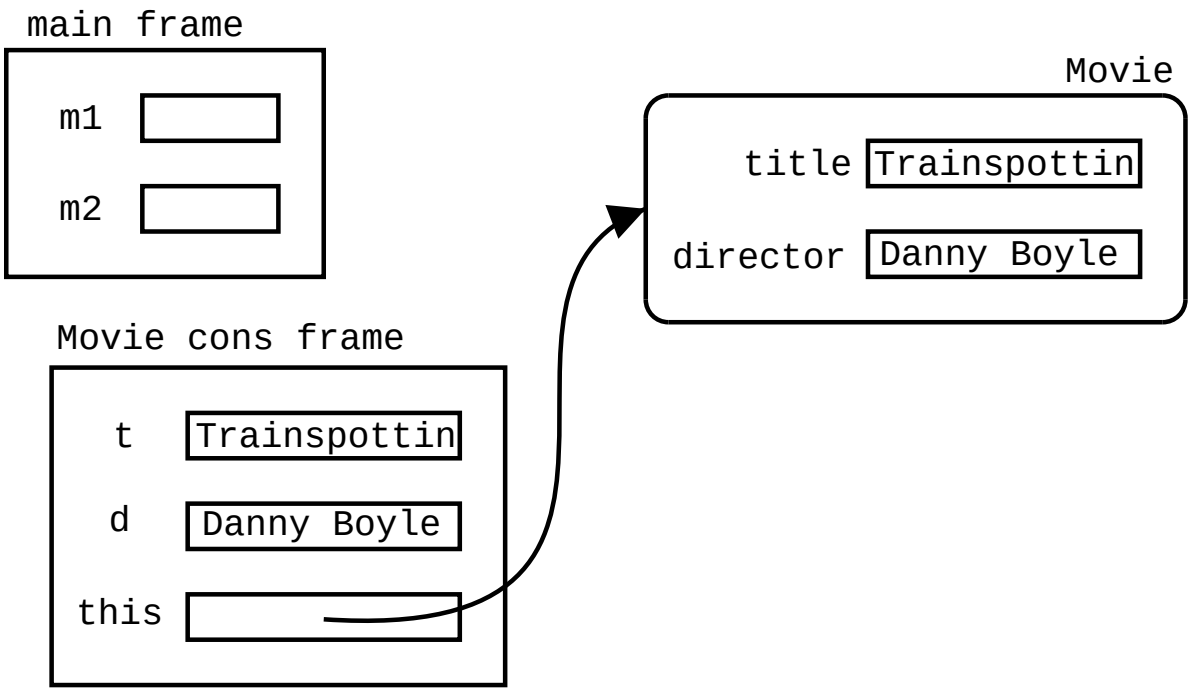
Execution


main frame
m1
m2
Movie
title
director
Movie cons frame
t
Trainspottin
d
Danny Boyle
this

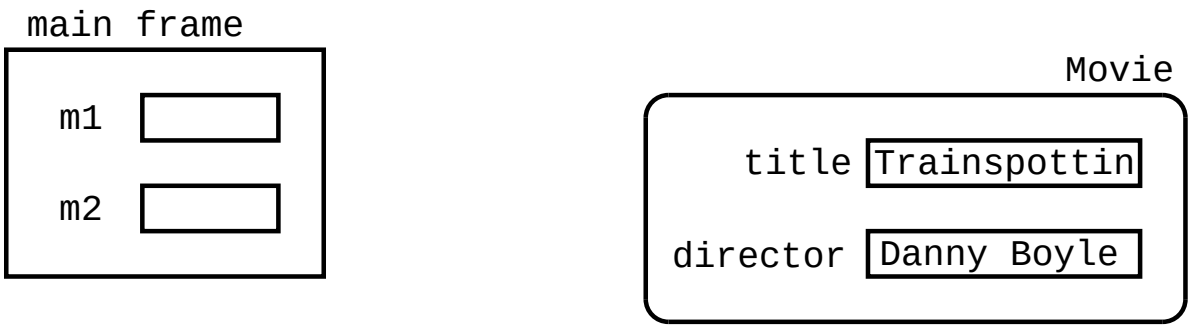
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Movie cons frame
t
Trainspottin
d
Danny Boyle
this

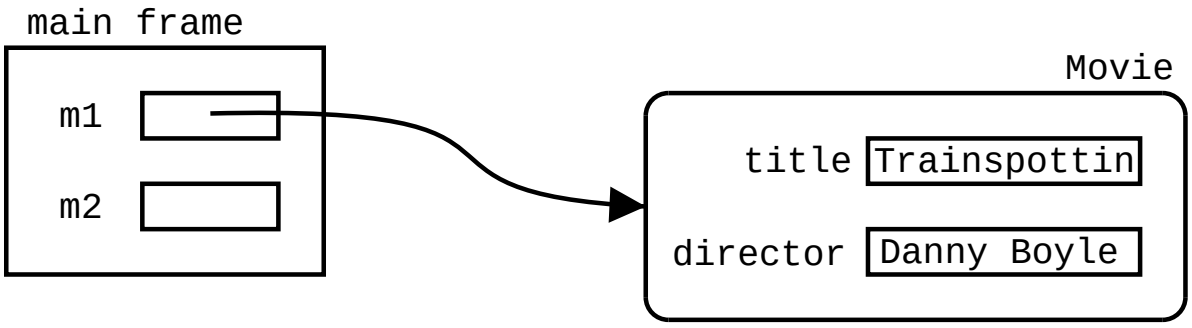
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
Trainspottin
d
Danny Boyle
this

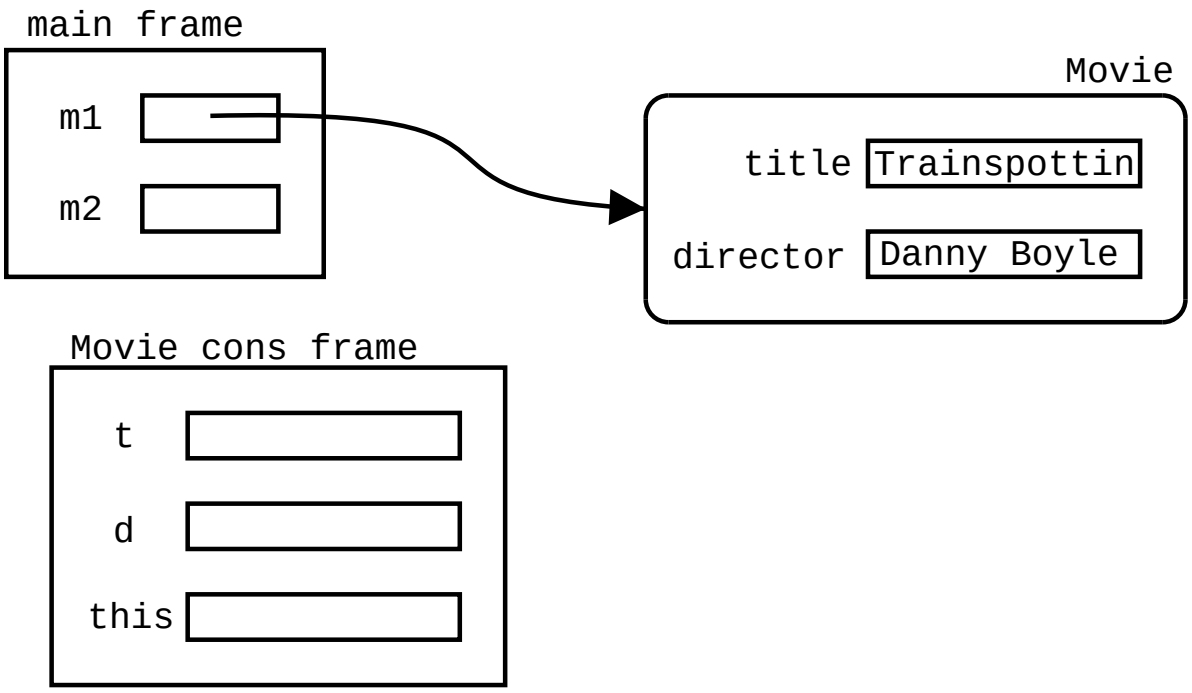
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle

