

# Лаб: Повторения с цикли – while-цикъл

Задачи за упражнение и домашно към курса "[Основи на програмирането](#)" в СофтУни.

Тествайте решенията си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1014>

## 1. Число в диапазона [1...100]

Напишете програма, която въвежда цяло положително **число** **n** в **диапазона [1...100]**. При въвеждане на число извън посочения диапазон, да се отпечата съобщение за грешка и потребителят да се подкани **да въведе ново число**.

### Примерен вход и изход

Вход / Изход
35 The number is: 35
105 Invalid number!
0 Invalid number!
-200 Invalid number!
77 The number is: 77

### Насоки

1. Инициализирайте променлива **num**, в която да пазите стойността на текущото число:

```
Scanner scanner = new Scanner(System.in);  
int num = Integer.parseInt(scanner.nextLine());
```

2. В **while** цикъл, до получаване на валидно число, четете ново:

```
while (num < 1 || num > 100) {  
    System.out.println("Invalid number!");  
    num = Integer.parseInt(scanner.nextLine());  
}
```

3. Когато намерите **валидно число** го **принтирайте**:

```
Scanner scanner = new Scanner(System.in);  
int num = Integer.parseInt(scanner.nextLine());  
  
while (num < 1 || num > 100) {  
    System.out.println("Invalid number!");  
    num = Integer.parseInt(scanner.nextLine());  
}  
  
System.out.println(num);
```

## 2. Парола

Напишете програма, която първоначално прочита име и парола на потребителски профил. След това чете парола за вход, при въвеждане на грешна парола, потребителя да се подкани да въведе нова парола.

### Примерен вход и изход

вход	изход
Nakov 1234 pass 1324 1234	Welcome Nakov!

вход	изход
Gosho secret secret	Welcome Gosho!

### Насоки

1. Инициализирайте две променливи **username** и **password**, които ще съдържат потребителското име и паролата:

```
Scanner scan = new Scanner(System.in);  
String username = scan.nextLine();  
String password = scan.nextLine();
```

2. Инициализирайте променлива **input**, която ще държи въведената от потребителя парола за вход:

```
String input = scan.nextLine();
```

3. В **while** цикъл, до въвеждане на валидна парола, четете нова:

```
while (!input.equals(password)) {  
    input = scan.nextLine();  
}
```

4. Когато се въведе валидна парола принтирайте съобщението за успешен вход:

```
Scanner scan = new Scanner(System.in);  
String username = scan.nextLine();  
String password = scan.nextLine();  
  
String input = scan.nextLine();  
  
while (!input.equals(password)) {  
    input = scan.nextLine();  
}  
  
System.out.printf("Welcome %s!", username);
```

### 3. Редица числа $2k+1$

Напишете програма, която чете число  $n$ , въведено от потребителя, и отпечатва **всички** числа  $\leq n$  от редицата: 1, 3, 7, 15, 31, .... Всяко следващо число се изчислява като умножим **предишното** с 2 и добавим 1.

#### Примерен вход и изход

вход	изход	вход	изход	вход	изход	вход	изход
3	1 3	8	1 3 7	17	1 3 7 15	31	1 3 7 15 31

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1014#7>

#### Насоки

1. Прочетете  $n$  – цяло число, което представлява края на редицата

```
Scanner scanner = new Scanner(System.in);  
int n = Integer.parseInt(scanner.nextLine());
```

2. В **while** цикъл до достигане на  $n$ , пресмятайте ново число от редицата и го принтирайте:

```
int number = 1;  
while (number <= n) {  
    System.out.println(number);  
    number = number * 2 + 1;  
}
```

### 4. Баланс по сметка

Напишете програма, която пресмята **колко общо пари** има в сметката, след като направите **определен брой вноски**. На първия ред ще получите **колко вноски трябва да се направят**. На всеки следващ ред ще получавате сумата, която трябва да внесете в сметката, **докато не се достигне броя вноски**. При всяка получена сума на конзолата трябва да се извежда **"Increase: "** + сумата и тя да се **прибавя в сметката**. Ако получите число **по-малко от 0** на конзолата трябва да се изведе **"Invalid operation!"** и програмата **да приключи**. Когато програмата приключи трябва да се принтира **"Total: "** + общата сума в сметката закръглена до втория знак след десетичната запетая.

#### Примерен вход и изход

вход	изход	вход	изход
3	Increase: 5.51	5	Increase: 120
5.51	Increase: 69.42	120	Increase: 45.55
69.42	Increase: 100	45.55	Invalid operation!
100	Total: 174.93	-150	Total: 165.55

#### Насоки

1. Прочетете **броя на вноските (транзакциите)**, които ще бъдат направени:

```
Scanner scanner = new Scanner(System.in);
int countOfTransactions = Integer.parseInt(scanner.nextLine());
```

2. В **while** цикъл, докато броя на **направените транзакции** е по-малък от тези, които трябва да се направят, четете сума на текущата транзакция и извършвайте пресмятания и проверки с нея:

```
int transactionsMade = 0;
while (transactionsMade < countOfTransactions){
    // Read transaction sum and update balance
    transactionsMade++;
}
```

3. Четете сумата на текущата транзакция. Валидирайте я и принтирайте подходящо съобщение, ако тя е по-малка от 0:

```
double amount = Double.parseDouble(scanner.nextLine());
if (amount < 0){
    System.out.println("Invalid operation!");
    break;
}
```

