MicroJava VM: Speicher-Layout


```
code →  ▢        data →  ▮        heap →  ▮        pstack →         estack →  ▮
        ▢                ▮                ▮                                   ▮
pc  →   ▢                                                                esp →
        ▢                        free →                   ra
        ▢                                         fp →    dl
        ▢                                         sp →
```

| Code | StaticData | Heap | ProcStack | ExprStack |
| (Bytearray) | (Wortarray) | (Wortarray) | (Wortarray) | (Wortarray) |
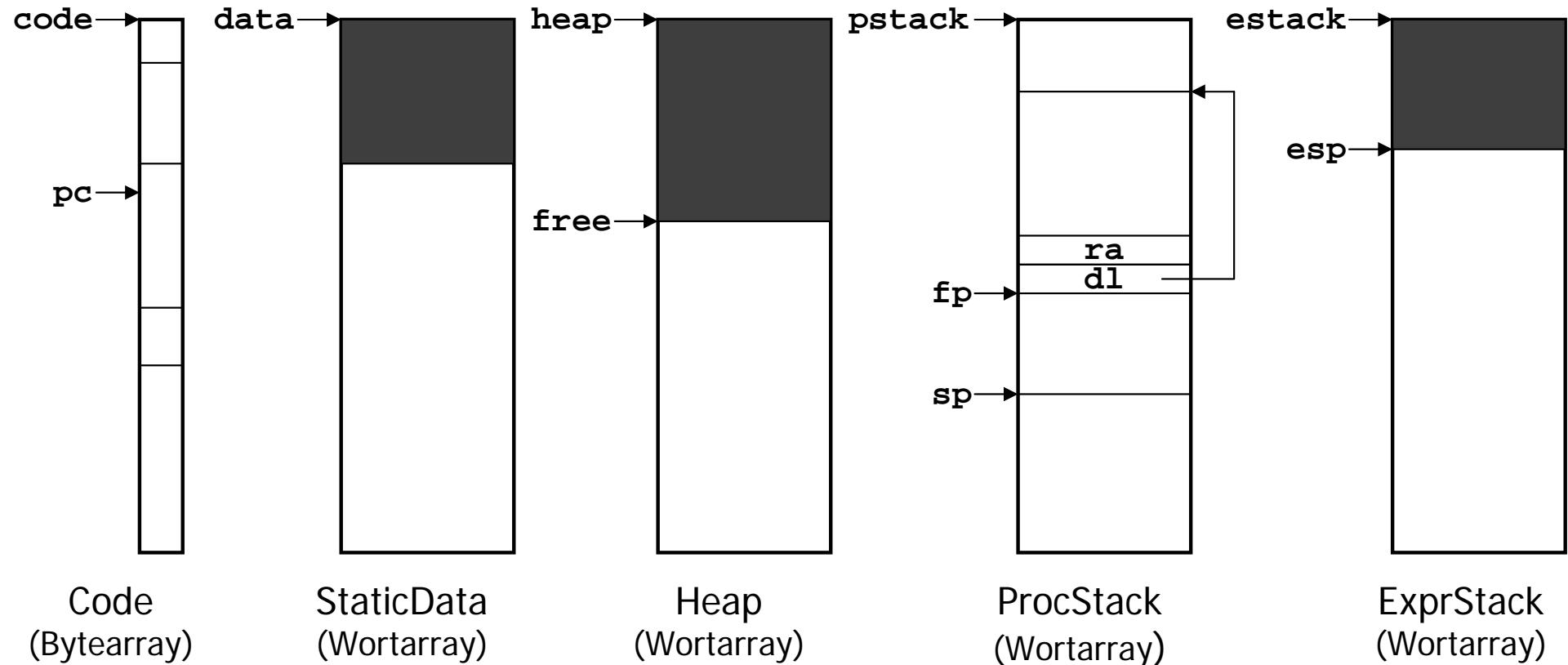
Symboltabelle