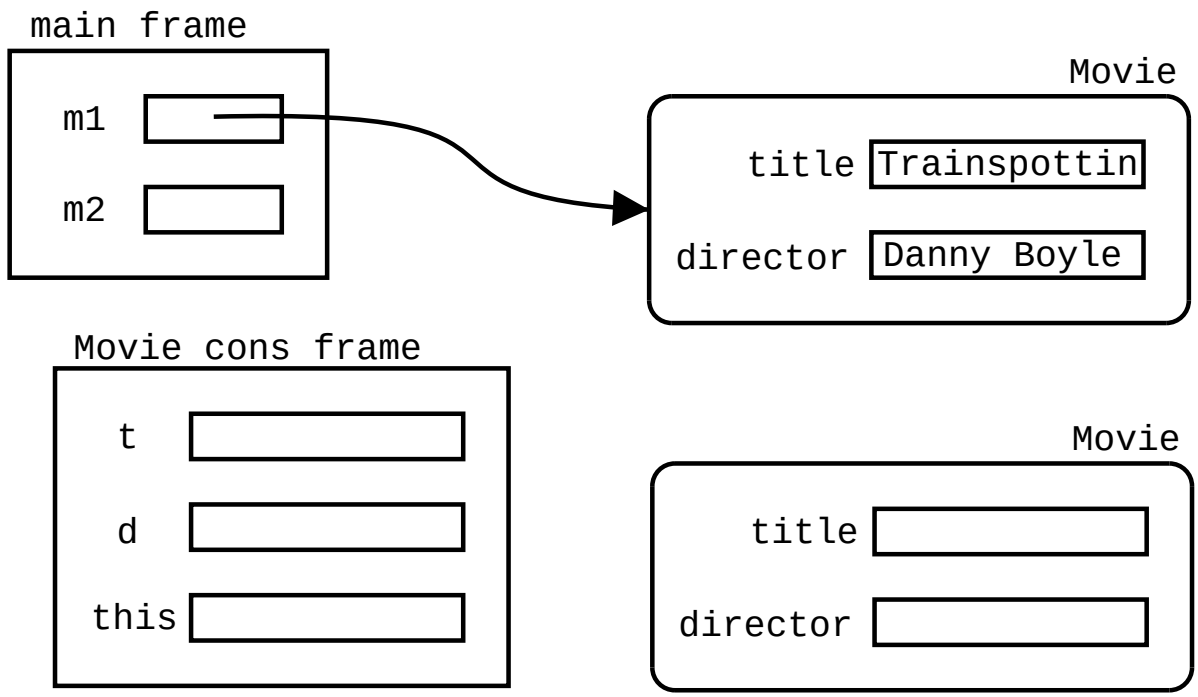
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle

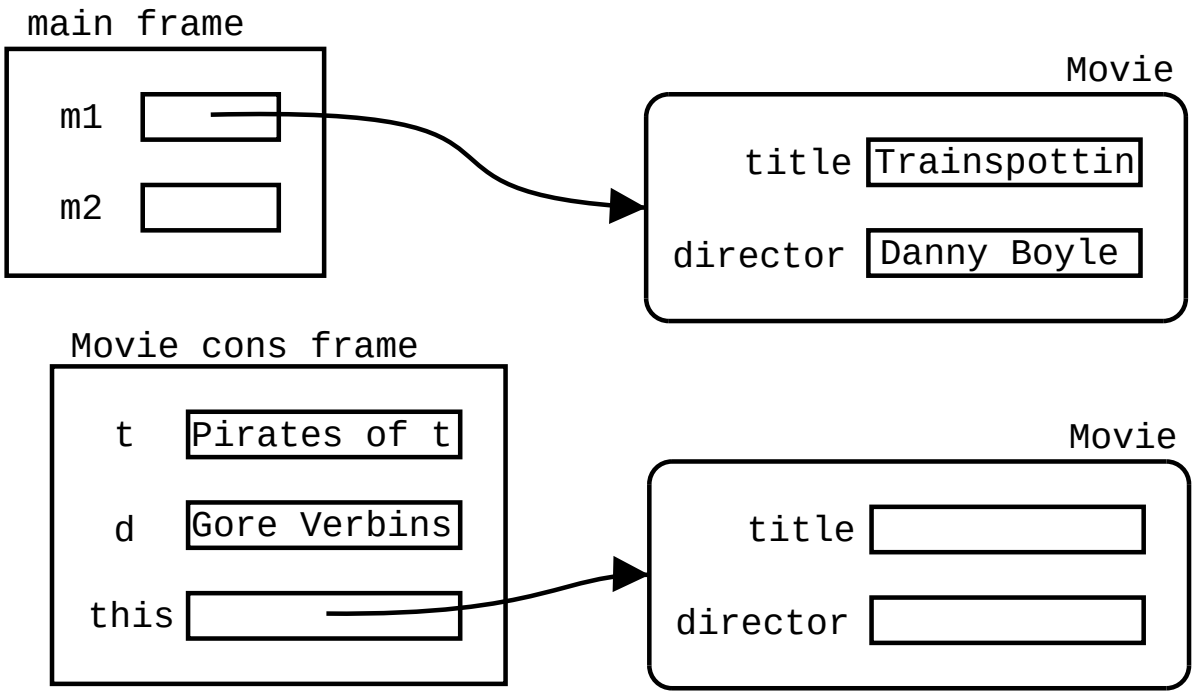
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
d
this

