

REFERENTIES

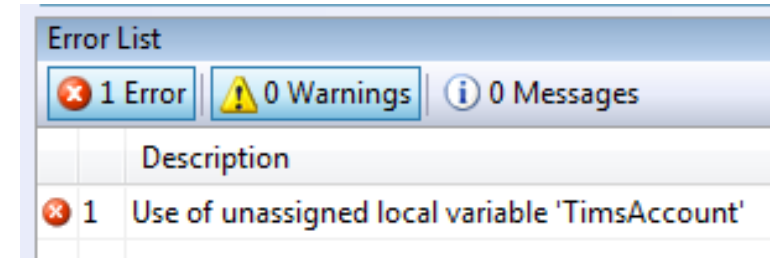
```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
    Account TimsAccount;
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```

- Classes

```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account TimsAccount;
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```

We maken een klasse Account aan (i.e. het model) en krijgen deze error.



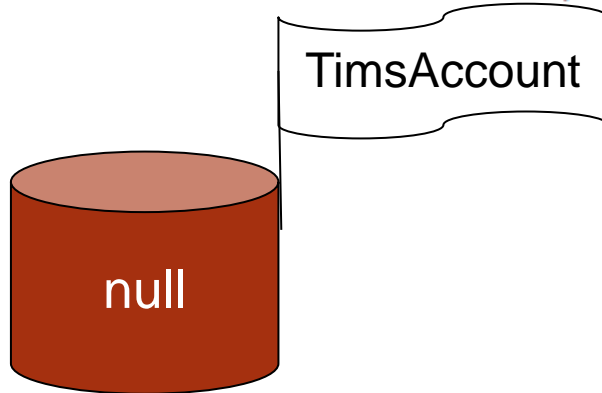
Struct by value, object by reference



- Classes

```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
```

```
    Account TimsAccount;
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```



Creëer een referentie (adrespointer) met naam `TimsAccount`.

Deze referentie wijst naar geheugenplek waar een instantie van het type `Account` zal staan.

Geheugen (heap)

Struct by value, object by reference

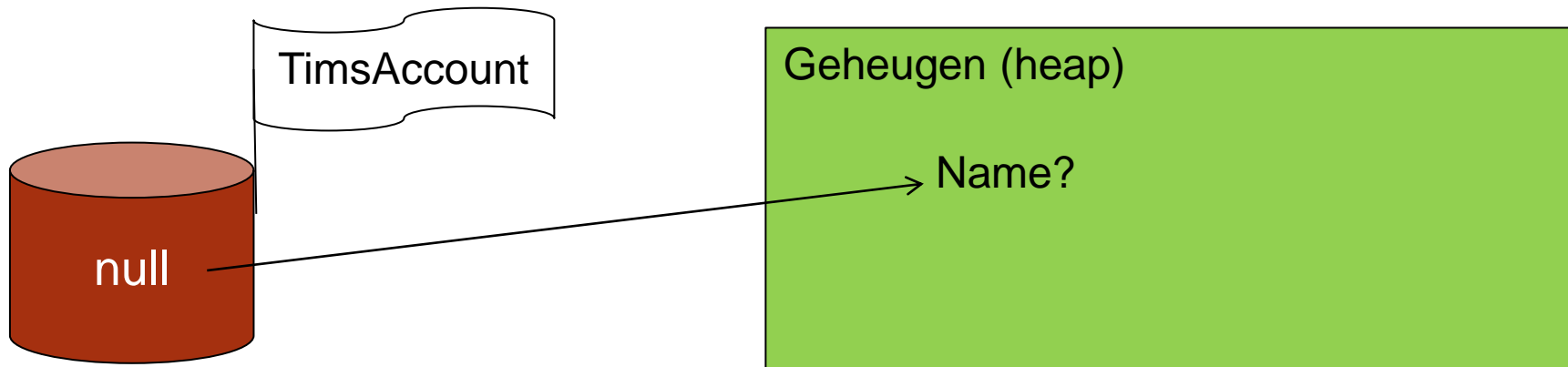


- Classes

```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
```

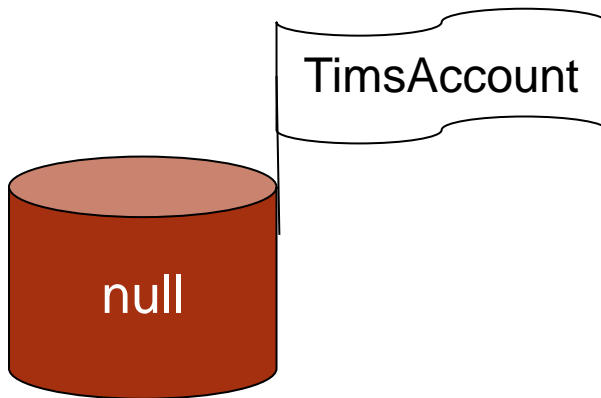
```
    Account TimsAccount;
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```

