

Informatyka 1 (EZ1D200 008)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr III, studia niestacjonarne II stopnia
(grupa dodatkowa, uzupełnienie efektów kształcenia)
Rok akademicki 2019/2020

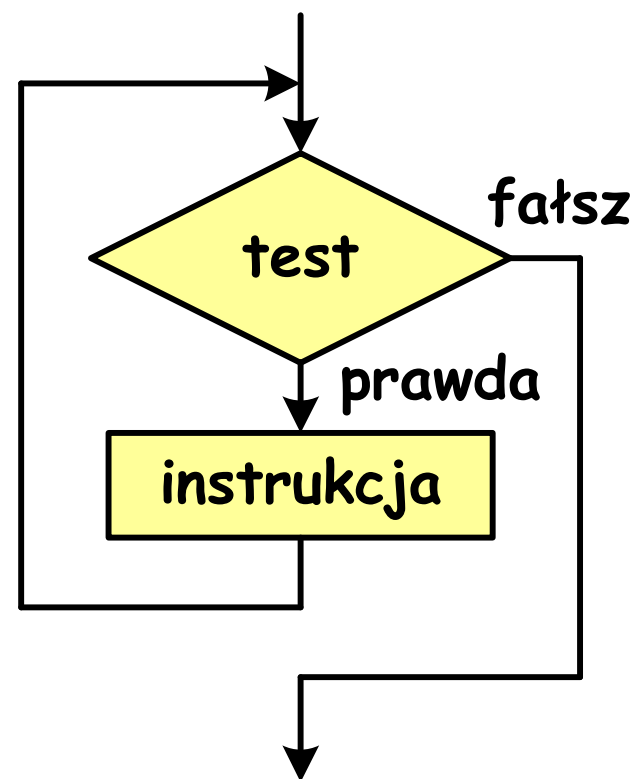
Pracownia nr 6

dr inż. Jarosław Forenc

Pętla while

```
while (test)  
instrukcja
```

- „dopóki **test** prawdziwy wykonuj **instrukcję**”
- **test** - dowolne wyrażenie (najczęściej wyrażenie logiczne)



Pętla while

```
while (test)  
instrukcja
```

- instrukcja prosta

```
int i=0;  
while (i<10)  
    i++;
```

- instrukcja złożona

```
int i=0;  
while (i<10)  
{  
    printf("%d ",i);  
    i++;  
}
```

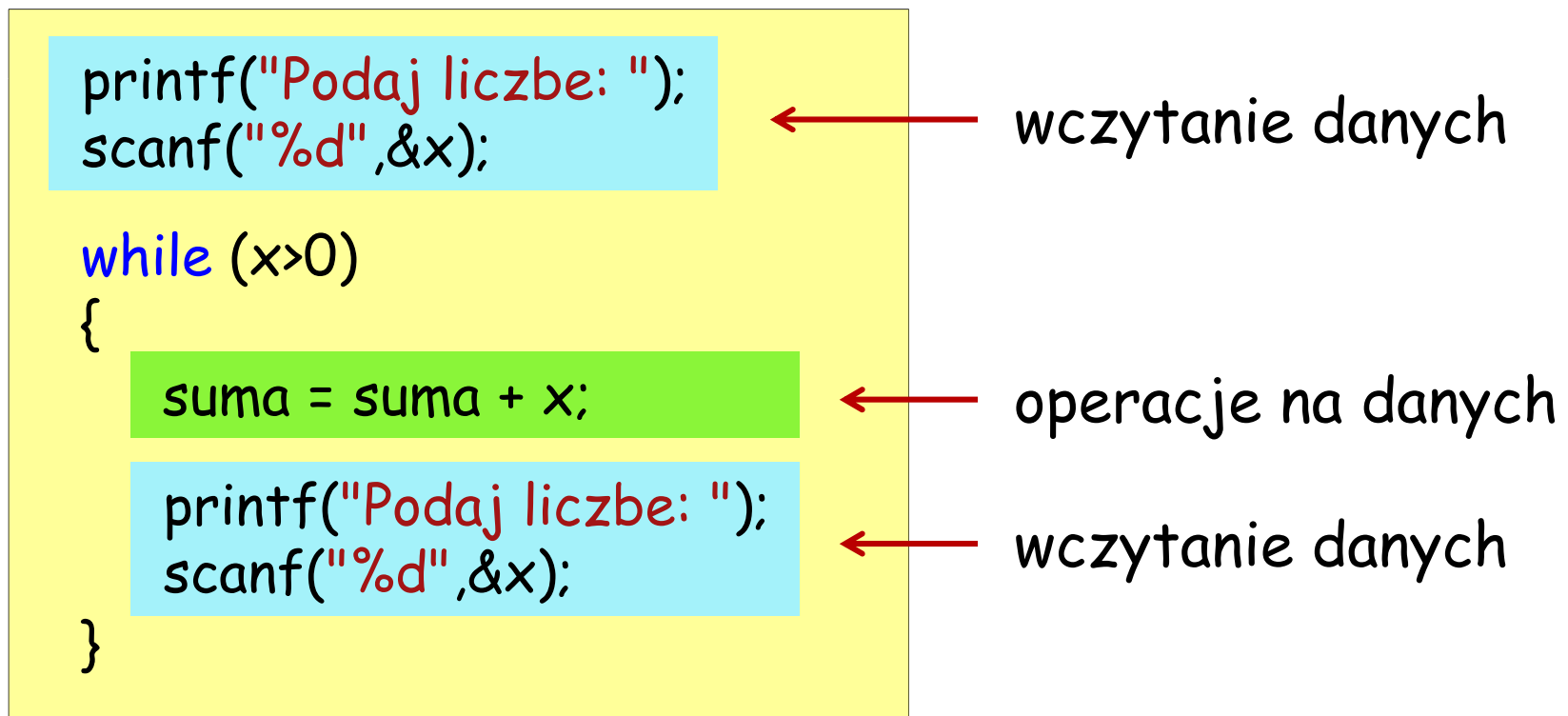
Przykład: suma liczb > 0

```
#include <stdio.h>
int main(void)
{
    int x, suma = 0;
    printf("Podaj liczbe: "); scanf("%d",&x);
    while (x>0)
    {
        suma = suma + x;
        printf("Podaj liczbe: "); scanf("%d",&x);
    }
    printf("Suma liczb: %d\n",suma);
    return 0;
}
```

```
Podaj liczbe: 4
Podaj liczbe: 8
Podaj liczbe: 2
Podaj liczbe: 3
Podaj liczbe: 5
Podaj liczbe: -2
Suma liczb: 22
```