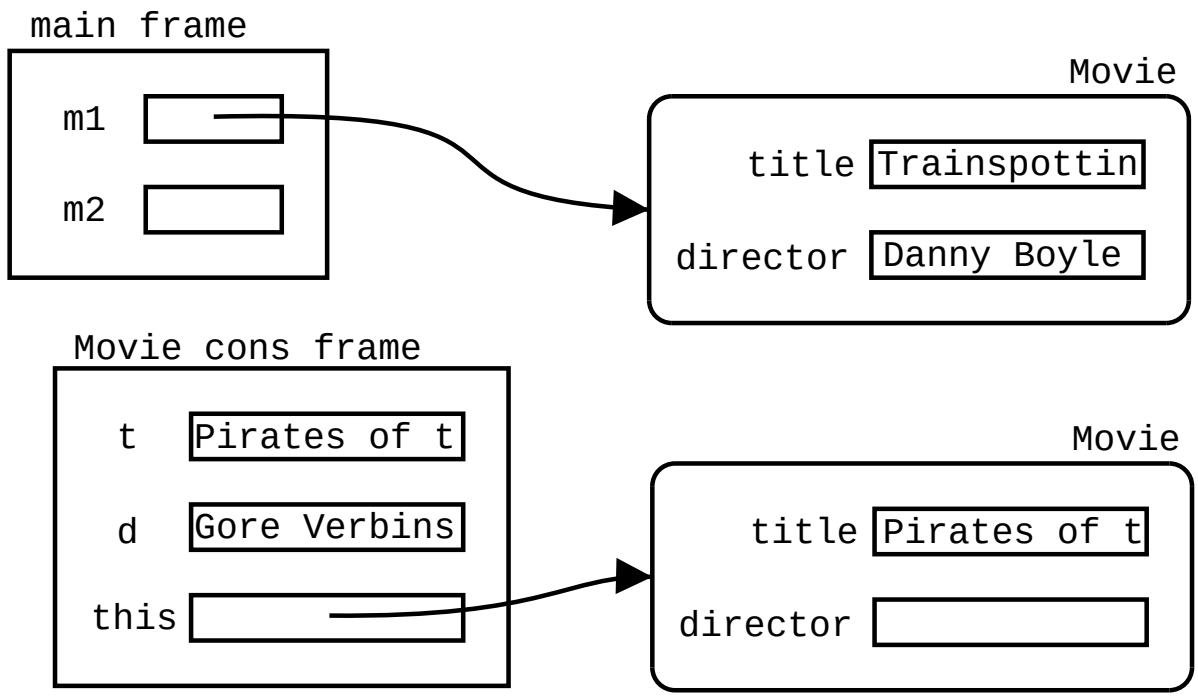
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
d
this
Movie
title
director

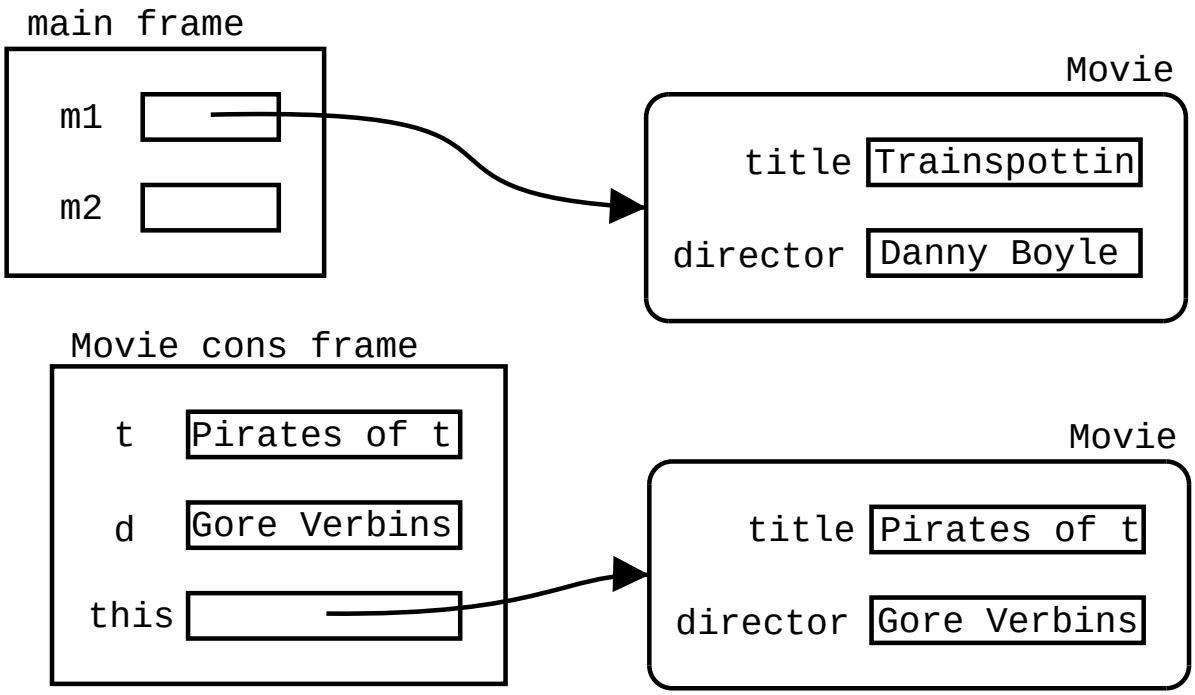
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
Pirates of t
d
Gore Verbins
this
Movie
title
director

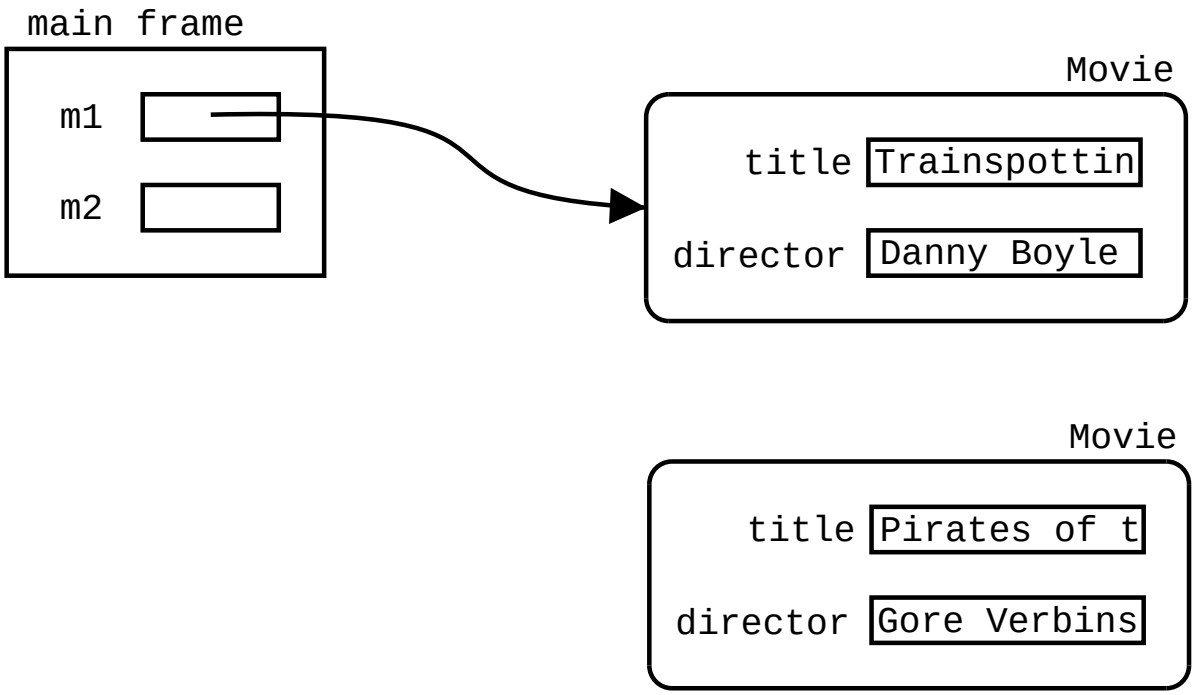
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
Pirates of t
d
Gore Verbins
this
Movie
title
Pirates of t
director

