SPOTKANIE 3 – grupa podstawowa

1. Pętla while
2. Pętla do-while
3. Pętla for
4. Zadania

1. **Pętla while**

   Po co nam pętle?

   Pętle w programowaniu są wykorzystywane niemal w każdym programie. Jest to jedna z podstawowych konstrukcji wykorzystywana we wszystkich językach programowania. Pozwalają one wykonywać określoną czynność cyklicznie w przypadku, gdy musimy na przykład wyświetlić 10 razy prawie to samo.

   Pętlę while najczęściej wykorzystuje się w miejscach, gdzie zakładana ilość powtórzeń jest bliżej nieokreślona, ale znamy warunek jaki musi być spełniony. Jej schematyczną postać przedstawiono poniżej:

```
while(warunek)
{
instrukcje do wykonania
}
```

   Gdy warunek jest spełniony instrukcja we wnętrzu pętli jest wykonywana. Jeżeli natomiast warunek jest fałszywy może się ona nie wykonać, ani razu.

```
public class Odczyt
{
	public static void main(String[] args)
	{
		int licznik = 0;

		while(licznik<10)
		{
			System.out.println("To jest petla");
			licznik++;
		}
	System.out.println("Koniec pętli");
	}
}
```

2. **Pętla do-while**

Pętla do- while Różni się ona od pętli while przede wszystkim tym, że to co znajduje się w jej wnętrzu wykona się przynajmniej raz, ponieważ warunek jest sprawdzany dopiero w drugiej kolejności.

```
do
{
instrukcje do wykonania
}
while(warunek);
```

Instrukcje do wykonania są egzekwowane przynajmniej raz, a dopiero później jest sprawdzany warunek. Kiedy jest on prawdziwy, instrukcje są powtarzane. Tę pętlę można zapamiętać jako „wykonywanie instrukcji dopóki warunek jest prawdziwy". Wcześniejszy przykład z użyciem pętli do-while wyglądałby tak:

```
public class Odczyt
{
	public static void main(String[] args)
	{
		int licznik = 0;
		do
		{
			System.out.println("To jest petla");
			licznik++;
		}
		while(licznik<10);

		System.out.println("Koniec pętli");
	}
}
```

Przebieg działania jest prawie taki sam, tylko, że gdyby przykładowo licznik był zainicjowany liczbą 20, to mimo wszystko wyświetliłby się raz napis „To jest pętla".

3. **Pętla for**

Pętlę for od dwóch poprzednich odróżnia przede wszystkim rodzaj zastosowań. W jej przypadku zazwyczaj możemy dokładnie założyć, ile razy ma się jakaś czynność powtórzyć. Jej schemat to:

```
for(wyrażenie początkowe ; warunek ; modyfikator_licznika)
{
      instrukcje do wykonania
}
```

Wyrażenie początkowe służy do zainicjowania jakiegoś licznika, którym zazwyczaj jest w tym samym miejscu zadeklarowana zmienna typu całkowitego - najczęściej oznacza się je przy pomocy liter od "i" wzwyż, jest to przydatne przy zagnieżdżonych pętlach, gdzie przy długich nazwach ciężko by się było połapać. Warunek sprawdza, czy ma się wykonać instrukcja z wnętrza pętli, natomiast modyfikator zmienia licznik. Wcześniejszy przykład z wykorzystaniem pętli for wyglądałby więc tak:

```
public class Odczyt
{
      public static void main(String[] args)
      {
            for(int i=0; i<10; i++)
            {
                  System.out.println("To jest pętla");
            }
            System.out.println("Koniec pętli");
      }
}
```

Informacyjnie:
z matematyki przyjęło się, że liczniki nazywają się
i (dla pętli pojedynczej lub nadrzędnej),
j (dla pętli wewnętrznej, czyli umieszczonej w innej pętli).

Na rozgrzewkę. Ile razy wykonają się poniższe pętle?

a) for(int i=1; i<=3; i++)

b) for(int i=-2; i<=2; i++)

c) for(int i=98; i<=103; i++)

d) for(int i=0; i<10; i++)

e) for(char z='a'; z<'e'; z++)

f) for(int i=100; i>95; i--)

g) for(int i=5; i>0; i++)

h) for(int i=0; i<10; i+=3)

4. **Zadania**

Zadanie 1 / do; do-while

Napisz program, który ma wczytywać liczby dopóki użytkownik nie poda wartości 100.

Dla ułatwienia: ustal na początku warunek końca i warunek kontynuacji pętli.

Zadanie 2 / do; do-while

Napisz program, który meczy użytkownika tak długo, aż ten poda prawidłowe hasło, a gdy to zrobi wyświetlana jest duża tajemnica ☺

Zadanie 3 / do; do-while

Napisz program, którego zadaniem jest wczytywanie liczb dopóki użytkownik nie poda przynajmniej trzech liczb, w tym przynajmniej jednej liczby parzystej.

Zadanie 4 / for

Napisz program przypominający działanie bankomatu, który wczytuje PIN karty. Dla przypomnienia karta jest pożerana po trzech nieudanych próbach. }

Zadanie 5 / for

Napisz program, który wczyta 10 liczb, a następnie wyświetli, ile wśród nich było liczb parzystych.

Zadanie 6 / for

Napisz program, który po wczytaniu 5 liczb wyświetla jaka była najmniejsza liczba wczytana od użytkownika.

Zadanie 7

Napisz program, który wczytuje liczby, dopóki nie wczyta wartości zero, a następnie wyświetla, ile było wśród wczytanych liczb (poza zerem, które funkcjonuje jako znacznik końca wczytywania danych):

a) liczb dodatnich,
b) liczb ujemnych,
c) liczb trzycyfrowych,
d) liczb parzystych.

nie licząc wartości zero, która jest traktowana **jako znacznik końca**.

Zadanie 8

Napisz program, który wczytuje liczby, dopóki nie wczyta wartości 100, a następnie wyświetla:

a) średnią wczytanych liczb,
b) maksymalną liczbę wśród tych, które podał użytkownik,

c) minimalną liczbę wśród tych, które podał użytkownik.

nie licząc wartości 100, która jest traktowana jako **znacznik końca**.