Het object waar TimsAccount naar verwijst in het geheugen is niet te vinden.



```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
```

```
    Account TimsAccount;
    TimsAccount = new Account();
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
```

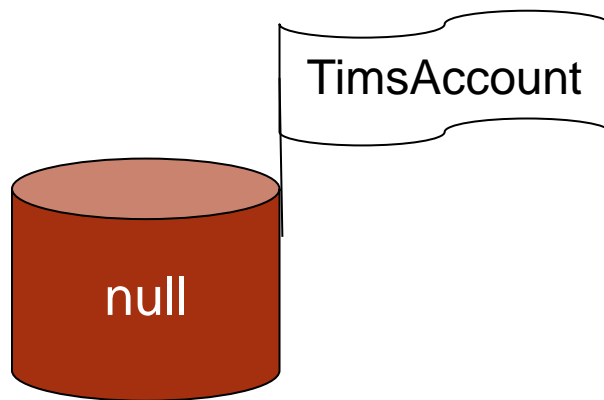


Maak een referentie met naam TimsAccount aan dat verwijst naar object van type Account.

Geheugen (heap)

```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
```

```
    Account TimsAccount;
    TimsAccount = new Account();
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```



Maak een object aan in het geheugen.

new doet 2 zaken:

- 1° Object in geheugen aanmaken
- 2° adres van object teruggeven

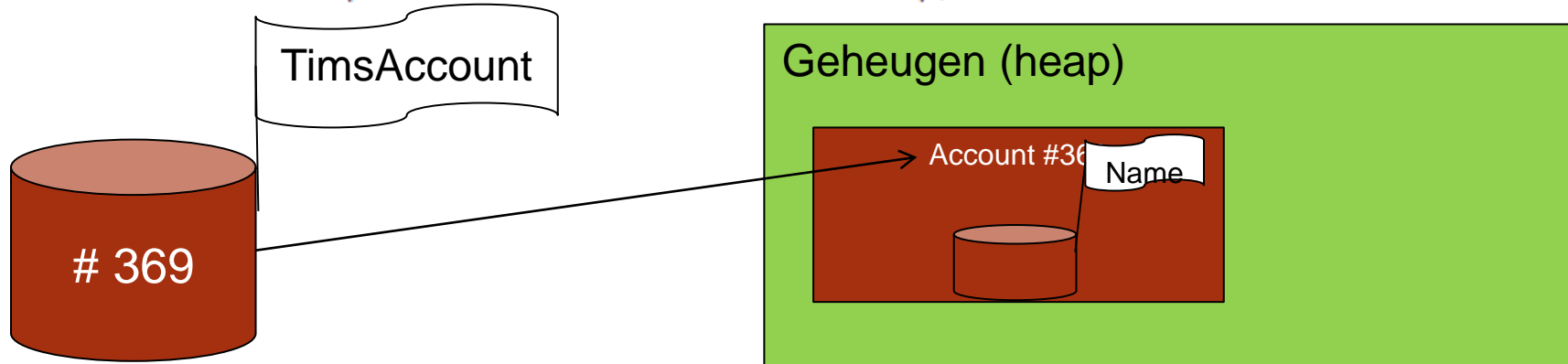
Geheugen (heap)



Wijs adres (=referentie) toe aan
TimsAccount

```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
```

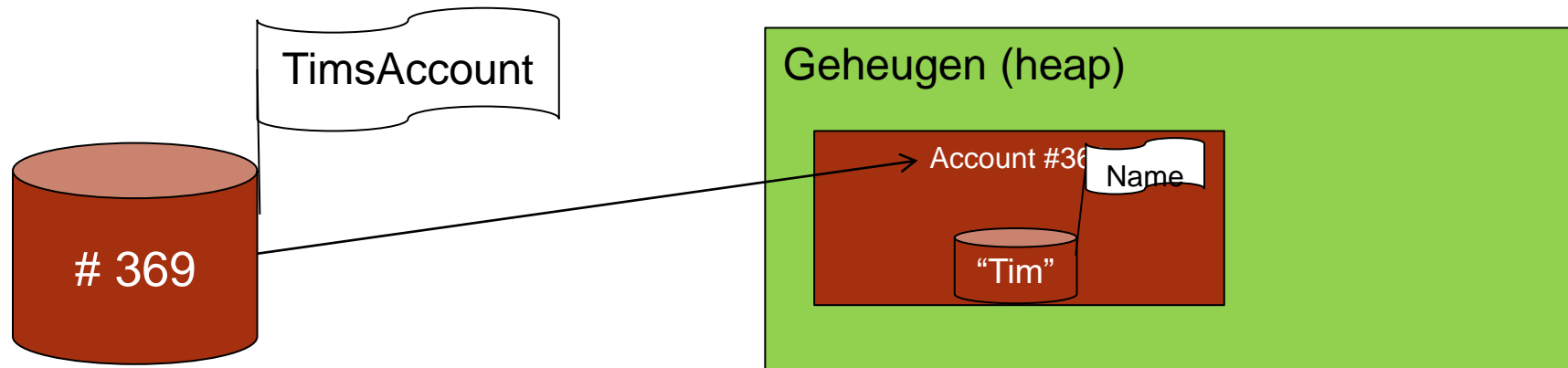
```
    Account TimsAccount;
    TimsAccount = new Account();
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```



