WYŻSZA SZKOŁA GOSPODARKI W BYDGOSZCZY

KIERUNEK : **MECHATRONIKA**

Przedmiot : PROGRAMOWANIE (STRUKTURALNE I OBIEKTOWE)

Prowadzący: INŻ. MARCIN JASIŃSKI

SPRAWOZDANIE Z ĆWICZEŃ

-LABORATORYJNYCH \*

Nr 3

1. **Rodzaje operatorów wykorzystywanych w Java oraz ich podział.**
2. **Instrukcja warunkowa if…else..**
3. **Instrukcja wyboru switch.**

Kolejny etap po zapoznaniu się z wiedzą na temat zmiennych dotyczy operacji wykonywanych na zmiennych za pomocą różnych operatorów, tj.

* arytmetyczne
  + dodawanie „ + „
  + odejmowanie „ - „
  + mnożenie „ \* „
  + dzielenie „ / :

Do działań arytmetycznych zaliczamy również :

* dzielenie % ( modulo ) czyli reszta z dzielenia
* operator inkrementacji ++ ( przyrost wartości o 1 ) - który w zależności od tego z, której strony zmiennej jest postawiony przyjmuje następujące wartości, np.

- ++x ( forma przedrostkowa ) –informuje nas, że wartość zmiennej zostanie zwiększona o 1 przed jej wykorzystaniem

- x++ (forma przyrostkowa ) – informuje, że wartość zmiennej x zostanie zwiększona o 1 po jej wykorzystaniu

* operator dekrementacji –( zmniejszenie wartości o 1 ) analogicznie jak poprzednio
* operatory bitowe – ( operacje te jak sama nazwa wskazuje są dokonywana na bitach, w systemach : dwójkowym, ósemkowym, dziesiętnym i szesnastkowym ), należą do nich:
* AND iloczyn bitowy – &
* OR suma bitowa - |
* NOT negacja – ~
* XOR alternatywa wykluczająca – ^
* przesunięcie bitowe w prawo – >>
* przesunięcie bitowe w lewo – <<
* przesunięcie bitowe w prawo z wypełnieniem zerami - >>>
* operatory logiczne
  + logiczne AND (&&)
  + Logiczne OR ( || )
  + Logiczne NOT ,negacja ( ! )

Argumenty operacji logicznych muszą posiadać wartość logiczną, czyli true lub false.

* operatory przypisania – powodują przypisanie wartości argumentu z prawej strony do argumentu z lewej strony

= , += , -= , \*= , /= , %= , <<= , >>= , &= , |= , ^= , np.

x \*= y znaczy x = x\*y

* operatory porównania ( relacyjne ) służą do porównywania argumentów, których wynikiem jest wartość logiczna true lub false

np. 1==2 false a 1==1 true

Rozróżniamy następujące operatory porównania:

== jeżeli argumenty są równe to true

!= jeżeli argumenty są różne to true

> jeżeli argument prawostronny jest mniejszy od lewostronnego to true

<

>=

<=

* operator warunkowy ma postać:

*warunek ? wartość1 : wartość2*

Wyrażenie przyjmuje wartość 1, gdy jest ono prawdziwe lub wartość 2 ,gdy jest inaczej.

Aby w pełni korzystać z operatorów musimy wiedzieć jaka jest kolejność ich wykonywania.

Najpierw wykonujemy:

- inkrementację przyrostkową

- inkrementację przedrostkową wraz z negacją

- mnożenie i dzielenie

- przesunięcia bitowe

- porównania

-bitowe AND

- bitowe XOR

- bitowe OR

- logiczne AND

- logiczne OR

- warunkowe

-przypisania

**Przejdźmy do kolejnego zagadnienia jakim jest instrukcja warunkowa**

**If….else** .

W pierwszej kolejności schemat instrukcji.

*If (wyrażenie warunkowe){*

*//instrukcja do wykonania gdy warunek jest prawdziwy*

*}*

*else {*

*//instrukcja do wykonania ,gdy warunek jest fałszywy*

*}*

Musimy zaznaczyć, że wynikiem wyrażenia warunkowego musi być wartość typu boolean, czyli true lub false, np.

*class Main {*

*public static void main (String args[]) {*

*int a = -10;*

*if (a > 0){*

*System.out.println ("Zmienna a jest większa od zera.");*

*}*

*else{*

*System.out.println ("Zmienna a nie jest większa od zera.");*

*}*

Rozumiemy przez to, że jeżeli warunek a > 0 jest spełniony to zostanie wykonana instrukcja z bloku if, w przeciwnym razie zostanie wykonana instrukcja bloku else.

***„Instrukcja switch„***

Schemat:

class Main {

public static void main (String args[]) {

int a = 10;

switch (a){

case 1 :

System.out.println("a jest równe 1.");

break;

case 10:

System.out.println("a jest równe 10.")

; break;

default:

System.out.println("a nie jest równe ani 1, ani 10.");

switch (wyrażenie){

case A:

// instrukcja1;

break;

case B:

// instrukcja2;

break;

case C:

// instrukcja3;

break;

default:

// instrukcja4;

}

W instrukcji tej sprawdzamy, czy wyrażenie zawarte w switch jest zgodne z którąś z wartości A,B,C. W sytuacji, gdy jedna z wartości spełnia wyrażenie wykonywana jest instrukcja zawarta w odpowiedniej klauzurze case.

**Wnioski**.

Po zapoznaniu się z zasadami korzystania z instrukcji „ jf…else” oraz instrukcji „switch” widzimy, że ich zaletą jest możliwość sprawdzenia w bardzo czytelny sposób jakiegoś warunku i w zależności od tego czy jest on prawdziwy, czy fałszywy możliwość wykonania różnego rodzaju instrukcje.

OCENA:……………

WYKONAŁ : SPRAWDZIŁ:

…Mariusz………………… ………………………..

Maksoń

/Imię i nazwisko nr albumu/

\*niepotrzebne skreślić