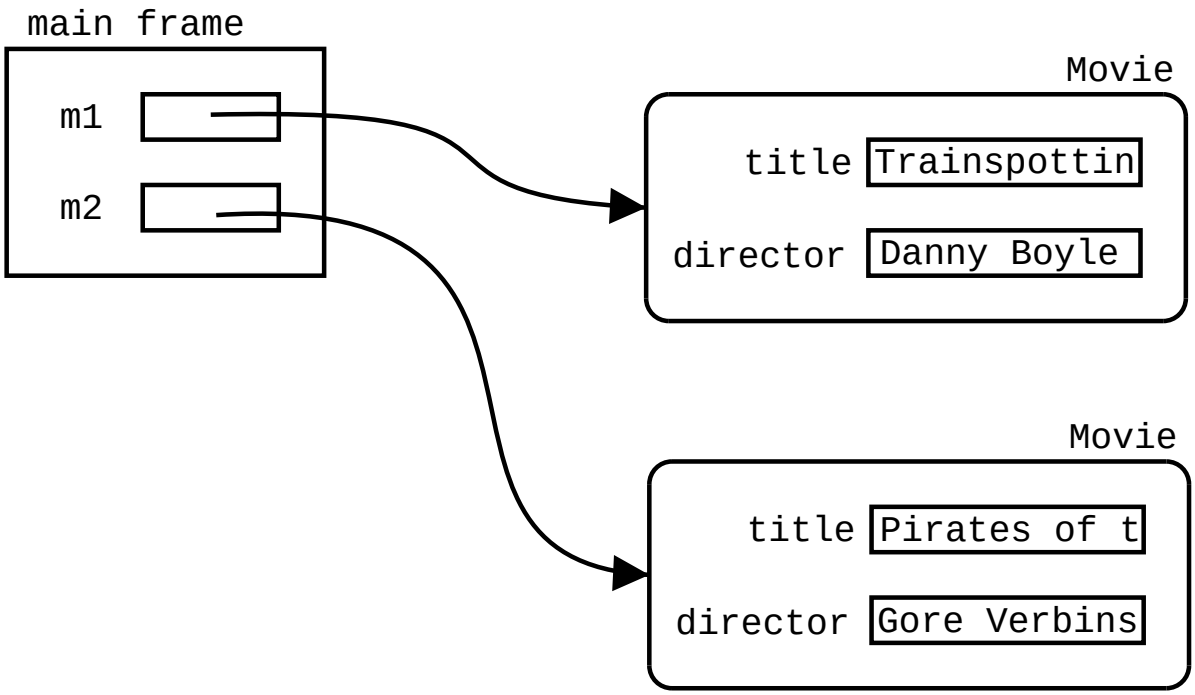
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie cons frame
t
Pirates of t
d
Gore Verbins
this
Movie
title
Pirates of t
director
Gore Verbins

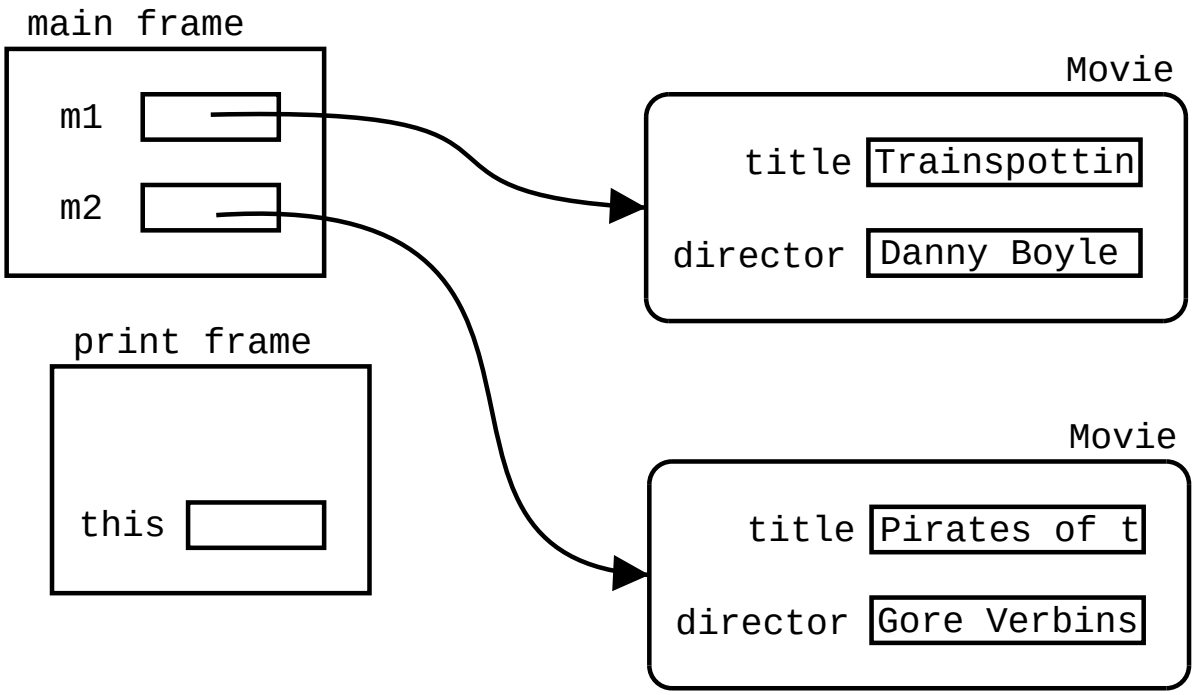
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