4. Ако сумата на транзакцията е положителна, принтирайте съобщение за промяна в баланса на сметката и увеличете нейната стойност. Увеличете броя на направените транзакции:

```
System.out.printf("Increase: %.2f\n", amount);
sum += amount;
transactionsMade++;
```

5. Когато достигнете броя на нужните транзакции, принтирайте **крайния баланс на сметката**:

```
Scanner scanner = new Scanner(System.in);
int countOfTransactions = Integer.parseInt(scanner.nextLine());

double sum = 0;
int transactionsMade = 0;
while (transactionsMade < countOfTransactions){
    double amount = Double.parseDouble(scanner.nextLine());
    if (amount < 0){
        System.out.println("Invalid operation!");
        break;
    }

    System.out.printf("Increase: %.2f\n", amount);
    sum += amount;
    transactionsMade++;
}

System.out.printf("Total balance: %.2f", sum);
```

## 5. Най-голямо число

Напишете програма, която чете **n-на брой цели числа** ( $n > 0$ ), въведени от потребителя, и намира **най-голямото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред.

## Примерен вход и изход

вход	изход
2 100 99	100

вход	изход
3 -10 20 -30	20

вход	изход
4 45 -20 7 99	99

вход	изход
1 999	999

вход	изход
2 -1 -2	-1

## 6. Най-малко число

Напишете програма, която чете **n-на брой цели числа** ( $n > 0$ ), въведени от потребителя и намира **най-малкото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред.

### Примерен вход и изход

вход	изход
2 100 99	99

вход	изход
3 -10 20 -30	-30

вход	изход
4 45 -20 7 99	-20

вход	изход
1 999	999

вход	изход
2 -1 -2	-2

## 7. Завършване

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. Ако оценката му е под 4.00, той ще повтори класа.

При успешно завършване на **12-ти** клас да се отпечата:

"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

Стойността трябва да бъде форматирана до втория знак след десетичната запетая.

### Примерен вход и изход

вход	изход	вход	изход	вход	изход
Pesho 4 5.5 6 5.43 4.5 6 5.55 5 6 6 5.43 5	Pesho graduated. Average grade: 5.37	Pesho 5 5 5 6 5.5 5 6 5.44 5 5 5 5 6 5.45	Pesho graduated. Average grade: 5.25	Ani 5 5.32 6 5.43 5 6 5.5 4.55 5 6 5.56 6	Ani graduated. Average grade: 5.45

## 8. Завършване - част 2

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. Ако ученикът бъде скъсан повече от един път, то той бива изключен и програмата приключва, като се отпечата **името на ученика и в кой клас бива изключен**.

При успешно завършване на **12-ти** клас да се отпечата :

"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

В случай, че ученикът е изключен от училище, да се отпечата:

"{име на ученика} has been excluded at {класа, в който е бил изключен} grade"

Стойността трябва да бъде форматирана до втория знак след десетичната запетая.

### Примерен вход и изход

вход	изход	вход	изход
Gosho	Gosho graduated. Average grade: 5.53	Mimi	Mimi has been excluded at 8 grade
5		5	
5.5		6	
6		5	
5.43		6	
5.5		5	
6		6	
5.55		6	
5		2	
6		3	
6			
5.43			
5			

## Примерна изпитна задача

### 9. \*Преместване

На осемнадесетия си рожден ден на Хосе взел решение, че ще се изнесе да живее на квартира. Опаковал багажа си в **кашони** и намерил подходяща обява за апартамент под наем. Той започва да пренася своя багаж **на части**, защото не може да пренесе целия наведнъж. Има ограничено **свободно пространство** в новото си жилище, където може да разположи вещите, така че мястото да бъде подходящо за живеене.

Напишете **програма**, която изчислява **свободния обем** от жилището на Хосе, който остава след като **пренесе багажа си**.

Бележка: Един кашон е с точни размери: **1m. x 1m. x 1m.**

### Вход

Потребителят въвежда следните данни на отделни редове:

1. Широчина на свободното пространство - цяло число в интервала [1...1000]
2. Дължина на свободното пространство - цяло число в интервала [1...1000]

3. Височина на свободното пространство - цяло число в интервала [1...1000]

4. На следващите редове (до получаване на команда "Done") - брой кашони, които се пренасят в квартирата - цели числа в интервала [1...10000]

Програмата трябва да приключи прочитането на данни при команда "Done" или ако свободното място свърши.

## Изход

Да се отпечата на конзолата **един** от следните редове:

- Ако стигнете до командата "Done" и има още свободно място:  
"{брой свободни куб. метри} Cubic meters left."
- Ако свободното място свърши преди да е дошла команда "Done":  
"No more free space! You need {брой недостигащи куб. метри} Cubic meters more."

## Примерен вход и изход

Вход	Изход	Обяснение
10 10 2 20 20 20 20 122	No more free space! You need 2 Cubic meters more.	$10 * 10 * 2 = 200$ кубични метра. $20 + 20 + 20 + 20 + 122 = 202$ кубични метра. $200 - 202 = 2$ недостигащи кубични метра
10 1 2 4 6 Done	10 Cubic meters left.	$10 * 1 * 2 = 20$ кубични метра. $4 + 6 = 10$ кубични метра. $20 - 10 = 10$ кубични метра.