Pętla while

- typowy schemat przetwarzania danych z wykorzystaniem pętli **while**



Przykład: $1 + 2 + \dots + ? < 100$

```
#include <stdio.h>

int main(void)
{
    int suma = 0, n = 0;

    while (suma < 100)
    {
        n++;
        suma = suma + n;
    }
    suma = suma - n;
    n--;
    printf("Ilosc liczb: %d\nSuma liczb: %d\n",n,suma);
    return 0;
}
```

Ilosc liczb: 13
Suma liczb: 91

Przykład: skrócenie zapisu

```
while (suma < 100)
{
    n++;
    suma = suma + n;
}
```

lub

```
while (suma < 100)
    suma = suma + ++n;
```

lub

```
while ((suma = suma + ++n) < 100);
```

lub

```
while ((suma += ++n) < 100);
```

Pętla while - błędy

- średnik na końcu **while** - pętla nieskończona:

```
i = 0;  
while (i < 10) ;  
    printf("%d ", i++);
```



- brak aktualizacji „i” - pętla nieskończona:

```
i = 0;  
while (i < 10)  
    printf("%d ", i);
```

0 0 0 0 0 0 0 0 ...

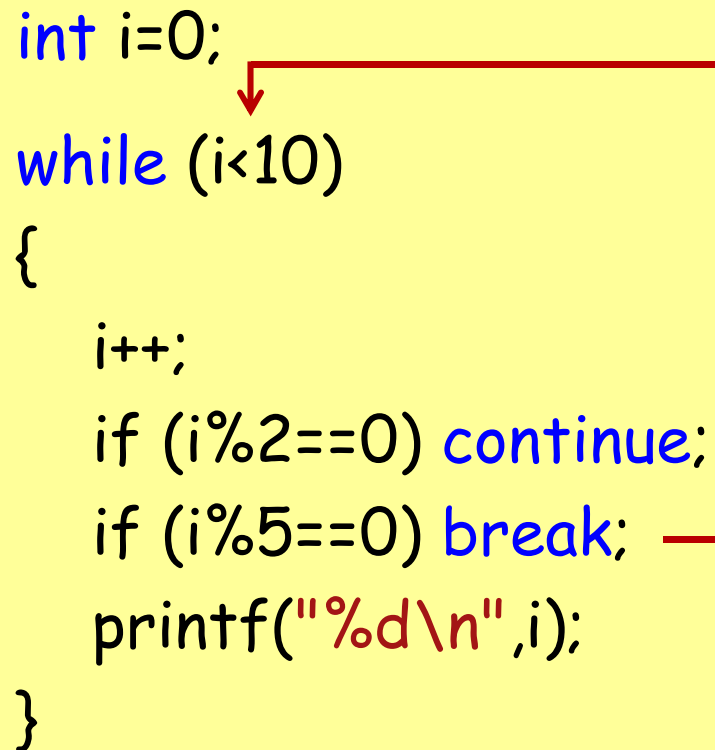
Pętla while - pętla nieskończona

- w pewnych sytuacjach celowo stosuje się pętlę nieskończoną:

```
while (1)
{
    instrukcja1;
    instrukcja2;
    ...
}
```