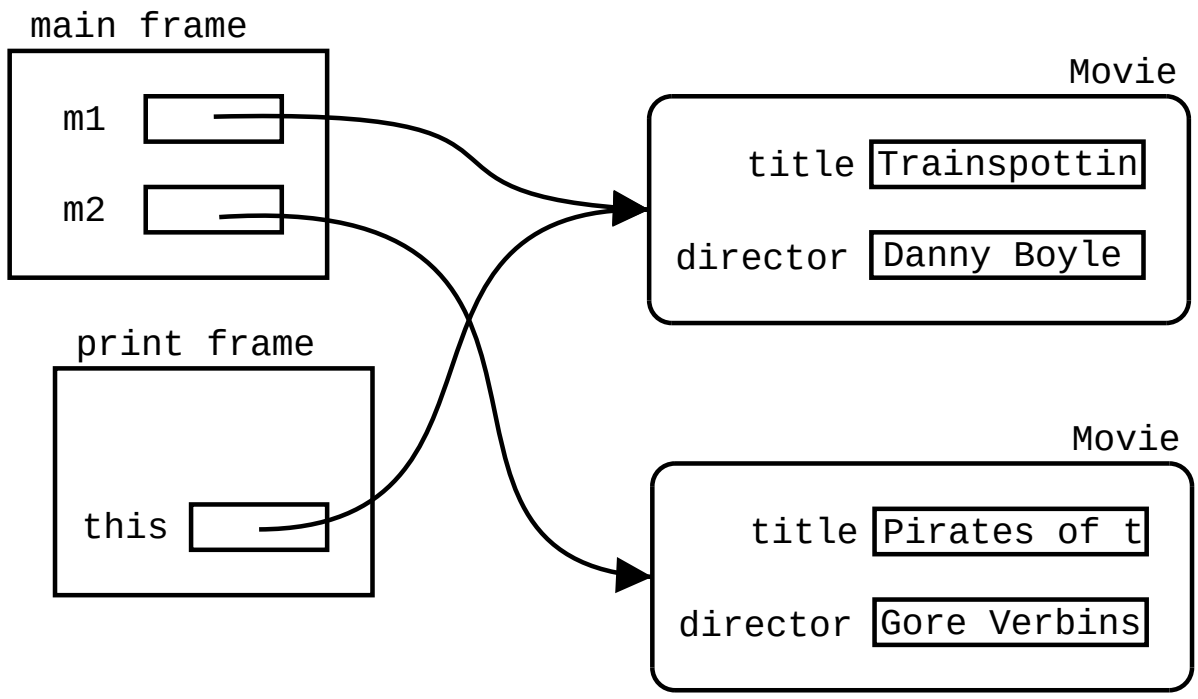
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

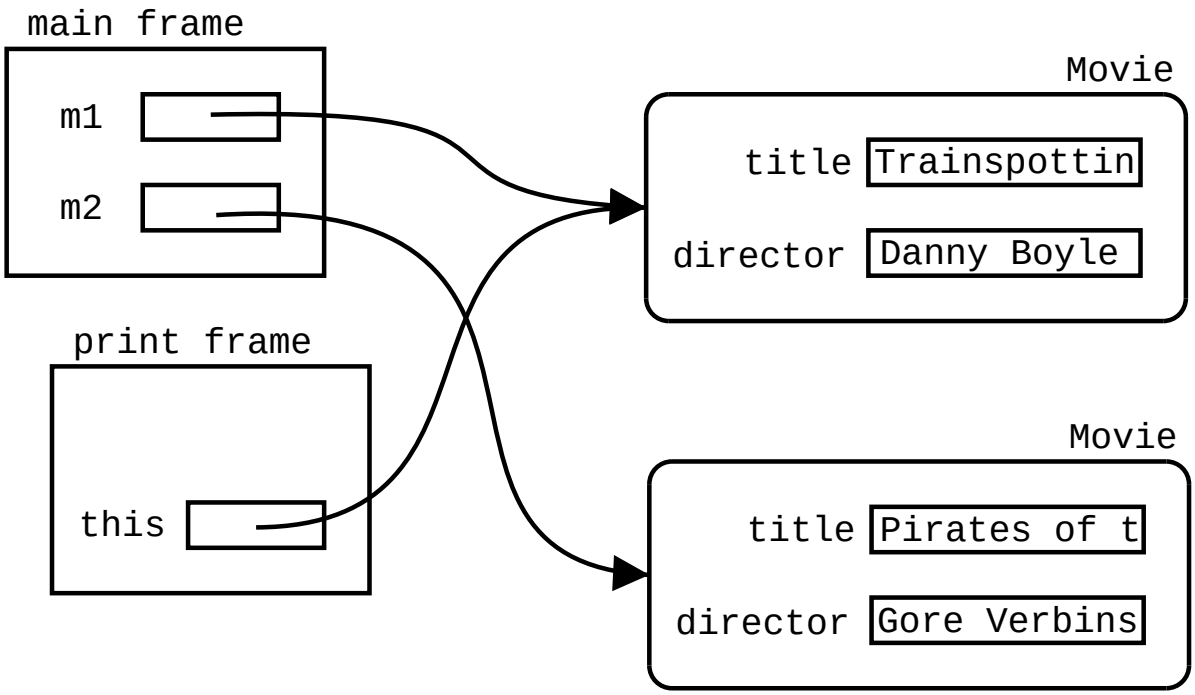
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

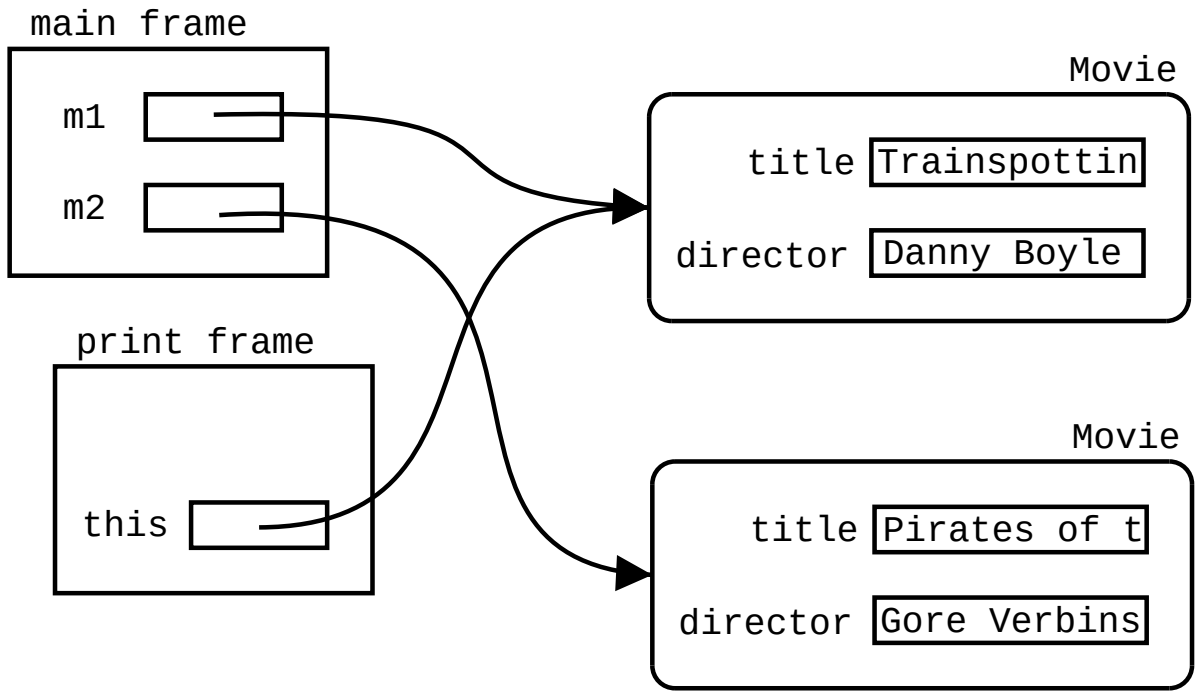
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

