

7. Насоки за програмиране

Лекционен курс “Програмиране на Java”
проф. д-р Станимир Стоянов

Структура на лекцията

- ▶ Конвенции за кода
- ▶ Избор на имена
- ▶ Лейаут

Насоки за програмиране: конвенции за кода, ръководства за стила

- Не са част от дефиницията на езиците: не се контролират от компилаторите
- Често вътрешни за фирмите
 - В частност различни
 - Въпреки това съществуват някои общоприети принципи

**Изисквания за външната форма на програмите:
повишаване на читаемостта**

Java-примерна програма: проблеми?

```
class S      {  
    int x, y; P p, q;  
    public void m()    {  
        switch(s)    {  
            case c1: d1();  
            break;  
            case c2:  
                d2();    break;  
            default: dA();  
            break;    }    }  
}
```

Java-примерна програма: какво прави впечатление?

```
class S{
    int x, y; P p, q;
    public void m()    {
        switch(s)    {
            case c1: d1();
            break;
            case c2:
            d2();    break;
            default: dA();
            break;    }    }
}
```

```
class Figure    {
    // only part
    int x, y;
    Point pLeft, pRight;

    public void drawFigure()    {
        switch(typFigure)    {
            case LINE:
                drawLine();
                break;

            case CIRCLE:
                drawCircle();
                break;

            default:
                drawAll();
                break;
        }
    }
}
```

Табулация и празни редове

```
class Figure {  
    int (x, y;  
        Point pLeft, pRight;  
        public void drawFigure() {  
            switch (typFigure) {  
                case LINE:  
                    drawLine();  
                    break;  
                case CIRCLE:  
                    drawCircle();  
                    break;  
                default:  
                    drawAll();  
                    break;  
            }  
        }  
    }  
}
```

The diagram illustrates the structure of the `Figure` class and its `drawFigure` method. It uses various annotations to highlight specific elements and control flow:

- A dashed line with an arrow points to the opening curly brace of the `Figure` class.
- A dashed line with an arrow points to the opening curly brace of the `drawFigure` method.
- A dashed line with an arrow points to the `switch` statement.
- A dashed line with an arrow points to the `drawLine()` call.
- A dashed line with an arrow points to the `break;` statement following `drawLine()`.
- A dashed line with an arrow points to the `break;` statement following `drawCircle()`.
- A dashed line with an arrow points to the `break;` statement in the `default` case.
- A dashed line with an arrow points to the closing curly brace of the `drawFigure` method.
- A dashed line with an arrow points to the closing curly brace of the `Figure` class.
- Horizontal arrows point from the right towards the closing curly braces of the `Figure` class, the `drawFigure` method, and the `switch` statement.
- The variable `x` in `int (x, y;` is circled.
- The variable `pLeft` in `Point pLeft, pRight;` is enclosed in a dashed box.
- The `LINE:` case label is enclosed in a box.
- The `break;` statement in the `default` case is circled.

Избор на имена: говорящи

Променливи

- Започват с малки букви
- Части от фразите започват с големи букви
- Кратки, но смислени
- Временни променливи (напр. броячи):
 - тип `int`: `i, j, k, m, n`
 - тип `char`: `c, d, ch`

`twoClients, actValue, color`

Константи

- С големи букви
- Части от фрази се разделят чрез `'_'`

`RED, GREEN, RED_GREEN`

Избор на имена (2)

Методи:

- Започват с малки букви
- Части от фрази започват с големи букви
- Желателно е да съдържат глаголи

`search, clientRecord`

Класове:

- Започват с големи букви
- Части от фрази започват с големи букви
- Желателно е да съдържат съществителни

`Tree, Client, GeometricFigure`

La u o u t: подреждане на програмата

Оформяне на текста и картините

Изрази:

Diagram illustrating the expression `help = a * b; i++;`. The expression is enclosed in a dashed box. Arrows point to the operators `=` and `*`, and the operands `help`, `a`, `b`, and `i++`.

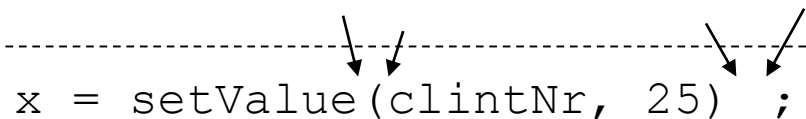
- Операциите да се разделят от операндите чрез празни символи, (изключение: унарни операции)

Diagram illustrating the expression `if (clientNr > NULL)`. The expression is enclosed in a dashed box. Arrows point to the operators `>` and `>`, and the operands `clientNr` and `NULL`.

- Празни символи вътре / извън скобите ?

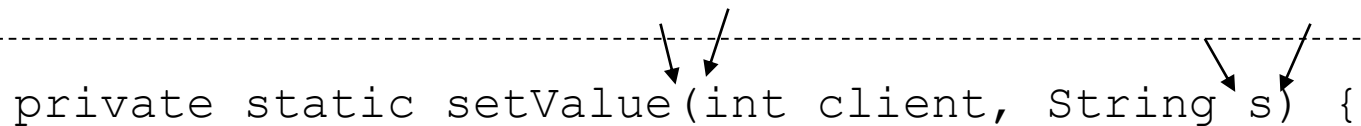
Layout (2)

Извикване и декларация на методи



```
x = setValue(clintNr, 25) ;
```

A diagram showing a method call `x = setValue(clintNr, 25) ;` enclosed in a dashed box. Four arrows point to the components of the call: two arrows point to `setValue` (one to the 'v' and one to the 'e'), one arrow points to `clintNr`, and one arrow points to the argument `25`.



```
private static setValue(int client, String s) {
```

A diagram showing a method declaration `private static setValue(int client, String s) {` enclosed in a dashed box. Four arrows point to the components of the declaration: two arrows point to `setValue` (one to the 'v' and one to the 'e'), one arrow points to the parameter `int`, and one arrow points to the opening curly brace `{`.

Layout (3)

Присвоявания, особено блокове: { ... }

C-Style

```
while (a > b) {  
    ...  
}
```

Pascal-Style

```
while (a > b)  
{  
    ...  
}
```

Layout (4)

If – оператор :

```
if (a > 0) {  
    ...  
} else if (a < 0) {  
    ...  
} else {  
    ...  
}
```

Layout (5)

```
class Shapes {  
    int x, y;  
    Point pFirst, pSecond;  
  
    public void selectColor () {  
        switch (selectedShape) {  
            case LINE:  
                drawLine();  
                break;  
        }  
    }  
}
```

Празни редове между
променливите и
методите

На ред най-много
един оператор

Layout (6)

Дълбочина на табулиране: 2, 3, 4

Дължина на редове: < 80 символа

По-дълги редове:

```
x = setValues(clientNr, accountNr,  
----->carNr, wheelNr);
```

След
запетаята

8 СИМВОЛА

```
nameV1 = nameV2 * (VAL1 + xxxx)  
* (VAL2 + yyyy);
```

Същата
колона

оператор: нов ред

Други

- Коментари: достатъчно и говорящи
- По една декларация на ред
- Методите да се отделят помежду си с празен ред
- В методите: локалните променливи и операторите да се разделят с празни редове
- В методите: различните логически части на един метод да се разделят с празни редове (напр. вход, I. подалгоритъм, ... изход)

Насоки за програмиране: референции

- **JAVA Coding Conventions from Oracle**

<http://www.oracle.com/technetwork/java/index-135089.html>

Благодаря за вниманието!

Край лекция 7. “Насоки за програмиране”