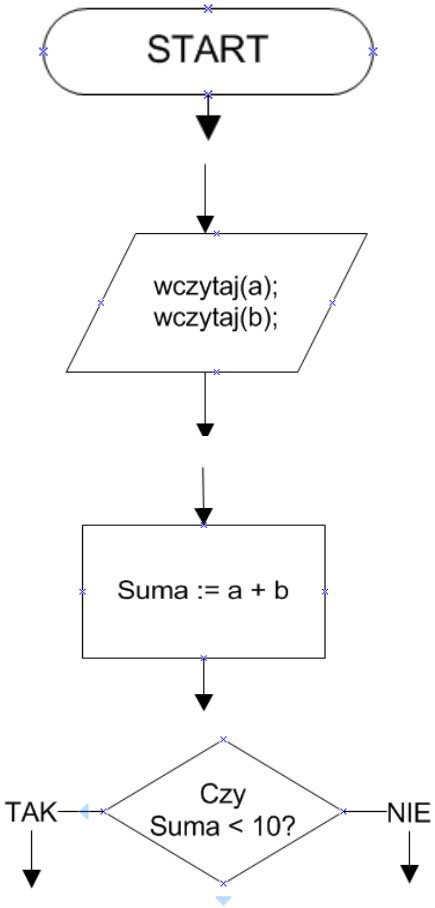
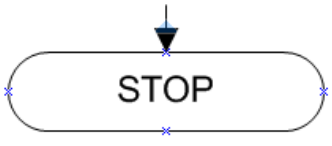
Podstawy programowania. Języki programowania.

Powtórka algorytmów.

Wprowadzenie do języka Java.

Przypomnienie schematów blokowych






```
BEGIN

Readln(a);
Readln(b);

Suma := 0;

IF Suma < 10 THEN
    Writeln(„Suma wynosi: ”, Suma);
ELSE
    Writeln(„Suma większa niż 10!”)

END.
```

Ćwiczenia

1. Narysować schemat blokowy i zapisać w pseudokodzie program wczytujący liczby aż do momentu pojawienia się zera. Po wczytaniu wszystkich liczb, program podaje największą i najmniejszą wpisaną liczbę (nie licząc 0).
2. Narysować schemat blokowy i zapisać w pseudokodzie program wczytujący pojedyncze znaki (do momentu podania znaku odpowiadającego literze ‘z’) i wyświetlające je użytkownikowi. Po wpisaniu znaku ‘e’ następny wprowadzony znak nie zostaje wyświetlony.
3. Narysować schemat blokowy i zapisać w pseudokodzie program wczytujący liczby (do momentu pojawienia się 0) oraz wyświetlający ich średnią arytmetyczną oraz sumę.

Ćwiczenia c.d.

4. Narysować schemat blokowy i zapisać w pseudokodzie program wczytujący liczby do momentu pojawienia się zera. Jeżeli liczba jest parzysta, należy jej wartość zwiększyć o 3. W przeciwnym przypadku wczytana liczba pozostaje bez zmian. Na końcu program wyświetla wprowadzone liczby po odpowiednich zmianach.
5. Narysować schemat blokowy i zapisać w pseudokodzie program wyświetlający silnię podanej przez użytkownika liczby.
6. Narysować schemat blokowy i zapisać w pseudokodzie program, który wypisuje kolejne potęgi liczby 2, dla których wykładnik jest liczbą parzystą, tj. $2^2, 2^4, 2^6, \ldots, 2^n$. Wejście: n.

Pierwszy program

```
package witajswiecie;

public class Main {

    public static void main(String[] args) {
        System.out.println("Witaj świecie Javy!");
    }
}
```

Pierwszy program – uwagii

- Java rozróżnia wielkość liter – println, Println, PRINTLN to są 3 różne nazwy!
- Główny plik źródłowy zawierający kod programu (z rozszerzeniem .java) musi nazywać się tak samo jak nazwa klasy.
- Wszystko w Javie musi się znajdować w jakieś klasie!
- Nazwa klasy musi się zaczynać od litery, po której mogą wystąpić dowolne kombinacje liter i cyfr. Nazwa nie może być zarezerwowanym słowem kluczowym.
- Funkcja (metoda main) musi się znajdować w pliku źródłowym klasy, która chcemy uruchomić.