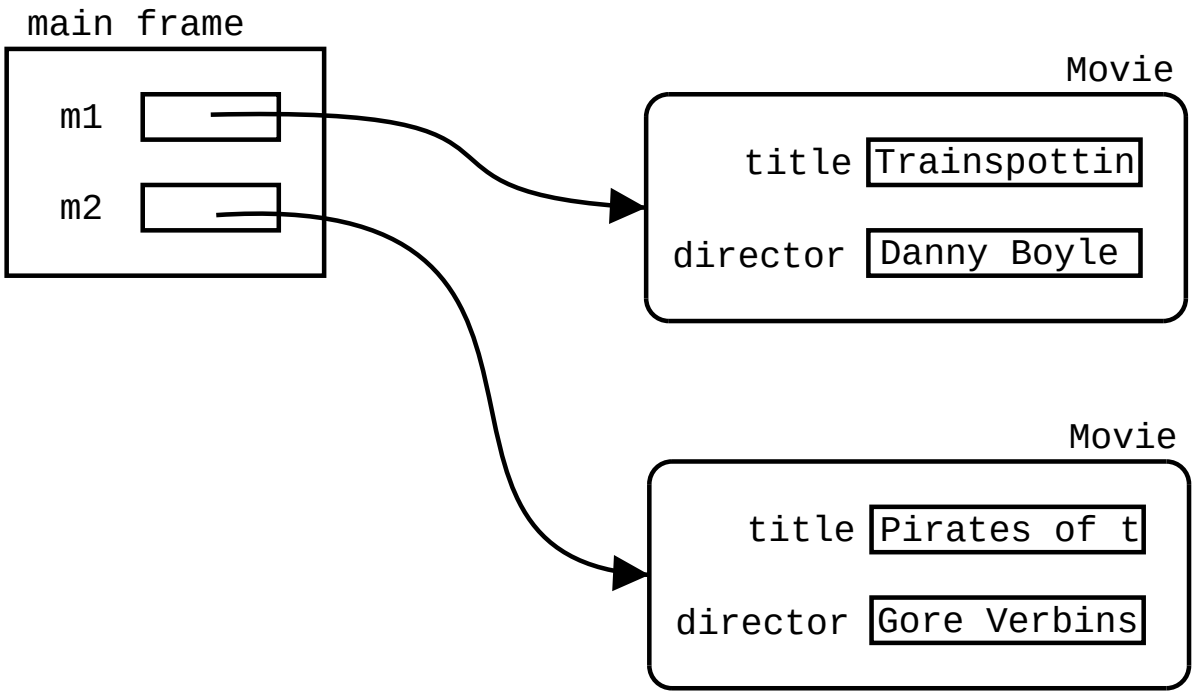
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

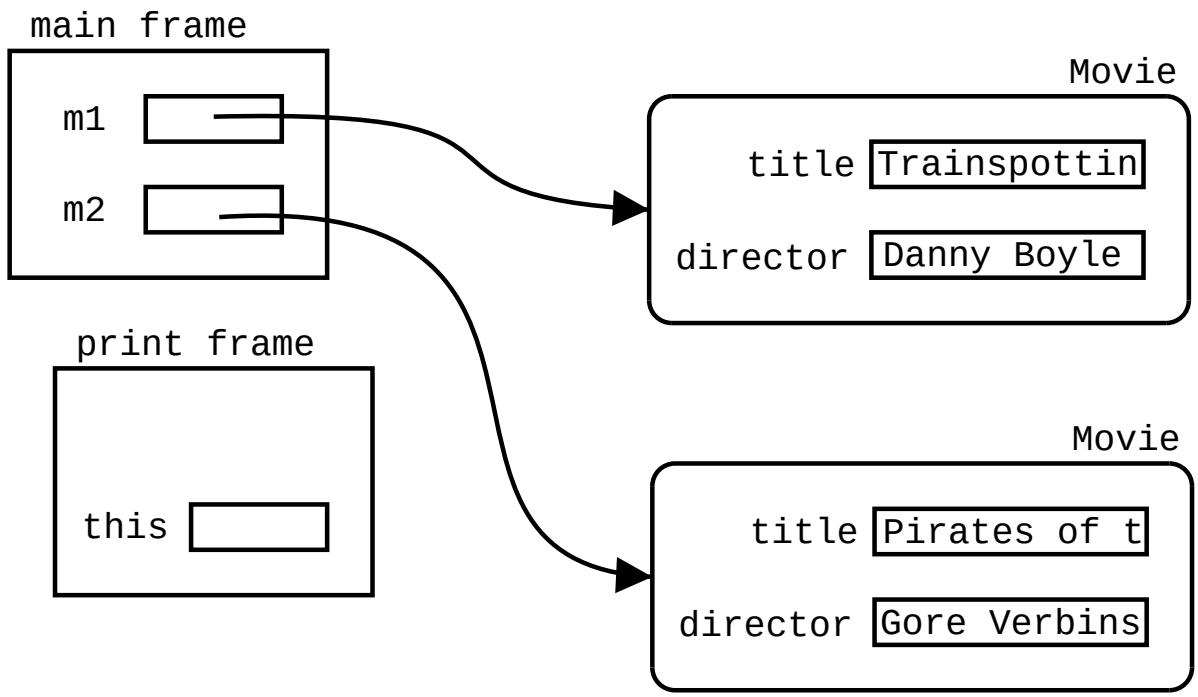
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

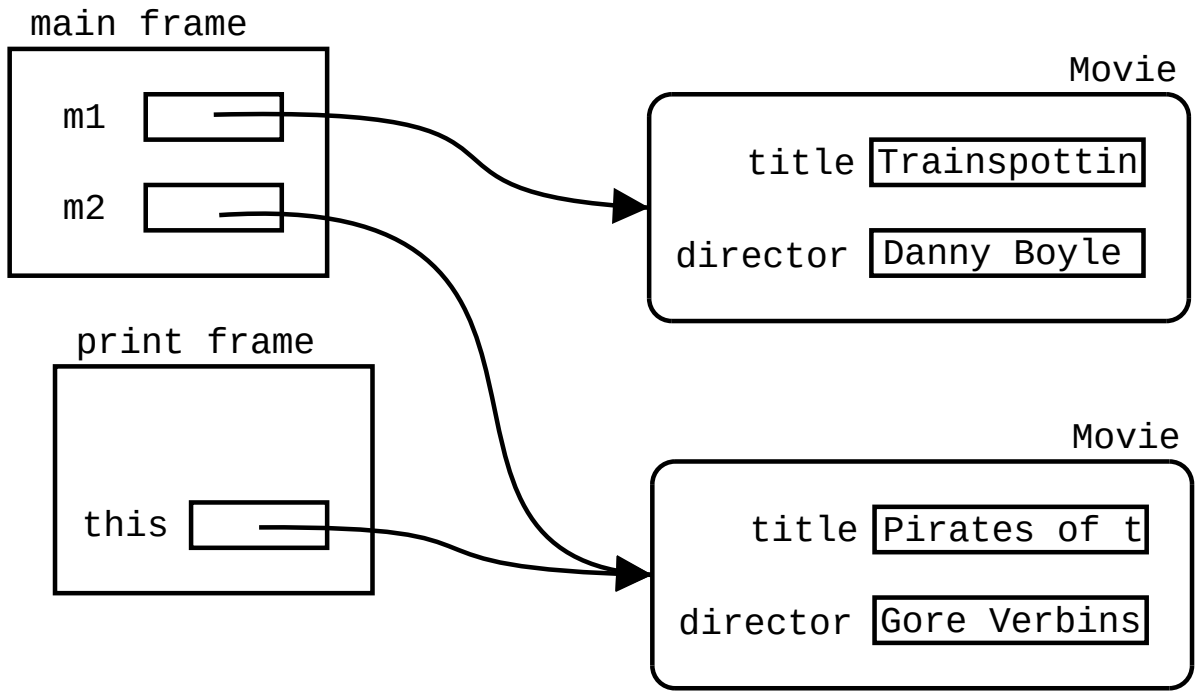
Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

