Kat - the language of calculations

Mikus Vanags

Parameter declaration explicitly

```
function(x) {return x;}
```

Parameter declaration implicitly

```
function() {return x;}
```

```
x
```

Solution without redundant syntactic code parts

Parameter order in parameters list

Order in which parameters are used in function body are different

Parameter declaration explicitly

```
function(x, y) {return y - x;}
```

Postfix form of Grace~ operator

Parameter declaration implicitly

```
function() {return y~ - x;}
function() {return y - ~x;}
```

Prefix form of Grace~ operator

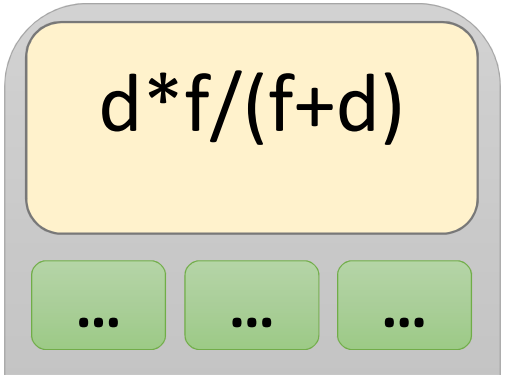
Simplest use case of implicit parameters: super calculator

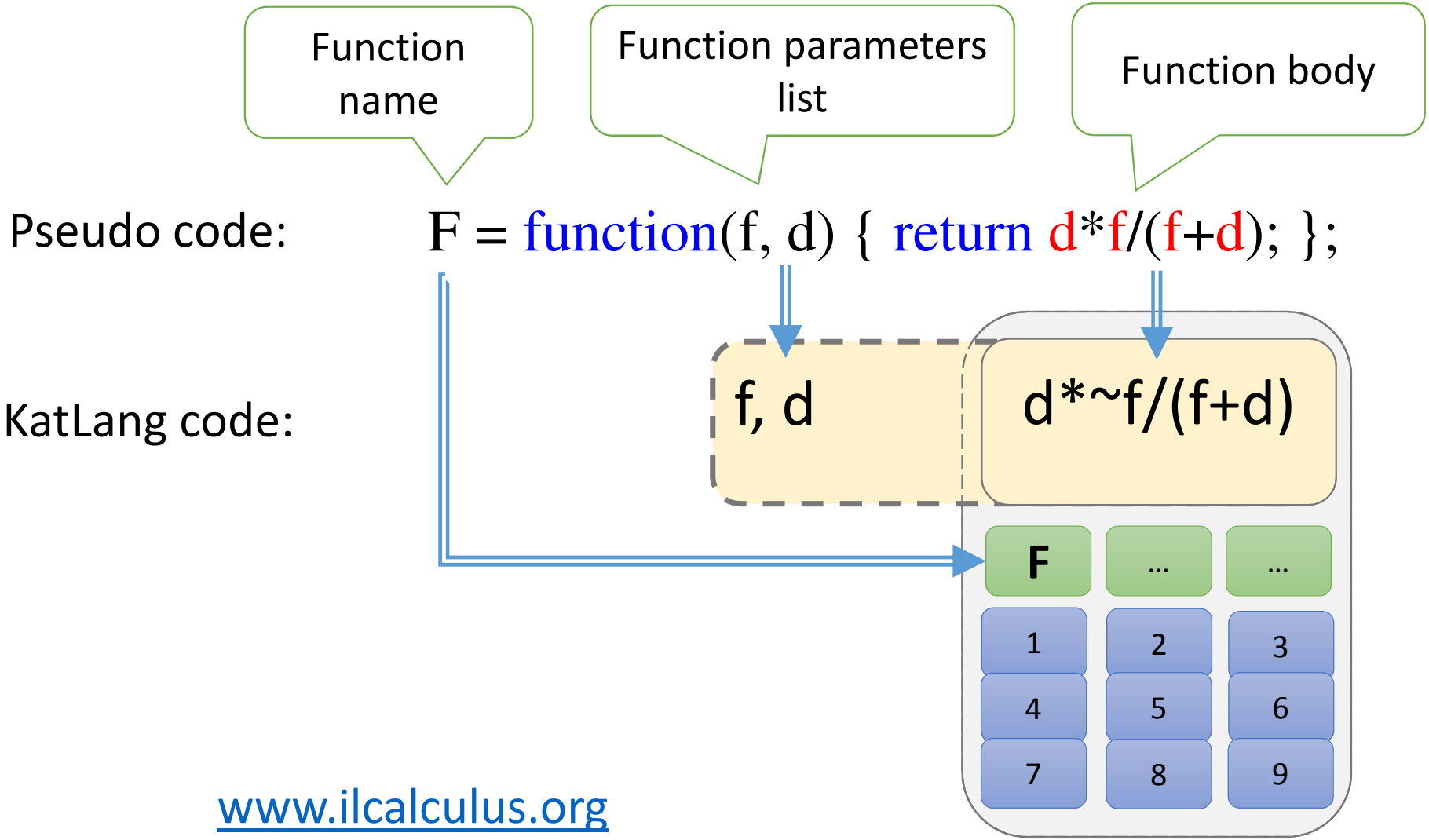
Lens focus

$$F = \frac{d * f}{f + d}$$

KatLang code:

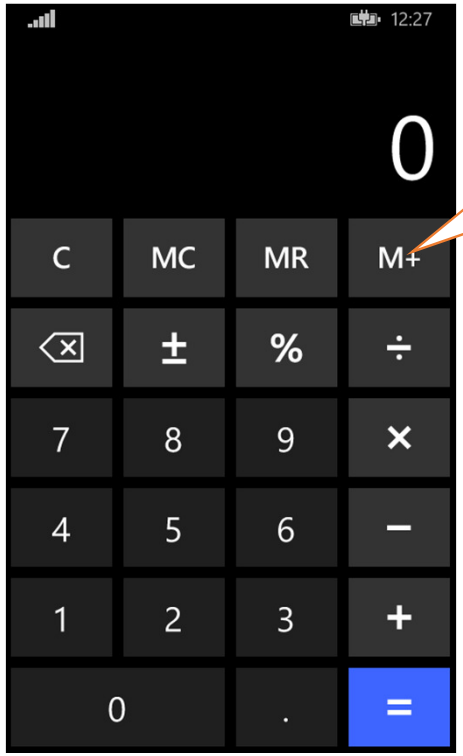
```
F=d*f/(f+d);
```





www.ilcalculus.org

Kat calculator demonstration



Touch me like old electronic calculator!

Drag me like using modern smartphone!

Windows Phone 8.1 standard calculator

Il Calculus 1.3 Windows Phone 8.1

www.ilcalculus.org

C# 3.0:

Parameter declaration explicitly

Lambda symbol

```
people.Sort((x, y) => x.LastName.CompareTo(y.LastName));
```

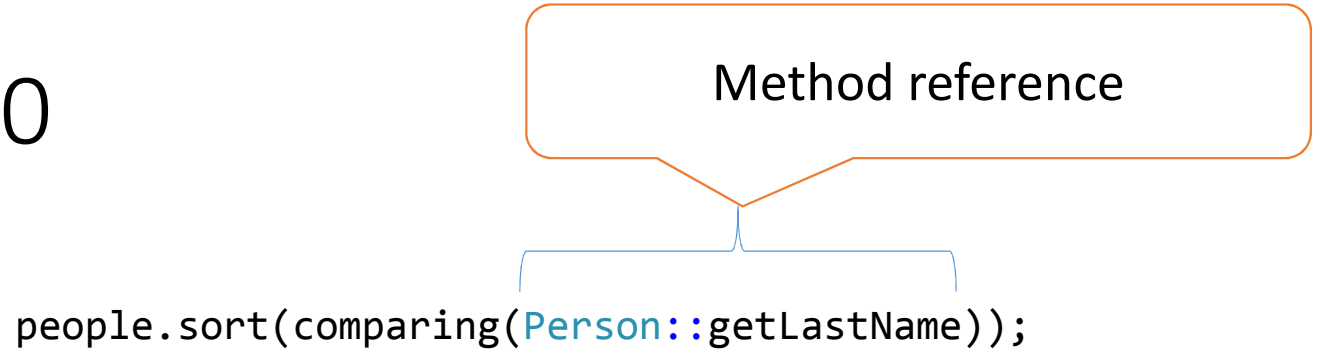
Potential of implicit parameters (pseudo code C):