Wstawianie komentarzy

Typy komentarzy:

- Jednolinijkowe (liniowe)

  // to jest komentarz składający się z 1 linii

- Wielolinijkowe (blokowe)

```
/* tutaj
   komentarz
   mający wiele
   linii */
```

Komentarzy blokowych NIE wolno zagnieżdżać /* /* błąd */ */

- Automatyczne generowanie dokumentacji

```
/** Jest to nasz pierwszy program
    @version 1.01
    @author Tomasz Xięski
 */
```

Ćwiczenia

1. Napisz program wyświetlający na ekranie napis: „To mój drugi program. Jestem programistą Javy!”. Skompiluj i uruchom.
2. Zapisz kod z poprzedniego zadania w pliku o innej nazwie niż Main.Java np. program.java. Zobacz co się stanie.
3. Zamień słowo void na Void. Skompiluj program i zobacz co się stanie.
4. Dołącz komentarz blokowy na początku swojego programu opisujący jego działanie. Sprawdź czy kod po zmianach się skompiluje.

Wbudowane typy danych

Java jest językiem implementującym tzw. **ścisłą kontrolę typów**. Oznacza to, że każda zmienna musi mieć ściśle określony typ. Typ jest przypisany zmiennej, wyrażeniu, lub innemu bytowi programistycznemu (metodzie, parametrowi, obiektowi itp.). Określa on **rodzaj wartości, jaką może przybrać** ten **byt**.

Do wbudowanych typów danych zalicza się:

- Typ znakowy.
- Typ logiczny.
- Typy arytmetyczne.
  - Całkowitoliczbowe.
  - Zmiennopozycyjne .
- Typ obiektowy (referencyjny).

Typy arytmetyczne całkowitoliczbowe

Służą one do reprezentacji liczb całkowitych. Są ich cztery rodzaje:

Typ	Liczba bitów	Zakres	Domyślna wartość
byte	8	od -128 do 127	0
short	16	od -32768 do 32767	0
int	32	od -2147483648 do 2147483647	0
long	64	od -9223372036854775808 do 9223372036854775807	0L

Duże liczby całkowite muszą posiadać modyfikator l lub L (np. 400000000000000L).

Liczby można zapisywać w formacie ósemkowym dodając przedrostek 0 (np. 010), lub w formacie szesnastkowym dodając przedrostek 0x (np. 0x12).

Typy arytmetyczne zmiennopozycyjne

Typy zmiennopozycyjne występują w dwóch rodzajach różniących się zakresem możliwych do reprezentacji liczb:

Typ	Liczba bitów	Zakres	Domyślna wartość
float	32	od $-3{,}4\cdot 10^{38}$ do $3{,}4\cdot 10^{38}$ (ok. 6-7 znaczących cyfr dziesiętnych)	0.0f
double	64	od $-1{,}8\cdot 10^{308}$ do $1{,}8\cdot 10^{308}$ (ok. 15 znaczących cyfr dziesiętnych)	0.0d

Liczby typu float mają przyrostek f lub F(np. 2.54f), natomiast typu double mają przyrostek d lub D (np. 54.2d).

Liczby zmiennoprzecinkowe pozbawione przyrostka są domyślnie traktowane jako typ double.

Istnieją również 3 szczególne wartości reprezentujące liczby, których wartości wybiegają poza wyznaczony zakres błędu:

- **Dodania nieskończoność**
- **Ujemna nieskończoność**
- **NaN** – Nie jest liczbą (*ang. Not a Number*) np. operacja 0 / 0;

Typ znakowy (char)

Typ char służy do reprezentacji **pojedynczych** znaków. Mogą to być stałe znakowe opatrzone apostrofami np. ‘A’, ‘z’, ‘0’. Typ ten zajmuje 16 bitów.

Można również wyświetlać symbole zakodowane w Unicode (których wartości mieszczą się w zakresie \u0000 do \uFFFF). Przykładowo \u03C0 to grecka litera π.

Ponadto wyróżnia się symbole reprezentujące znaki specjalne jak znak nowej linii czy tabulacja:

- \n – nowa linia
- \t – tabulacja
- \b – backspace
- \r – powrót karetki
- \” – cudzysłów
- \’ – znak apostrofu
- \\ – lewy ukośnik (*ang. backslash*)

Typ logiczny (boolean)

- Typ boolean (logiczny) przechowuje jedną z dwóch wartości: **true** (prawda) bądź **false** (fałsz). Służą one sprawdzaniu warunków logicznych.
- **Do programistów C++:** Wartości logicznych nie można konwertować na całkowitoliczbowe.