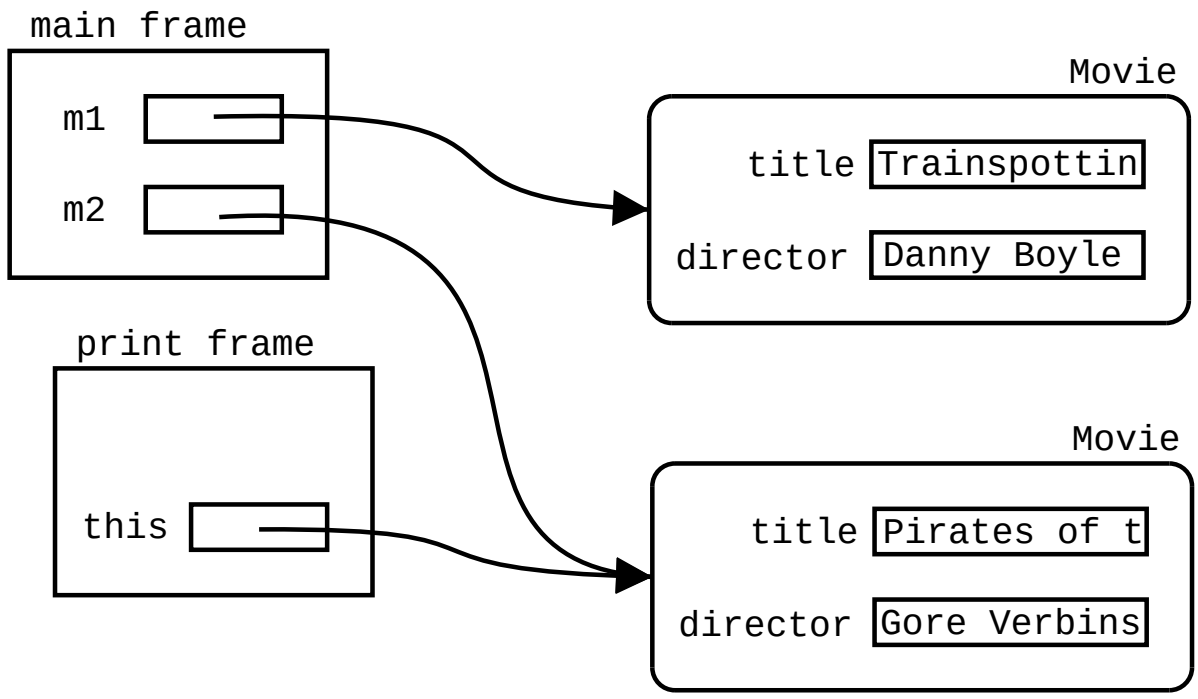
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

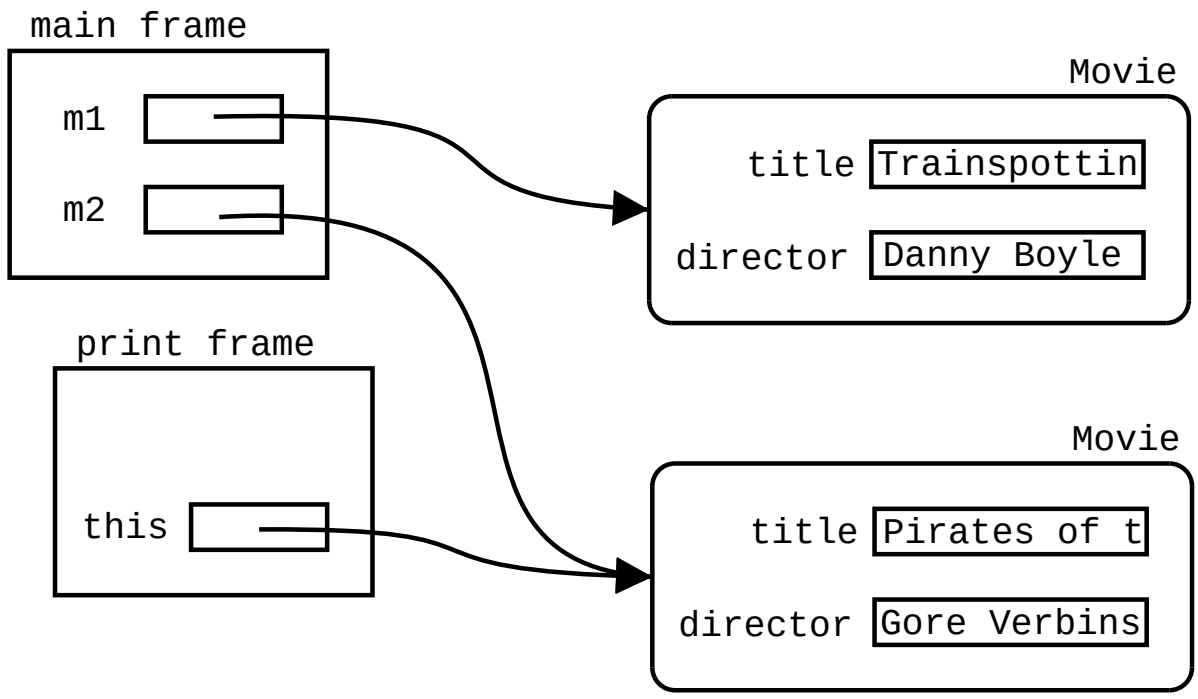
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

