



Преговор

1. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 3;) {  
    System.out.print(i);  
}
```

123

Infinite loop

111

Compile time
error

1. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 3;) {  
    System.out.print(i);  
}
```

123

Infinite loop

111

Compile time
error

2. Какъв ще е резултатът от изпълнението на следния код:

```
for(; ;) {  
    System.out.print("SoftUni");  
}
```

Infinite loop

Runtime error

Compile
time error

SoftUni

2. Какъв ще е резултатът от изпълнението на следния код:

```
for(; ;) {  
    System.out.print("SoftUni");  
}
```

Infinite loop

Runtime error

Compile
time error

SoftUni

3. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 97; i < 100; i++) {  
    System.out.print((char) i + " ");  
}
```

a b c

Compile time
error

97 98 99

Infinite loop

3. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 97; i < 100; i++) {  
    System.out.print((char) i + " ");  
}
```

a b c

Compile time
error

97 98 99

Infinite loop

4. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 0; i < 'A'; i++) {  
    System.out.print(i + " ");  
}
```

ASCII symbols:
0 ... 'A'

Infinite loop

0... 64

Compile time
error

4. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 0; i < 'A'; i++) {  
    System.out.print(i + " ");  
}
```

ASCII symbols:
0 ... 'A'

Infinite loop

0... 64

Compile time
error

5. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 0; i < 2; i += 0.5) {  
    System.out.print(i + ", ");  
}
```

0, 0.5, 1, 1.5

0, 1

Compile time
error

Infinite loop

5. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 0; i < 2; i += 0.5) {  
    System.out.print(i + ", ");  
}
```

0, 0.5, 1, 1.5

0, 1

Infinite loop

Compile time
error

6. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 5; i++) {  
    if (i == 2 || i == 3) {  
        continue;  
    }  
    System.out.print(i + " ");  
}
```

1 2 3 4 5

1 4 5

1 2 3 4

Compile time
error

6. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 5; i++) {  
    if (i == 2 || i == 3) {  
        continue;  
    }  
    System.out.print(i + " ");  
}
```

1 2 3 4 5

1 4 5

1 2 3 4

Compile time
error

7. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
    if (i % 10 == 5) {  
        break;  
    }  
}
```

1 2 3 4 5

1 2 3 4

Compile
time error

5 10

7. Какъв ще е резултатът от изпълнението на следния код:

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
    if (i % 10 == 5) {  
        break;  
    }  
}
```

1 2 3 4 5

1 2 3 4

Compile
time error

5 10

Работа с вложени цикли

По-сложни задачи



СофтУни

Преподавателски екип



SoftUni
Foundation



Софтуерен университет

<http://softuni.bg>

sli.do

#pb-oct



Вложени цикли

По-сложни комбинаторни задачи

Пример – часовник (1)

Часовете се променят
когато минутите
надвишат 59

19:03

Докато минутите се
променят часовете
остават същите



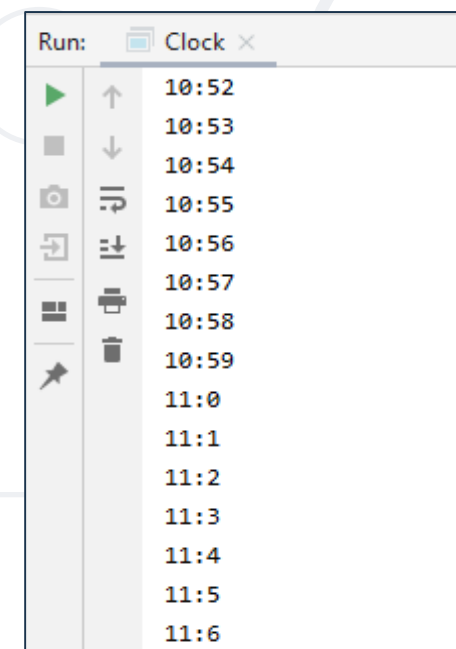
Как може да си направим часовник с код?