Zet Name gelijk aan "Tim"

```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account TimsAccount;
    TimsAccount = new Account();
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```



Oplossing

```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{
    Account TimsAccount;
    TimsAccount = new Account();
    TimsAccount.Name = "Tim";
    Console.WriteLine("Name is: " + TimsAccount.Name);
}
```

THE TYPICAL REFERENCE BUG (?) OR IS IT

Stel volgende voorbeeld:



```
public class Account
{
    public string Name;
}
static void Main(string[] args)
{

    Account RobAccount;
    RobAccount = new Account();
    RobAccount.Name = "Rob";
    Console.WriteLine(RobAccount.Name);

    Account Temp;
    Temp = RobAccount;
    Temp.Name = "Jim";

    Console.WriteLine(Temp.Name);
    Console.WriteLine(RobAccount.Name);
}
```

- PS kan je output voorspellen zonder naar volgende slide te gaan?

We werken met referenties!

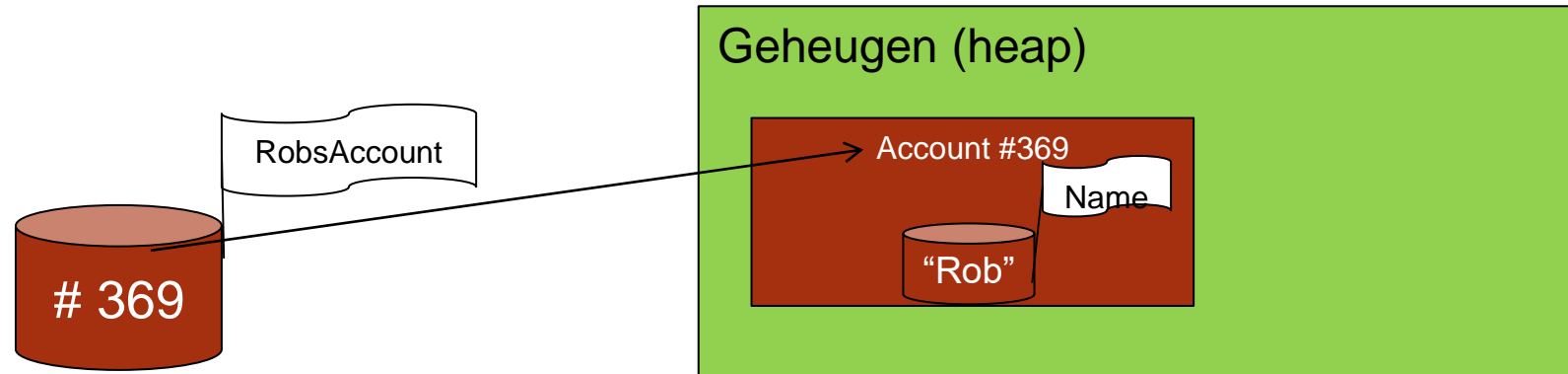


```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```



We werken met referenties!