Deklaration: **program A**
final int max = 12;   *// Konstante*
char c; int i;   *// globale Variablen*
class B { int x, y; }   *// innere Klasse mit Feldern*
{ void foo () int[] iarr; B b; int n; {...} }

```
           ⊤
         ┌────┐
         │  - │──────►
         └────┘
            ▲
         ┌────┐   Con      Var      Var      Type                    Meth
         │  2 │   "max"    "c"      "i"      "B"                     "foo"
         └────┘   intType  charType intType                         noType
            ▲     ──►──    ──►──    ──►──    ──┐           ──►──    ──┤
                  12       0        1        │                        -
                  -        0        0        │                        -
                  -        -        -        │                        -
                  -        -        -        -▼         Class         -
                                                        -
                              Arr                       2
                              intType
                           ─► -                 ──►
         ┌────┐   Var       -         Var      Var                    Var      Var
         │  3 │   "iarr"              "b"      "n"                    "x"      "y"
         └────┘                                intType                intType  intType
            ▲     ──►──               ──►──    ──┤        ──►──       ──►──    ──┤
                  0                   1        2                      0        1
                  1                   1        1                      1        1
                  -                   -        -                      -        -
                  -                   -        -                      -        -
```
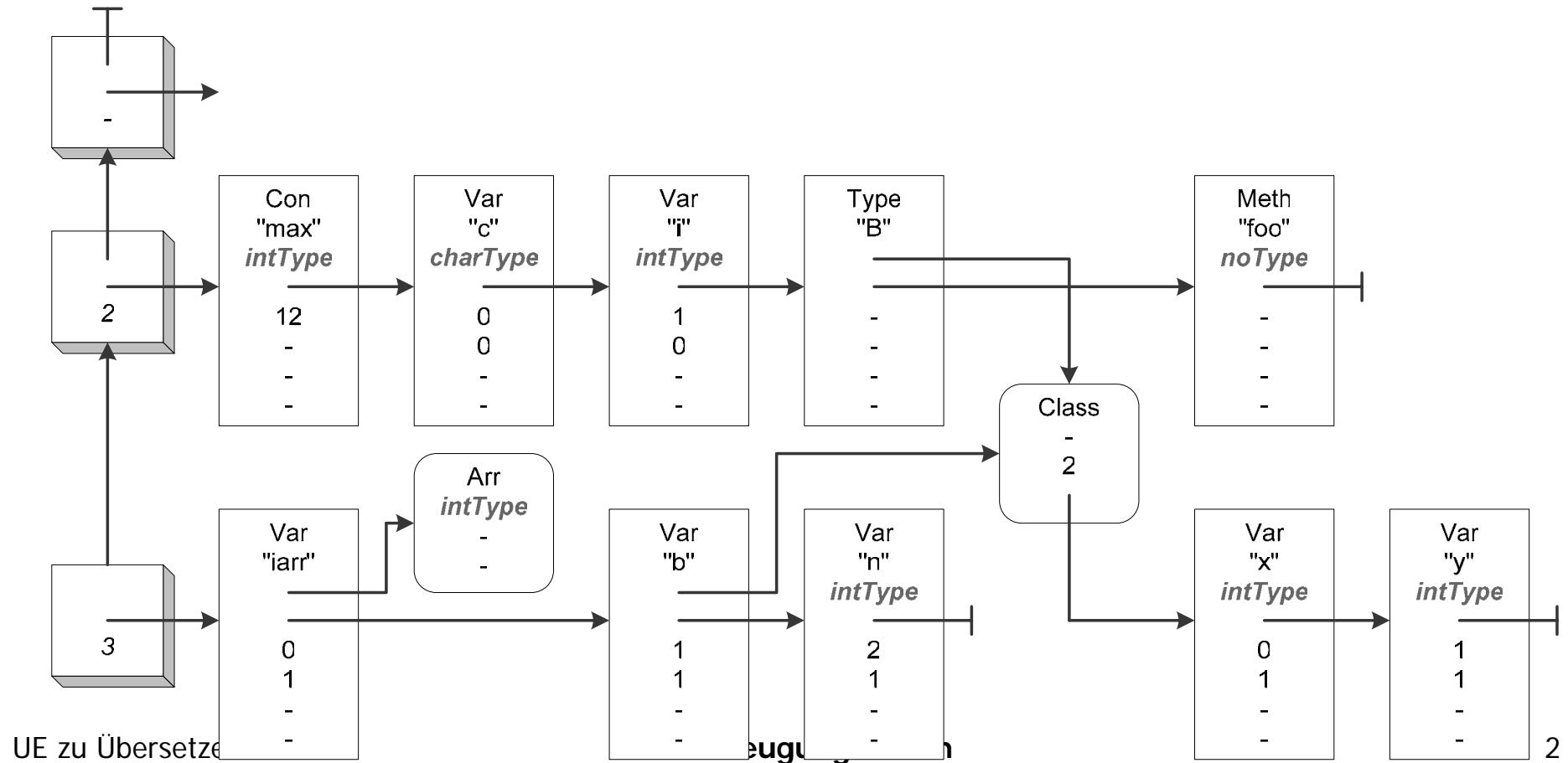
Bsp 1:   **n = 3**;