Демо

Пример – часовник (2)

- Външният цикъл отговаря за часовете, а вътрешния за минутите

```
for (int h = 0; h <= 23; h++) {  
    for (int m = 0; m <= 59; m++) {  
        System.out.printf("%d:%d%n",  
            h, m);  
    }  
}
```



```
for (int i = 0; i < n; i++)  
    for (int j = 0; j < n; j++)  
        ...
```

Имената на
итераторите трябва
да бъдат различни

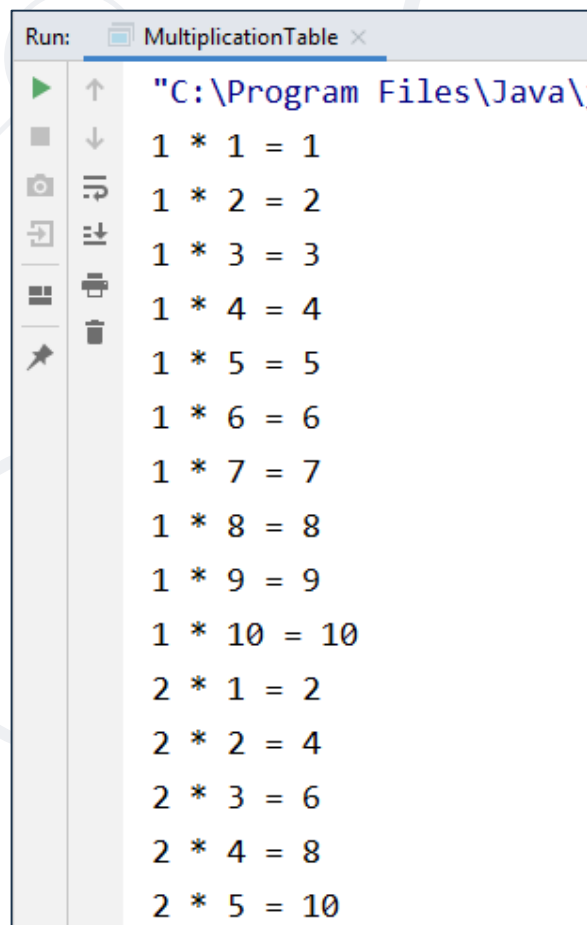
- За всяка итерация на външния цикъл вложения се изпълнява **n** - на брой пъти



Таблица за умножение - условие

- Отпечатайте на конзолата таблицата за умножение за числата от 1 до 10

- Изход:



```
Run: MultiplicationTable x
"C:\Program Files\Java\
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
```

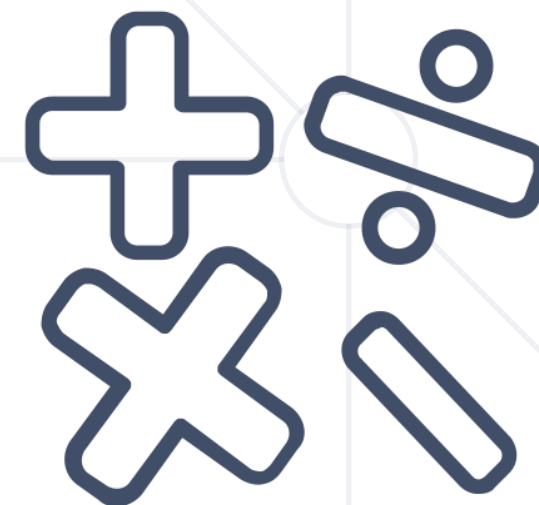


Таблица за умножение - решение

```
for (int x = 1; x <= 10; x++) {  
    for (int y = 1; y <= 10; y++) {  
        int product = x * y;  
        System.out.printf("%d * %d = %d%n", x, y, product);  
    }  
}
```


- За прекъсване на вложени цикли, използваме булеви променливи.

```
boolean flag = false;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (condition)
            flag = true;
            break;
    if (flag)
        break;
```