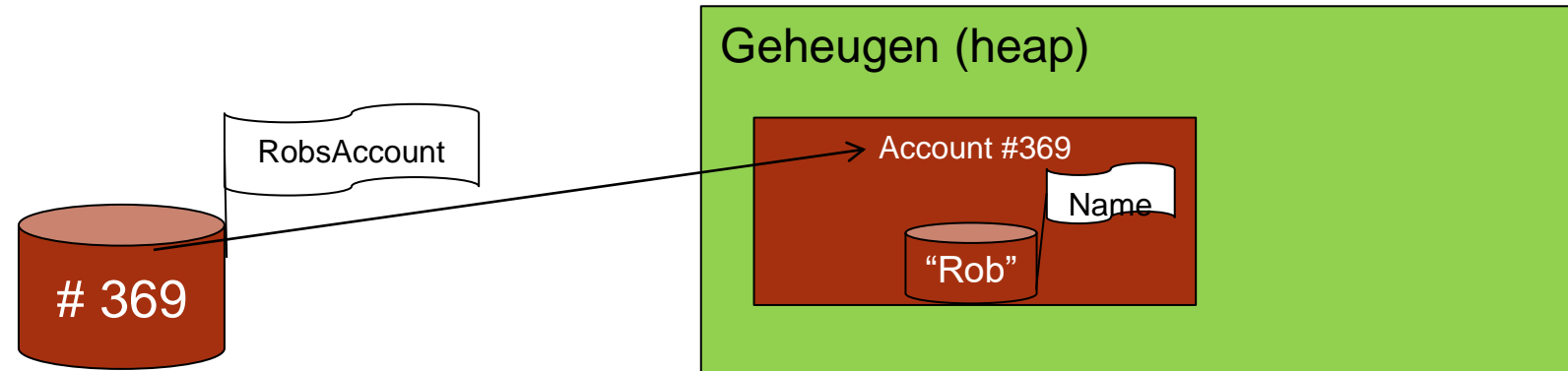
```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

C:\Windows\system32\cmd.exe
Rob



We werken met referenties!



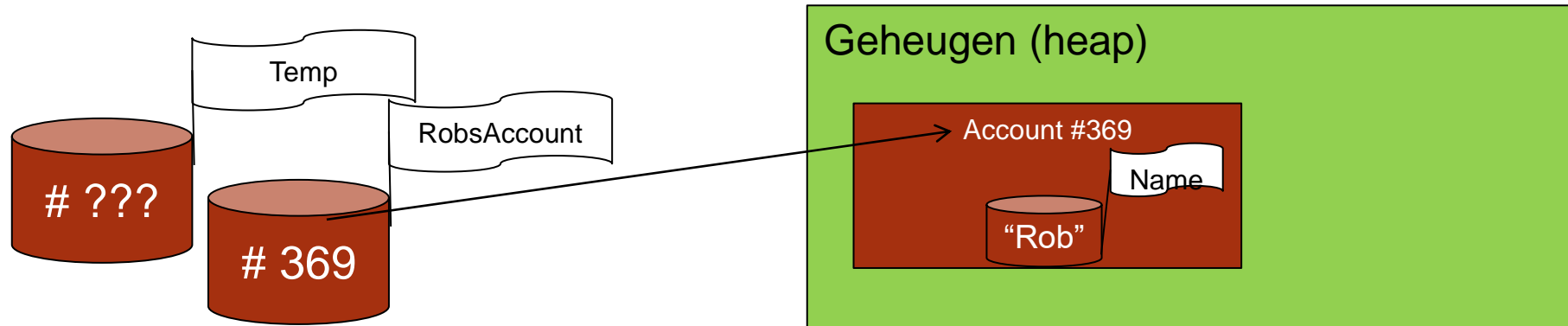
```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

```
C:\Windows\system32\cmd.exe
Rob
```



We werken met referenties!



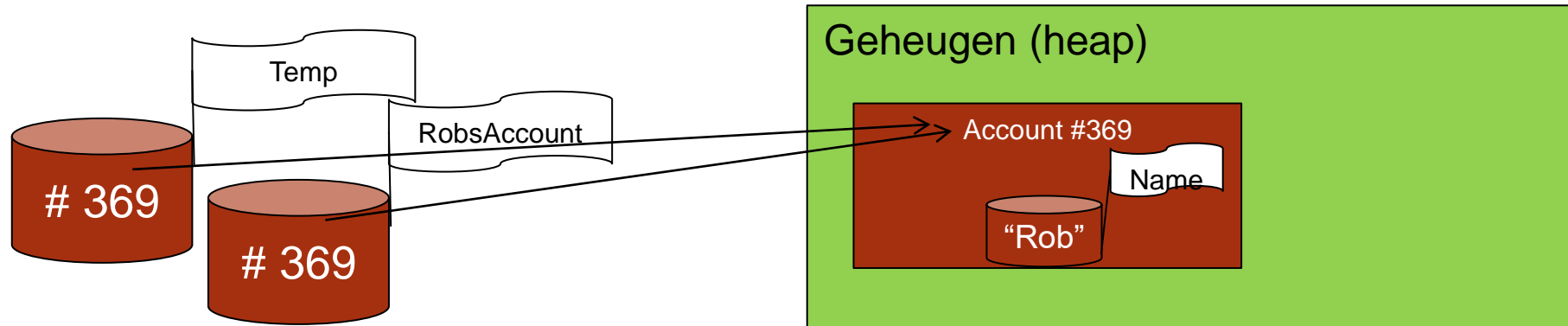
```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

