



# Metody kompilacji

## Wykład 14, Elementy obiektowe w kompilatorach

Włodzimierz Bielecki, Piotr Błaszyński

Wydział Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego

8 czerwca 2020



# Czy klasa jest strukturą?

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

- Tak
- Z punktu widzenia kompilatora:
  - tak, z dodatkowymi elementami.
- Z punktu widzenia semantyki języka:
  - raczej tak, z dodatkowymi elementami.



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Kod w C++, dla przykładowego kodu:

```
class example
{
    int x;
    float z;
    int y;
};
example e1;
e1.y = 42;
e1.x = 10;

example e2;
e2.y = 242;
e2.x = 210;
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Należy wygenerować kod:

```
.data
    e1:      .space 12
    e2:      .space 12

.text
    li $t0, 42
    la $t4, e1
    add $t4, $t4, 8
    sw $t0, ($t4)

    li $t0, 10
    la $t4, e1
    add $t4, $t4, 0
    sw $t0, ($t4)
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

dalszy ciąg:

```
li $t0, 242
la $t4, e2
add $t4, $t4, 8
sw $t0, 8($t4)

li $t0, 210
la $t4, e2
add $t4, $t4, 0
sw $t0, ($t4)
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Stosując uproszczenie procesu generowania można wygenerować krótszy kod (ponieważ nie ma potrzeby indeksowania wyrażeniem):

```
.data
    e1:      .space 12
    e2:      .space 12
.text
    li $t0, 42
    la $t4, e1
    sw $t0, 8($t4)

    li $t0, 10
    la $t4, e1
    sw $t0, 0($t4)
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Dalszy ciąg:

```
li $t0, 142  
la $t4, e2  
sw $t0, 8($t4)
```

```
li $t0, 210  
la $t4, e2  
sw $t0, 0($t4)
```



# Generowanie kodu klas

Metody  
kompilacji

Rzeczywisty kod w C++, przetwarzany popularnym kompilatorem (gdbolt):

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

```
class example
{
public:
    int x;
    float z;
    int y;
};

example e1;
example e2;

int adder() {
    e1.y = 42;
    e1.x = 10;
    e2.y = 242;
    e2.x = 210;
    return e1.y + e1.x + e2.y + e2.x;
}
```





# Generowanie kodu klas

Metody  
kompilacji

Wygenerowany kod:

```
adder():
    addiu    $sp, $sp, -8
    sw      $fp, 4($sp)
    move    $fp, $sp
    lui     $2, %hi(e1)
    addiu   $2, $2, %lo(e1)
    li      $3, 42                # 0x2a
    sw      $3, 8($2)
    lui     $2, %hi(e1)
    li      $3, 10                # 0xa
    sw      $3, %lo(e1)($2)
    lui     $2, %hi(e2)
    addiu   $2, $2, %lo(e2)
    li      $3, 242               # 0xf2
    sw      $3, 8($2)
    lui     $2, %hi(e2)
    li      $3, 210               # 0xd2
```

Klasy

Przykładowy kod

Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Wygenerowany kod, ciąg dalszy:

```
sw      $3,%lo(e2)($2)
lui    $2,%hi(e1)
addiu  $2,$2,%lo(e1)
lw     $3,8($2)
lui    $2,%hi(e1)
lw     $2,%lo(e1)($2)
nop
addu   $3,$3,$2
lui    $2,%hi(e2)
addiu  $2,$2,%lo(e2)
lw     $2,8($2)
nop
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Wygenerowany kod, ciąg dalszy:

```
addu    $3, $3, $2
lui     $2, %hi(e2)
lw      $2, %lo(e2)($2)
nop
addu    $2, $3, $2
move    $sp, $fp
lw      $fp, 4($sp)
addiu   $sp, $sp, 8
j       $31
nop
```



# Generowanie kodu klas

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

Kod wygenerowany dla procesorów z rodziny x86:

```
e1:      .zero    12
e2:      .zero    12
adder():
    push   rbp
    mov    rbp,   rsp
    mov    DWORD PTR e1[rip+8], 42
    mov    DWORD PTR e1[rip], 10
    mov    DWORD PTR e2[rip+8], 242
    mov    DWORD PTR e2[rip], 210
    mov    edx,  DWORD PTR e1[rip+8]
    mov    eax,  DWORD PTR e1[rip]
    add    edx,  eax
    mov    eax,  DWORD PTR e2[rip+8]
    add    edx,  eax
    mov    eax,  DWORD PTR e2[rip]
    add    eax,  edx
    pop    rbp
    ret
```



# Ale co z metodami?

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

- Metody są zwykłymi funkcjami, dodatkowo otrzymują dodatkowy niejawnny
  - lub przynajmniej niejawnie przekazywany argument:
    - obiekt, na którego rzecz są wywoływane,
    - argument ten jest zwykle nazywany „this” lub „self” w obrębie tej funkcji (czyli metody).
    - symbole o zasięgu klasowym mogą być (C++, JavaScript) lub nie (PHP, Python) dostępne domyślnie w ramach metod.



# Dziedziczenie

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

- Dziedziczenie polega zasadniczo na sklejeniu struktur i ewentualnie także scalaniu tablic symboli,
- w zależności od języka są lub nie są dostępne symbole klasy bazowej z poziomu metod klasy, która jest analizowana.



# Zasięg symboli

Metody  
kompilacji

Klasy

Przykładowy kod  
Metody

Dziedziczenie

Zasięg symboli

Metody  
wirtualne

- W momencie napotkania symbolu (pola lub metody) w metodzie,
  - można wykonać wyszukiwanie w górę, zaczynając od bieżącej klasy, idąc w górę hierarchii.
  - można to sprawdzać tylko w jednej tablicy symboli, która jest wynikiem połączenia.



# Metody wirtualne

## Metody kompilacji

### Klasy

Przykładowy kod  
Metody

### Dziedziczenie

### Zasięg symboli

### Metody wirtualne

- Metody wirtualne są wywoływane pośrednio.
- W niektórych językach wszystkie metody są domyślnie wirtualne.
- Implementacja zależy od tego, czy jest to w pełni dynamiczny język, w takim przypadku zawsze wyszukujemy nazwę funkcji w klasie w czasie wykonywania i dlatego wszystkie metody stają się wirtualne automatycznie,
- w przypadku języków statycznych kompilatory zwykle budują tak zwane wirtualne tablice metod.