

PODSTAWY PROGRAMOWANIA W JAVA

dr inż. Michał Tomaszewski

katedra Metod Programowania
Polsko-Japońska Akademia Technik Komputerowych

INSTRUKCJA WARUNKOWA

Instrukcja warunkowa ma postać

Instrukcja warunkowa ma postać

`if (exp) Inst`

Instrukcja warunkowa ma postać

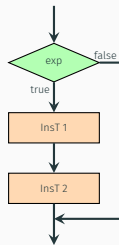
`if (exp) Inst`

w której **exp** jest wyrażeniem orzecznikowym, a **Inst** jest instrukcją.

Instrukcja warunkowa ma postać

`if (exp) InsT`

w której **exp** jest wyrażeniem orzecznikowym, a **InsT** jest instrukcją.



Instrukcja warunkowa ma postać

`if (exp) InsT`

albo

`if (exp) InsT else InsF`

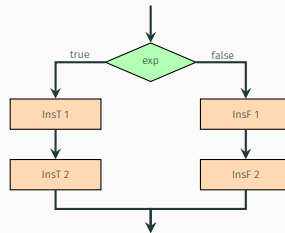
w której **exp** jest wyrażeniem orzecznikowym, a **InsT** oraz **InsF** jest instrukcją.

Instrukcja warunkowa ma postać

`if (exp) InsT`

albo

`if (exp) InsT else InsF`



w której **exp** jest wyrażeniem orzecznikowym, a **InsT** oraz **InsF** jest instrukcją.

Jeśli **InsT** jest instrukcją **warunkową**, to stosuje się zasadę, że każdej frazie **else** odpowiada najbliższa, poprzedzająca ją fraza **if**.

Jeśli **InsT** jest instrukcją **warunkową**, to stosuje się zasadę, że każdej frazie **else** odpowiada najbliższa, poprzedzająca ją fraza **if**.

W szczególności, poniższa instrukcja, w której **Ins1** ani **Ins2** nie są instrukcjami warunkowymi, zapis:

```
if (exp1) if (exp2) Ins1 else Ins2
```

INSTRUKCJA WARUNKOWA

Jeśli **InsT** jest instrukcją **warunkową**, to stosuje się zasadę, że każdej frazie **else** odpowiada najbliższa, poprzedzająca ją fraza **if**.

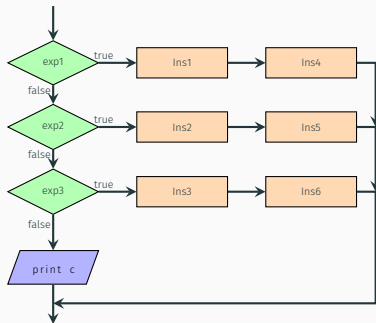
W szczególności, poniższa instrukcja, w której **Ins1** ani **Ins2** nie są instrukcjami warunkowymi, zapis:

```
if (exp1) if (exp2) Ins1 else Ins2
```

jest równoważny instrukcjom

```
if(exp1) {  
    if(exp2) {  
        Ins1  
    } else {  
        Ins2  
    }  
}
```

INSTRUKCJA IF/ELSE IF



INSTRUKCJA DECYZYJNA

Instrukcja decyzyjna ma postać

```
switch (exp0){  
    Case Case ... Case Default  
}
```

Instrukcja decyzyjna ma postać

```
switch (exp0){  
    Case Case ... Case Default  
}
```

w której każde **Case** jest frazą o postaci

```
case exp: Ins Ins ... Ins break;
```

a **Default** jest frazą w postaci

```
default: Ins Ins ... Ins break;
```

Instrukcja decyzyjna ma postać

```
switch (exp0){  
    Case Case ... Case Default  
}
```

w której każde **Case** jest frazą o postaci

```
case exp: Ins Ins ... Ins break;
```

a **Default** jest frazą w postaci

```
default: Ins Ins ... Ins break;
```

W takim zapisie, **exp0** jest wyrażeniem całkowitym, każde **exp** jest wyrażeniem stałym całkowitym, a każde **Ins** jest instrukcją albo jest napisem pustym.

W Java 5 wprowadzono typ wylicznikowy umożliwiając wybór stanu przez podanie stanu

W Java 5 wprowadzono typ wylicznikowy umożliwiając wybór stanu przez podanie stanu

```
public enum Day { SUNNY, CLOUDY }  
// ...  
Day day = Day.SUNNY;  
switch (day) {  
    case SUNNY:  
        break;  
    case CLOUDY:  
        break;  
    default:  
        System.out.println();  
}
```

W Java 7 wprowadzono zmienną typu `String` jako wyróżnik

W Java 7 wprowadzono zmienną typu `String` jako wyróżnik

```
String animal = "CAT";

switch (animal) {
    case "DOG":
        result = "domestic animal";
        break;
    case "CAT":
        result = "feline";
        break;
}
```

```
Day day = Day.SUNNY;  
switch (day) {  
    case SUNNY -> System.out.println(6);  
    case CLOUDY -> System.out.println(8);  
}
```

```
Day day = Day.SUNNY;
int result = switch (day) {
    case SUNNY:
        yield 6;
    case CLOUDY:
        yield 8;
    default:
        System.out.println("hmmm...");
        yield 0;
};
```

DZIĘKUJE