```
C:\Windows\system32\cmd.exe
Rob
```



We werken met referenties!



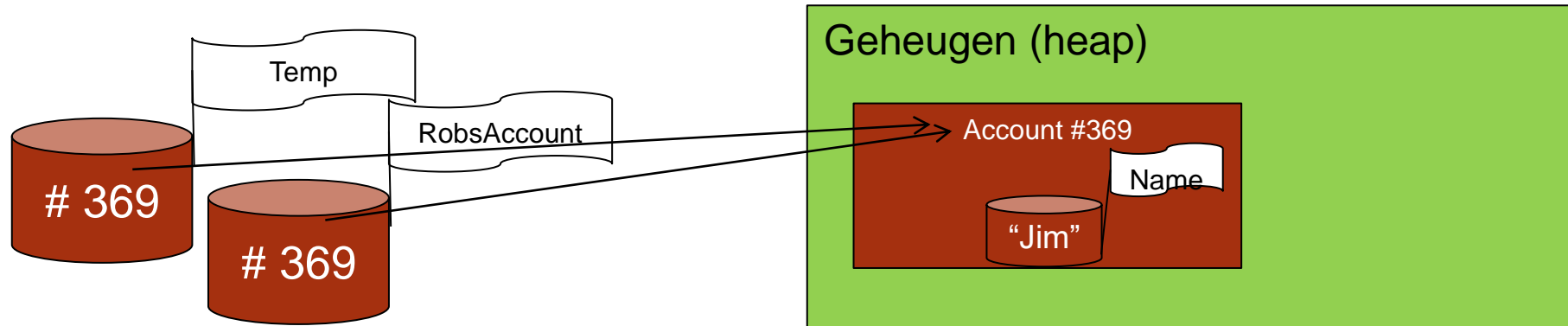
```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

```
C:\Windows\system32\cmd.exe
Rob
```



We werken met referenties!

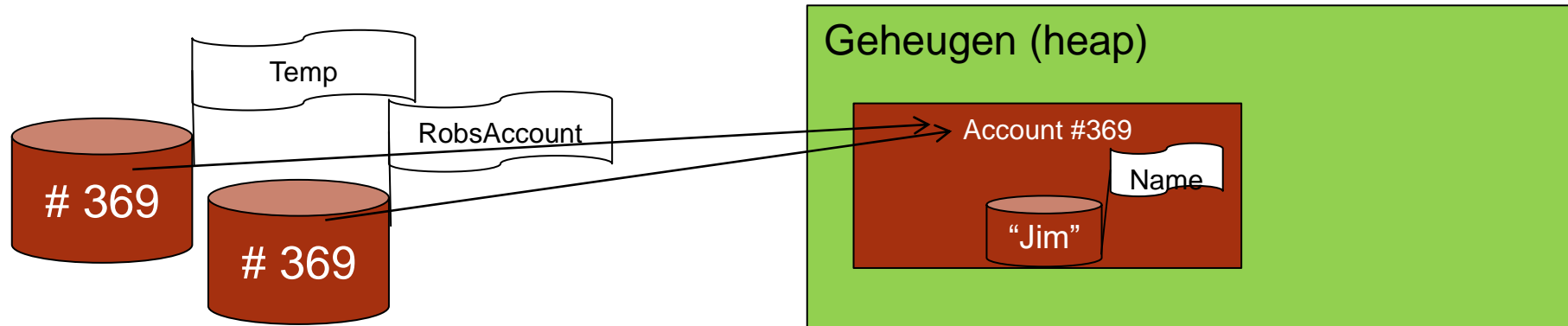


```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";
    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

```
C:\Windows\system32\cmd.exe
Rob
Jim
```



We werken met referenties!



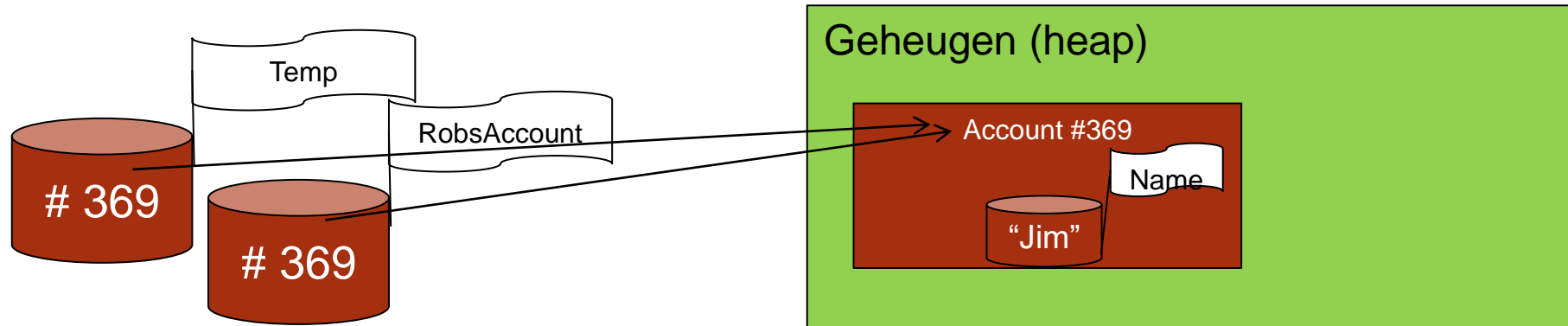
```
public class Account
{
    public string Name;
}

static void Main(string[] args)
{
    Account RobsAccount;
    RobsAccount = new Account();
    RobsAccount.Name = "Rob";
    Console.WriteLine (RobsAccount.Name);

    Account Temp;
    Temp = RobsAccount;
    Temp.Name = "Jim";