Deklaration: **program A**

      final int max = 12;   *// Konstante*

      char c; int i;   *// globale Variablen*

      class B { int x, y; }   *// innere Klasse mit Feldern*

    {   void foo ()   int[] iarr; B b; int n;   {...}   }

const_3   = **2** byte

store_2

Bsp 2:   i = 10;

Deklaration: program A

      final int max = 12;   *// Konstante*
      char c; int i;   *// globale Variablen*
      class B { int x, y; }   *// innere Klasse mit Feldern*
{   void foo ()   int[] iarr; B b; int n;   {…}   }

const   10   = **8** byte
putstatic   1

Bsp 3:   n = 3 + i;

Deklaration: **program A**

      final int max = 12;   *// Konstante*

      char c; int i;   *// globale Variablen*

      class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo ()   int[] iarr; B b; int n;   {...}   }

const_3   = **6** byte
getstatic   1
add
store_2

Bsp 4: n = 3 + i * max - n;

Deklaration: **program A**

    final int max = 12;   *// Konstante*

    char c; int i;   *// globale Variablen*

    class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo () int[] iarr; B b; int n; {...} }

const_3   = **14** byte

getstatic 1

const 12

mul

add

load_2

sub

store_2

Bsp 5: iarr[5] = 10;

Deklaration: **program A**

    final int max = 12;   *// Konstante*

    char c; int i;   *// globale Variablen*

    class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo () int[] iarr; B b; int n; {...} }

```
load_0                                    = 8 byte
const_5
const  10
astore
```

Bsp 6: b.y = iarr[5] * 3;

Deklaration: **program A**
final int max = 12; *// Konstante*
char c; int i; *// globale Variablen*
class B { int x, y; } *// innere Klasse mit Feldern*
{ void foo ()   int[] iarr; B b; int n;   {…}   }

load_1 = **9** byte
load_0
const_5
aload
const_3
mul
putfield   1

Bsp 7:   n--;

Deklaration: **program A**

      final int max = 12;   *// Konstante*

      char c; int i;   *// globale Variablen*

      class B { int x, y; }   *// innere Klasse mit Feldern*

{   void foo ()   int[] iarr; B b; int n;   {...}   }

inc   2   255   = **3** byte

Bsp 8: **i--**;

Deklaration: **program A**

      final int max = 12;   *// Konstante*

      char c; int i;   *// globale Variablen*

      class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo () int[] iarr; B b; int n; {…} }

getstatic 1   **= 8** byte

const_m1

add

putstatic 1

Bsp 9: **b.y--**;

Deklaration: **program A**

final int max = 12;   *// Konstante*

char c; int i;   *// globale Variablen*

class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo () int[] iarr; B b; int n; {...} }

load_1   = **10** byte

dup

getfield 1

const_m1

add

putfield 1

Bsp 10: iarr[0]--;

Deklaration: **program A**

final int max = 12;   *// Konstante*

char c; int i;   *// globale Variablen*

class B { int x, y; }   *// innere Klasse mit Feldern*

{ void foo () int[] iarr; B b; int n; {...} }

load_0   = **7** byte

const_0

dup2

aload

const_m1

add

astore