Външният цикъл ще се прекъсне, само ако стойността на flag бъде true

- Напишете програма, която проверява всички възможни комбинации от двойка числа в даден интервал
 - Ако се намери комбинация, чийто **сбор от числата е равен** на дадено **магическо число** на изхода **се отпечатва съобщение** и програмата приключва изпълнение
 - Ако не се намери **ниито една комбинация**, отговаряща на условието се отпечатва **съобщение, че не е намерено**

Сума от две числа – условие (2)

- Примерен вход и изход:

1
10
5



Combination N:4 ($1 + 4 = 5$)

23
24
20



4 combinations - neither equals 20

Сума от две числа - решение

```
int startingNumber = Integer.parseInt(scan.nextLine());
int finalNumber = Integer.parseInt(scan.nextLine());
int magicNumber = Integer.parseInt(scan.nextLine());
int combinations = 0;
boolean flag = false;
for (int i = startingNumber; i <= finalNumber; i++)
    for (int j = startingNumber; j <= finalNumber; j++)
        combinations++;
        if (i + j == magicNumber)
            System.out.printf("Combination N:%d (%d + %d = %d)%n",
                               combinations, i, j, magicNumber);
            flag = true;
            break;
    if (flag)
        break;
// TODO: Finish logic
```

Ако намерим
комбинация, прекъсваме
вътрешният цикъл

- Напишете програма, която изчислява **ASCII стойността** на няколко имена, като знаем че:
 - Името с **най-голяма стойност** е победител
 - Стойността на името е **сбора от ASCII стойностите** на всички **букви**
 - От конзолата ще се четат имена до получаването на команда **"STOP"**, след което трябва да се изпише:
"Winner is {името на победителя} – {стойността на името му}!".

Война на имена – условие (2)

- Примерен вход и изход:

Petar
Georgi
Stanimir
STOP



Winner is Stanimir – 839!

Ivo
Niki
Valio
Konstantin
STOP



Winner is Konstantin – 1065!

Война на имена - решение

```
String input = scanner.nextLine();
int maxCombination = 0;
String winner = "";
int currentSum = 0;
while ((!input.equals("STOP")))
    currentSum = 0;
    for (int i = 0; i < input.length(); i++) {
        currentSum += (int)(input.charAt(i));
    }
    if (currentSum > maxCombination)
        maxCombination = currentSum;
        winner = input;
        input = scanner.nextLine();
System.out.printf("Winner is %s - %d!", winner,
maxCombination);
```