    Console.WriteLine (Temp.Name);
    Console.WriteLine (RobsAccount.Name);
}
```

```
C:\Windows\system32\cmd.exe
Rob
Jim
Jim
```



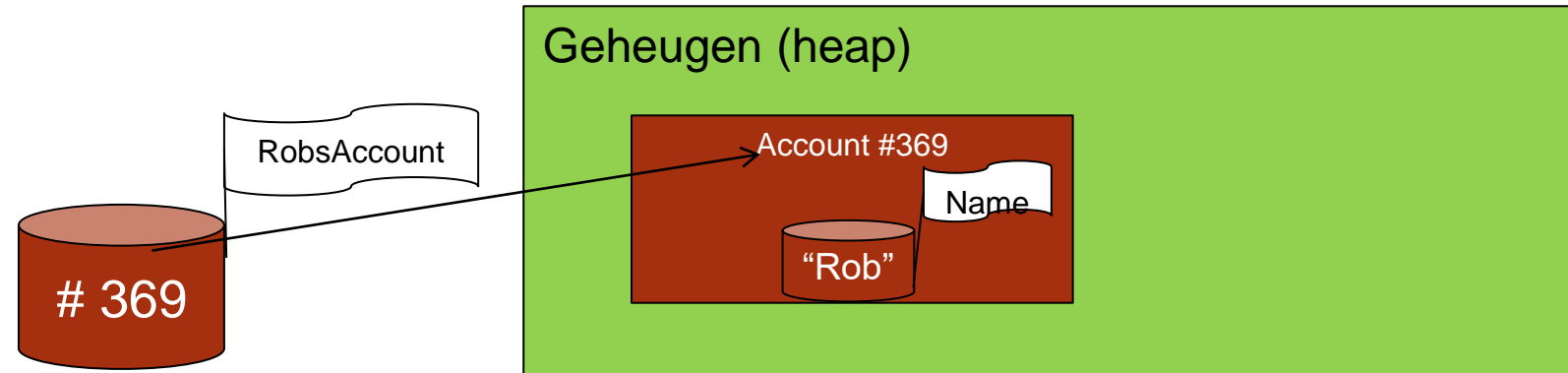
**AND HOW DOES THIS RELATE TO THE
GARBAGE COLLECTOR?**



- Wanneer een object werd aangemaakt (mbv new) en er zijn geen referenties naar dit object, dan wordt het automatisch verwijderd.
 - Dit heet 'garbage collection' (zie heap/stack eerder)
- Vergelijk met de scope van een lokale variabele (scope is binnen de accolades waar variabele werd aangemaakt)

```
Account RobsAccount;  
RobsAccount = new Account();  
RobsAccount.Name = "Rob";  
Console.WriteLine(RobsAccount.Name);
```

```
RobsAccount = new Account();  
RobsAccount.Name = "Jim";  
Console.WriteLine(RobsAccount.Name);
```

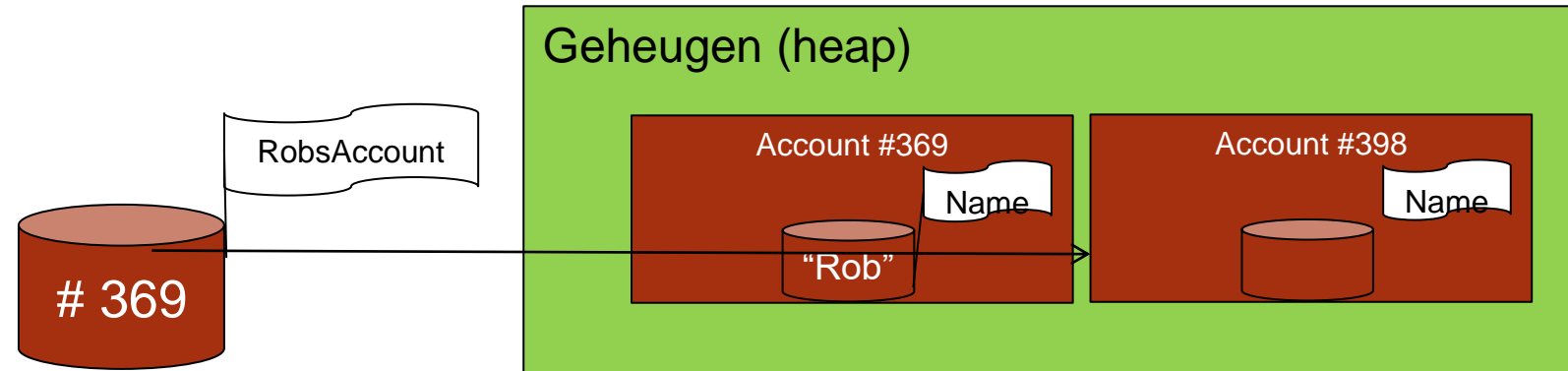


Garbage collector