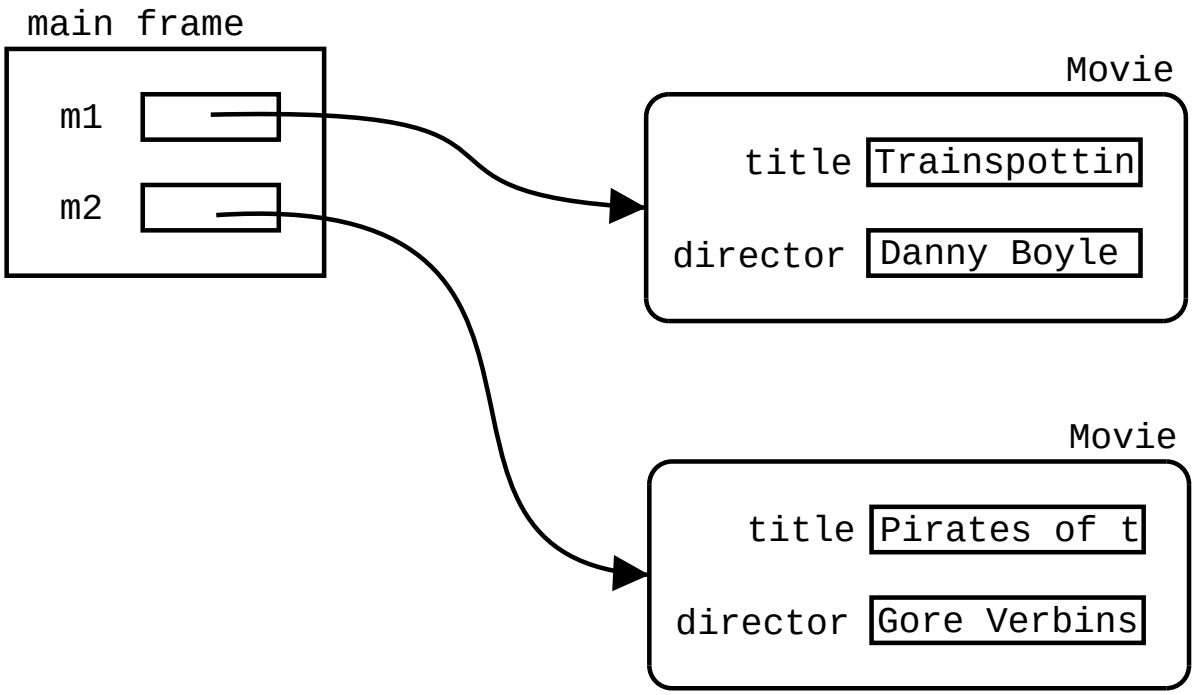
Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

Execution


main frame
m1
m2
print frame
this
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

Execution


main frame
m1
m2
Movie
title
Trainspottin
director
Danny Boyle
Movie
title
Pirates of t
director
Gore Verbins

Classes are data types

```
public class Theater {
  void play(Movie m)
  {
    m.print();
  }
}

public class MovieApplication3 {
  public static void main(String[] args)
  {
    Movie m1;
    Theater t = new Theater();
    m1 = new Movie(“Les Invasions barbares”,
                   “Denys Arcand”);
    t.play(m1);
  }
}
```

Aggregation

- Analisys:
  - Identify relevant information: objects and types of objects (classe)
  - Identify relationships between objects
- Different kinds of relationships depending on the type of the objects involved
- For example:
  - Numeric relationships:
    * account balance > 0
    * car fuel > 10
    * hitPoints <= maxHitPoints
    * number of heads <= 2
    * number of fingers > 1
    * tax_payable = base + (income - cutoff)*rate
  - Structural relationships:
    * A bank account *has a* balance and an owner
    * A car *has an* engine
    * A person *has a* name and a head

Classes are data types

```
class Car
{
    double speed = 0.0;
    boolean on = false;

    void start()
    {
      on = true;
    }
}
```

Objects and Aggregation

```
public class Engine {
    // ...
}
public class Car {
    Engine engine;
    // ...
}
public class StreetSimulation {
    void p()
    {
        Car mycar = new Car();
        mycar.engine = new Engine();
    }
}
```

Classes are data types

```
class Engine
{
    void turnon() { ... }
}

class Car
{
    double speed = 0.0;
    boolean on = false;
    Engine my_engine;

    Car()
    {
        my_engine = new Engine();
    }
    void start()
    {
      on = true;
      my_engine.turnon();
    }
}
```

Objects as first class values

- Objects can be attibutes of other objects

```
public class Rabbit {
  void jump() { ... }
}
public class Cage {
  Rabbit my_rabbit;
  void put(Rabbit a)
  {
    my_rabbit = a;
  }
  Rabbit get()
  {
    return my_rabbit;
  }
}
```

...elsewhere...

```
Rabbit bugs = new Rabbit();
Cage c = new Cage();
c.put(bugs);
Rabbit wester = c.get();
```

Classes are data types

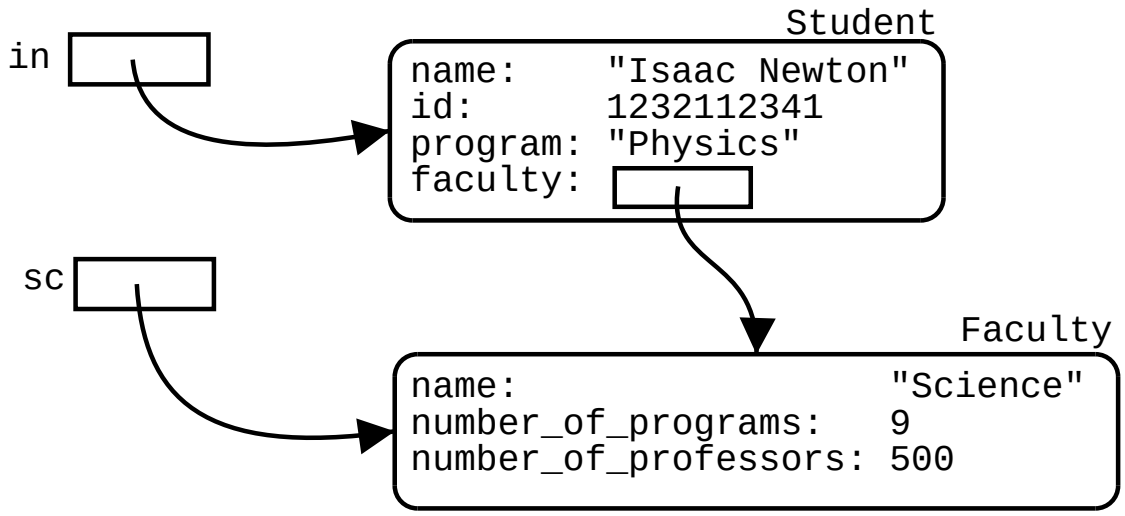
```
public class Student
{
    String name;
    long id;
    String program;
    String faculty;
    //...
    void set_prog_and_faculty(String p, String f)
    {
        program = p;
        faculty = f;
    }
    //...
}
```

Classes are data types

```
public class Faculty
{
    String name;
    int number_of_programs, number_of_professors;
    //...
}

public class Student
{
    String name;
    long id;
    String program;
    Faculty faculty;
    //...
    void set_prog_and_faculty(String p, Faculty f)
    {
        program = p;
        faculty = f;
    }
    //...
}
```

```
public class StudentDatabase
{
  public static void main(String[] args)
  {
    Faculty sc = new Faculty();

    sc.name = "Science";

    Student in = new Student("Isaac Newton",
                             1232112341);

    in.set_prog_and_faculty("Physics", sc);

    //...
    System.out.println(sc.name);
    System.out.println(in.name);
  }
}
```

Object structure in memory




```
in.set_prog_and_faculty("Physics", new Faculty());
```

doesn't create the variable `sc`, but then, the `Faculty` object cannot be shared between different `Student` objects.

The end