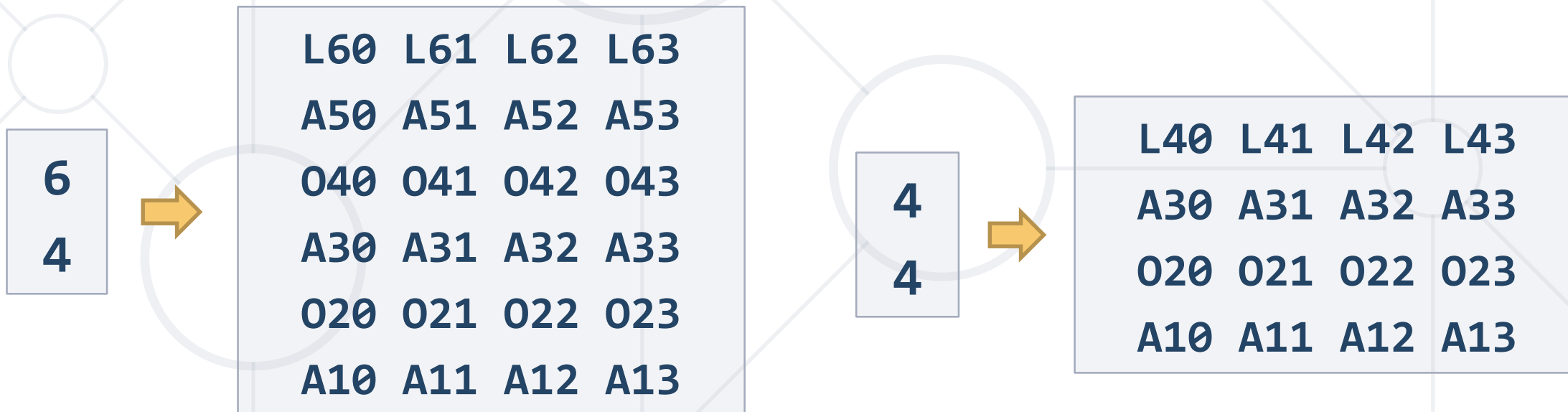
**Вложеният цикъл
обхожда името**

- Напишете програма, която извежда номерата на стаите в една сграда (в низходящ ред)
 - На всеки **четен** етаж има само **офиси**
 - На всеки **нечетен** етаж има само **апартаменти**
- Етажите се означават по следния начин:
 - Апартаменти: "А{номер на етажа}{номер на апартамента}"
 - Офиси: "О{номер на етажа}{номер на офиса}"
 - Номерата им винаги започват с 0



Сграда – условие (2)

- На последният етаж винаги има големи апартаменти, които се означават с 'L', вместо с 'A'
- Ако има само един етаж, то има само големи апартаменти
- Примерен вход и изход:



```
Scanner scanner = new Scanner(System.in);
int floors = Integer.parseInt(scanner.nextLine());
int rooms = Integer.parseInt(scanner.nextLine());
for (int i = floors; i >= 1; i--) {
    for (int j = 0; j < rooms; j++) {
        if (i == floors) {
            System.out.printf("L%d%d ", i, j);
        }
        // TODO: print according to floor number
    }
    System.out.println();
}
```

Вложеният цикъл
итерира стаите

- Напишете програма, която чете **последователност от продукти**, с които се приготвя тесто за бисквити
 - На първи ред ще се подаде едно цяло число – **броят на партидите**
 - След това **за всяка партида** ще се четат продукти до подаване на команда **"Bake"**:
 - Ако сместа **съдържа** продуктите **брашно, яйца и захар**, да се изписва на конзолата **"Baking batch number {номер на партидата}..."**
 - В **противен случай** да се изпише следното съобщение:
"The batter should contain flour, eggs and sugar!"
и програмата да продължи четенето

Фабрика за бисквити - условие (2)

- Примерен вход и изход:

2
flour
eggs
sugar
chocolate
Bake!
flour
eggs
sugar
caramel
peanuts
Bake!