Beginning of lambda expression

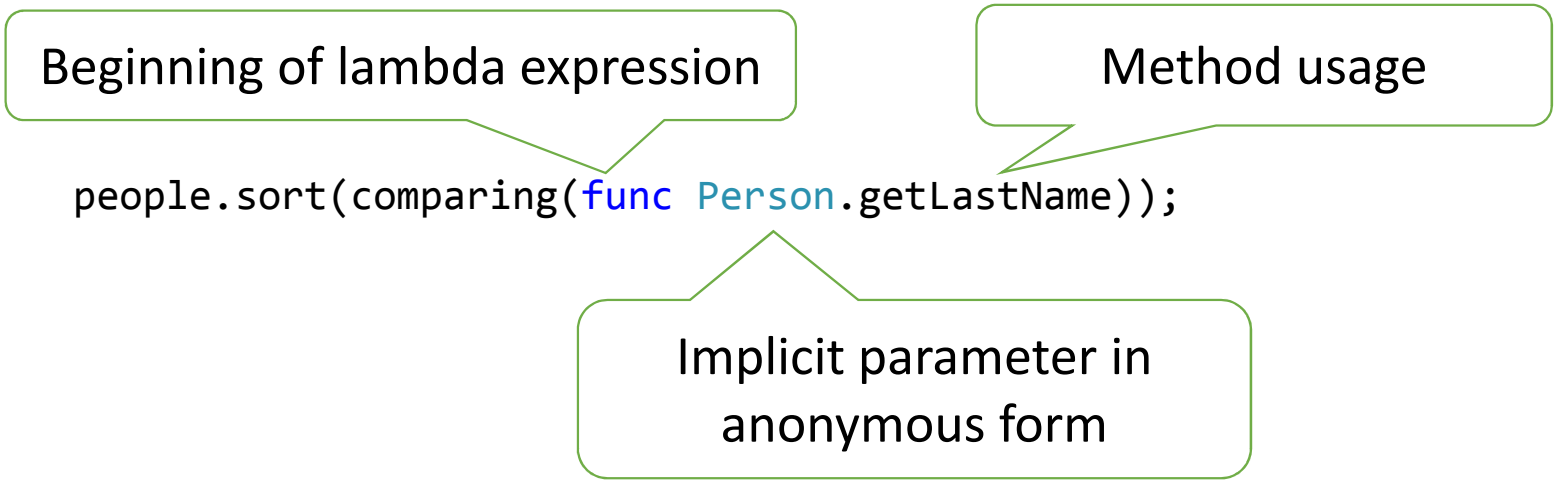
```
people.Sort(func x.LastName.CompareTo(y.LastName));
```

Implicit parameter in canonical form

Java 8.0



Potential of implicit parameters (pseudo code J):



Swift

```
let names = ["Chris", "Alex", "Ewa", "Barry", "Daniella"]

var reversed = sorted(names, { $0 > $1 } )
```

Shorthand argument name

Potential of implicit parameters (pseudo code S):

```
var reversed = sorted(names, { current > next } )
```

Implicit parameter in canonical form

If You are interested in good abstractions in symbolic computation and programming languages,
email to:
mikus.vanags@logicsresearchcentre.com