```
Account RobsAccount;  
RobsAccount = new Account();  
RobsAccount.Name = "Rob";  
Console.WriteLine(RobsAccount.Name);
```

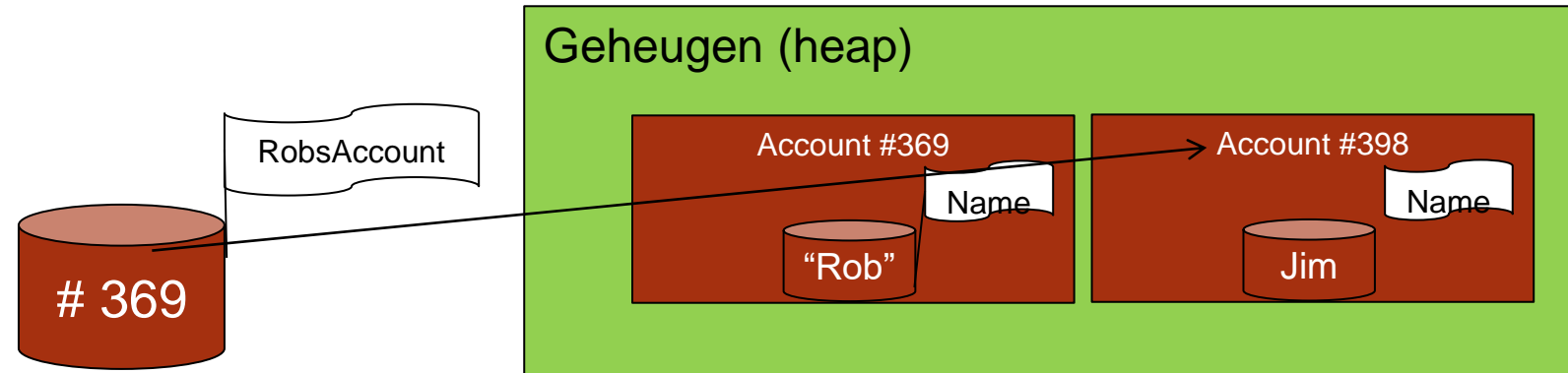
```
RobsAccount = new Account();  
RobsAccount.Name = "Jim";  
Console.WriteLine(RobsAccount.Name);
```



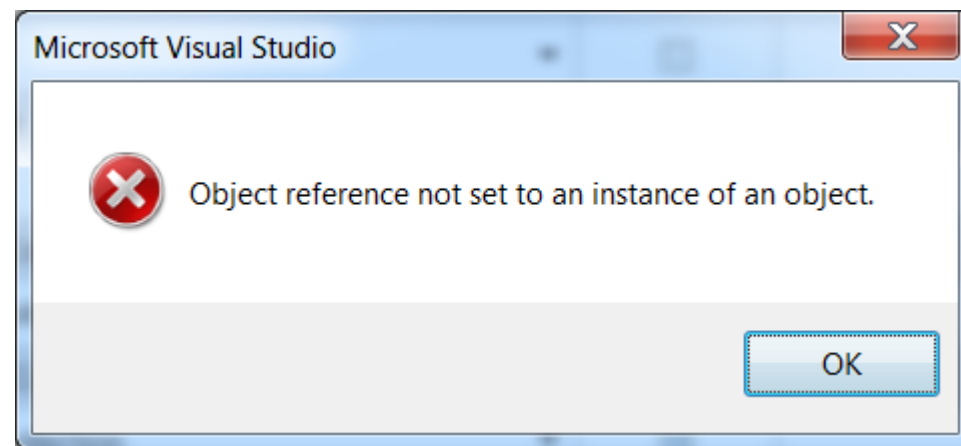
Garbage collector



```
Account RobsAccount;  
RobsAccount = new Account();  
RobsAccount.Name = "Rob";  
Console.WriteLine(RobsAccount.Name);  
  