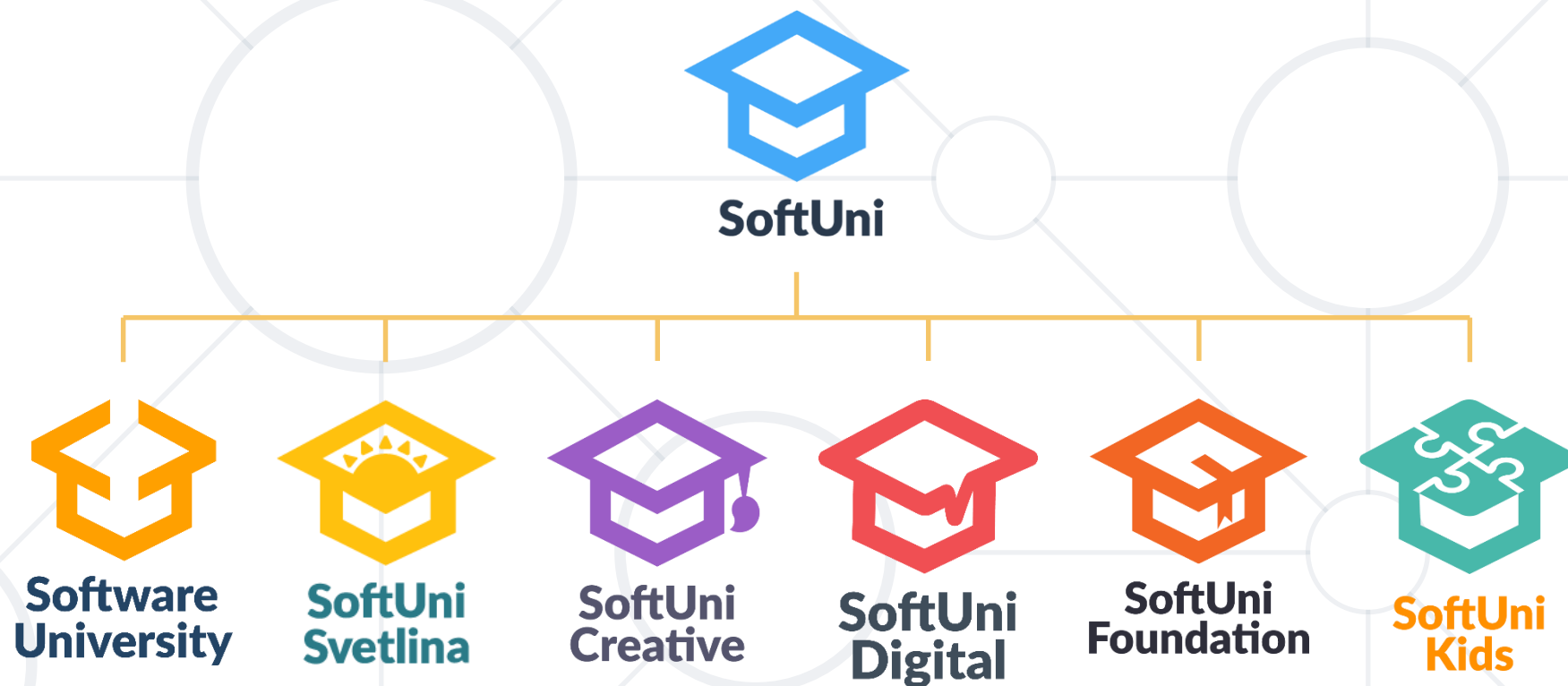
Baking batch number 1...
Baking batch number 2...

```
int numberOfBatches = Integer.parseInt(scanner.nextLine());
for (int i = 1; i <= numberOfBatches; i++) {
    boolean flour = false;
    boolean sugar = false;
    boolean eggs = false;
    boolean isBakingBatch = false;
    while (!isBakingBatch) {
        String command = scanner.nextLine();
        switch (command) {
            case "flour": flour = true; break;
            case "eggs": eggs = true; break;
            case "sugar": sugar = true; break;
        }
    }
    ...
}
```

Фабрика за бисквити – решение (2)

```
...  
    case "Bake!":  
        if (eggs && flour && sugar) {  
            isBakingBatch = true;  
            System.out.printf("Baking batch number %d...%n", i);  
            flour = false; eggs = false; sugar = false;  
            break;  
        } else {  
            System.out.println("The batter should contain flour, eggs  
and sugar!");  
        }  
        break;  
    }  
}
```

Въпроси?



SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре



INDEAVR

Serving the high achievers



INFRAGISTICS®



STEMO®
Computer Systems & Software

SUPERHOSTING.BG

SoftUni Organizational Partners

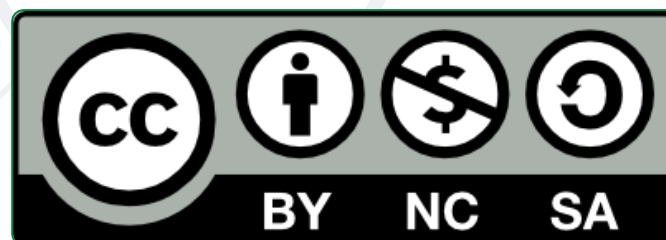


OneE
SOFTWARE



WORLD
OF
MYTHS

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
 - Книга "Основи на програмирането с Java" от Светлин Наков и колектив с лиценз CC-BY-SA

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