Pętla while - break i continue

```
int i=0;
while (i<10)
{
    i++;
    if (i%2==0) continue;
    if (i%5==0) break;
    printf("%d\n",i);
}
```



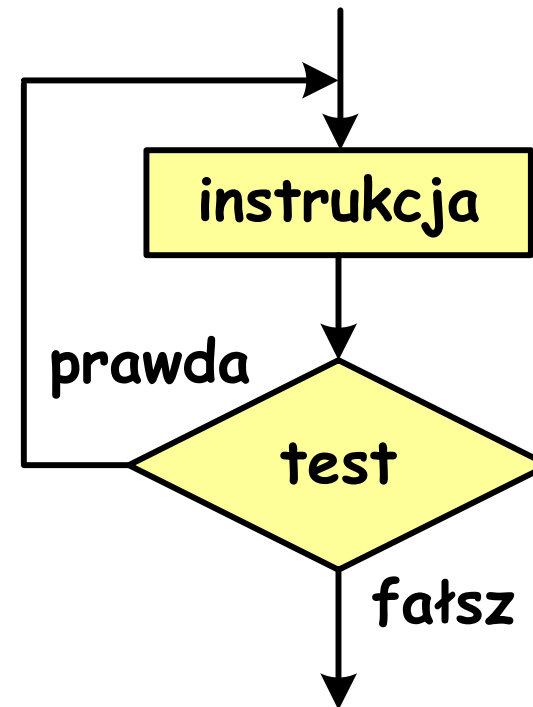
- **continue** przerywa bieżącą iterację

- **break** przerywa wykonywanie pętli

Pętla do...while

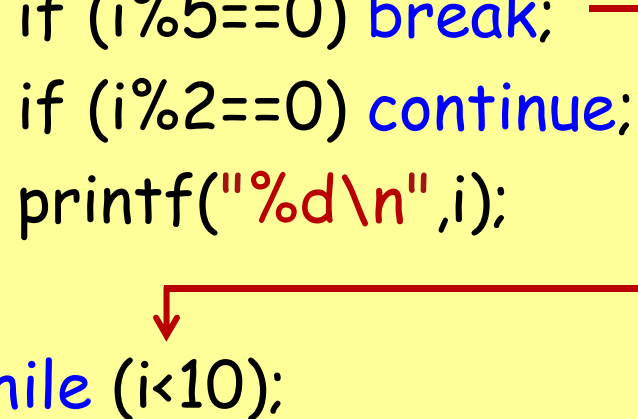
```
do  
  instrukcja  
while (test);
```

- „wykonuj **instrukcję** dopóki **test** prawdziwy”



Pętla do...while

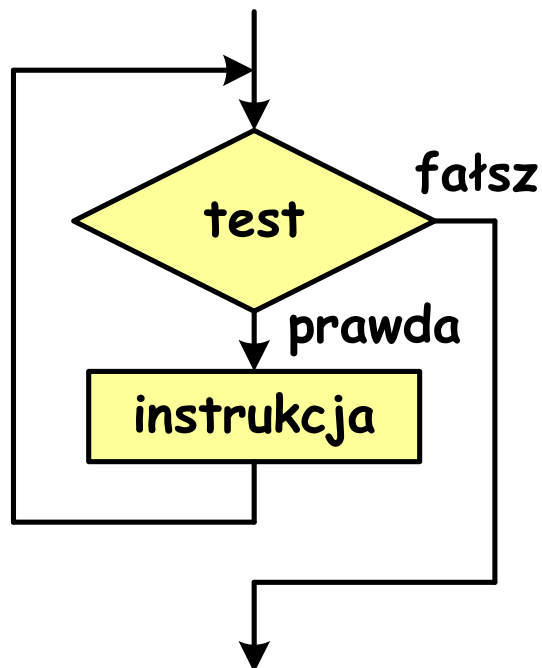
```
int i=0;
do
{
    i++;
    if (i%5==0) break;
    if (i%2==0) continue;
    printf("%d\n",i);
}
while (i<10);
```



- instrukcja:
prosta lub złożona
- **break** przerywa wykonywanie pętli
- **continue** przerywa bieżącą iterację

Porównanie pętli while i do...while

`while (test)`
instrukcja



`do`
instrukcja
`while (test);`