RobsAccount = new Account();  
RobsAccount.Name = "Jim";  
Console.WriteLine(RobsAccount.Name);
```

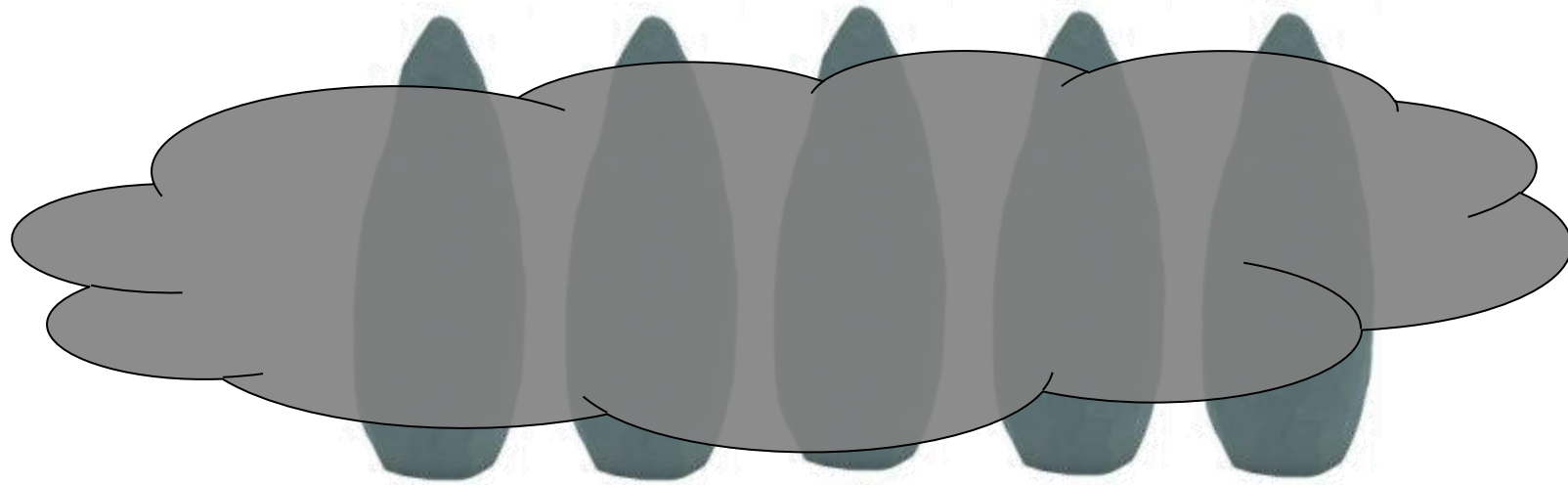


- Objecten worden steeds by reference gebruikt in C#, overall!



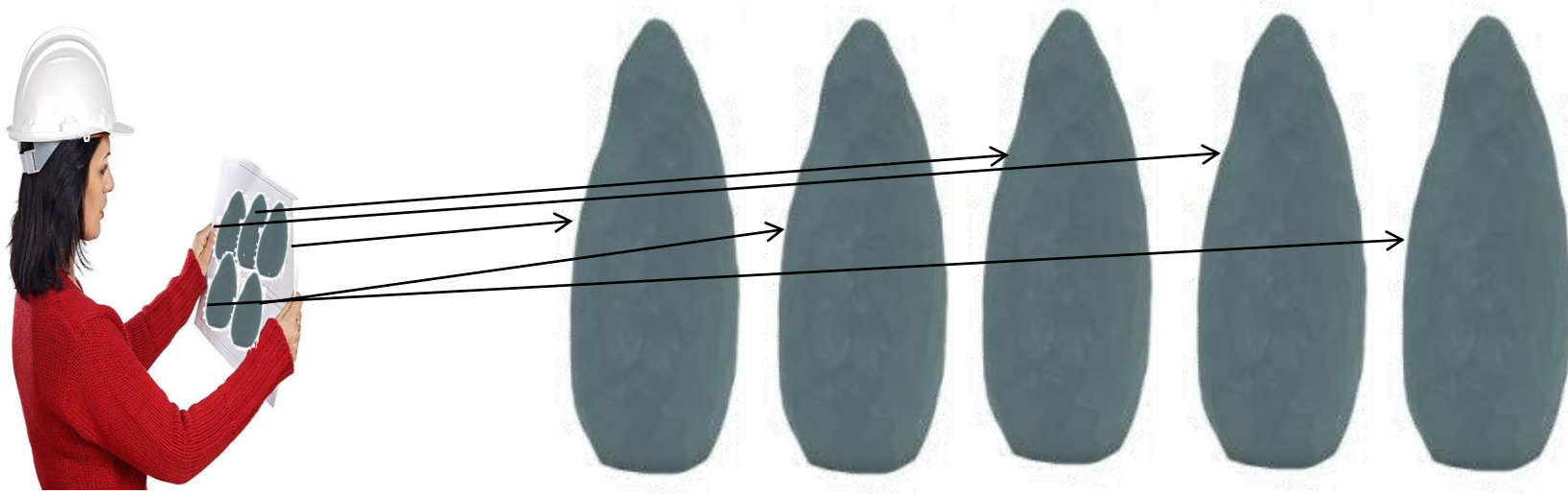
- Krijg je bovenstaande Exception: je probeert een object aan te roepen dat niet bestaat.

- Menhir= groot object in geheugen



- **Zonder referenties:** Deze van plaats veranderen (bv sorteren) vergt veel kracht (kopieerwerk)

- Menhir= groot object in geheugen



- **Met referenties:** menhires blijven staan, referenties wijzen gewoon naar ander object

NULL EN NULL REFERENCE EXCEPTION

- Veel voorkomende fout.
- **Null** geeft aan dat een object verwijst naar 'niets' (null, nada, noppes)
- Maw, je hebt nooit een object geïntantieerd met new operator.
- Oplossing:

```
Location = new Point(1,1);
```



NullPointerException was unhandled