Zmienne w Javie

Zmienna – konstrukcja programistyczna posiadająca swoją unikalną **nazwę**, oraz **typ danych** jakie zmienna jest w stanie przechowywać.

Każda zmienna przed wykorzystaniem w programie musi zostać zadeklarowana (stworzona). Deklaracja polega na podaniu typu i nazwy zmiennej w formie:

typ_zmiennej *nazwa_zmiennej;*

np. **int** suma;

Aby naraz zadeklarować wiele zmiennych tego samego typu oddzielamy ich nazwy przecinkiem:

np. **int** suma, pensja, obroty;

Na końcu każdej deklaracji musi znajdować się średnik!

Nazwa zmiennej może się składać z liter (małych lub wielkich), cyfr, oraz znaku podkreślenia, ale **nie może się zaczynać od cyfry**.

Zmienne w Javie c.d.

Aby przypisać zmiennej jakąś określoną wartość wpisujemy:

nazwa_zmiennej = wartość;

np. suma = 100;

Pierwsze przypisanie wartości zmiennej nazywa się jej inicjalizacją. Inicjalizacja zmiennej może się odbyć w dowolnym miejscu programu po jej wcześniejszym zadeklarowaniu. Może też wystąpić równocześnie z deklaracją:

int suma = 100;

Wyświetlanie wartości zmiennych

W celu wyświetlenia wartości zmiennych korzystamy ze znanej już instrukcji:

System.out.println(zmienna_do_wyswietlenia);

Przykładowo aby wyświetlić zawartość zmiennej suma:

```
int suma = 100;
System.out.println(suma);
```

Aby wyświetlić jakiś napis, podajemy go jako argument w/w instrukcji opatrzony w cudzysłów:

```
System.out.println("Jakiś napis do wyświetlenia");
```

Można połączyć wyświetlanie napisu i wartości zmiennych za pomocą operatora łączenia +:

```
System.out.println("Wartość zmiennej suma wynosi" + suma);
```

Wyświetlanie wartości zmiennych c.d.

- Aby wyświetlić zawartość dwóch zmiennych w jednej linii można posłużyć się instrukcją:

```
System.out.print(zmienna);
```

Przykładowo:

```
int zawodnik = 1;
double czas = 33;
System.out.print(" Zawodnik nr ");
System.out.print(zawodnik);
System.out.print(" miał czas ");
System.out.println(czas);
```

- LUB użyć operatora łączenia +

```
System.out.println(" Zawodnik nr " + zawodnik + " miał czas " + czas);
```

Wyświetlanie danych na ekranie

- Wszystkie sekwencje znaków specjalnych przedstawionych przy okazji omówienia typu char działają również na łańcuchach znaków.

- \n – nowa linia
- \t – tabulacja
- \b – backspace
- \r – powrót karetki
- \” – cudzysłów
- \’ – znak apostrofu
- \\ – lewy ukośnik (*ang. backslash*)

- Aby wyświetlić tekst w 3 liniach wykorzystujemy sekwencję \n:

System.out.println(„To\njest teskt\nw 3 liniach”);

Ćwiczenia

1. Zadeklaruj zmienną typu float i nadaj jej wartość 4.32. Wyświetl wartość zmiennej na ekranie.
2. Zadeklaruj dwie zmienne typu short i nadaj im przykładowe wartości. Następnie wyświetl zawartość obu zmiennych na ekranie w dwóch wierszach nie korzystając z instrukcji System.out.println().
3. Zadeklaruj zmienną o nazwie bilion typu long i nadaj jej wartość jednego biliona. Następnie korzystając z tej zmiennej wyświetl napis na ekranie: Na moim koncie jest 1000000000000 zł dlatego jestem bilionerem.

Ćwiczenia c.d.

4. Napisz program wyświetlający napis jak na obrazku poniżej:

```
|---  /---\  |     |   /---\
    |  |    |  |     |  |    |
    |  |    |  |     |  |    |
    |  |---|  |     |  |---|
    |  |    |   \   /   |    |
\--/  |    |    \/     |    |

          W I T A
```