# Упражнения: Mетоди

# Деклариране и извикване на методи

## Празна касова бележка

Създайте метод, който отпечатва празна касова бележка. Методът трябва да извиква три други метода: един за отпечатване на хедъра, един за основната част на бележката и един за футъра.

|  |  |
| --- | --- |
| Хедърът трябва да съдържа следния текст: | CASH RECEIPT  ------------------------------ |
| Основната част на бележката съдържа текста: | Charged to\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Received by\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Ето го и текста на футъра: | ------------------------------  © BG |

### **Пример**

|  |
| --- |
| Изход |
| CASH RECEIPT  ------------------------------  Charged to\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Received by\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ------------------------------  © BG |

### **Подсказки**

1. Първо създайте метод без параметри за отпечатването на хедъра. Дефиницията му започва със static void. Дайте му смислено име, например "PrintReceiptHeader" и напишете кода, който ще изпълнява този метод:

A picture containing graphical user interface

Description automatically generated

1. Направете същото и за същинската част и футъра на касовата бележка.
2. Създайте метод, който ще извиква тези три метода в правилния ред. И на него дайте смислено и описателно име, примерно "PrintReceipt" и напишете неговия програмен код:

A picture containing graphical user interface

Description automatically generated

1. За отпечатване на "©" използвайте Уникод-а "\u00A9"
2. Извиквайте методът PrintReceipt от main метода.

Text

Description automatically generated with low confidence

## Знак на цяло число

Създайте метод, отпечатващ знака на цяло число n.

### **Пример**

|  |  |
| --- | --- |
| Вход | Изход |
| 2 | The number 2 is positive. |
| -5 | The number -5 is negative. |
| 0 | The number 0 is zero. |

### **Подсказки**

1. Създайте метод с описателно име като "PrintSign". Методът трябва да получава един параметър от тип int.

A picture containing text

Description automatically generated

1. Изградете и тялото на метода, като обработите трите случая:
   1. Ако числото е по-голямо от нула
   2. Ако числото е по-малко от нула
   3. И ако числото е равно на нула
2. Извикайте новосъздадения метод от метода main.

Graphical user interface, text, application

Description automatically generated

## Отпечатване на триъгълник

Създайте метод за отпечатване на триъгълници както е показано по-долу:

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| 3 | 1  1 2  1 2 3  1 2  1 |
| 4 | 1  1 2  1 2 3  1 2 3 4  1 2 3  1 2  1 |

### **Подсказки**

1. След като прочетете входните данни
2. Започнете със създаването на метод за отпечатване на един ред от дадено число start до дадено число end. Изберете му смислено име, което описва неговото предназначение:

Text

Description automatically generated

1. Помислете как може да го използвате, за да намерите решение на задачата
2. След като анализирате заданието, би трябвало да стигнете до извода, че ви трябват два цикъла
3. В първия цикъл може да отпечатате горната половина от триъгълника без средния ред:

A picture containing text

Description automatically generated

1. След това отпечатваме средния ред:



1. Накрая отпечатваме и останалото от триъгълника:

A picture containing chart

Description automatically generated

## Изчертаване на запълнен квадрат

Изчертайте на конзолата запълнен квадрат с дължина на страната n като в примера:

### **Пример**

|  |  |
| --- | --- |
| Вход | Изход |
| 4 | --------  -\/\/\/-  -\/\/\/-  -------- |

### **Подсказки**

1. Прочетете входните данни
2. Създайте метод, който ще печати най-горния и най-долния ред (тъй като те са еднакви). Не забравяйте да му дадете описателно име и като параметър да му подадете някаква дължина.
   1. Вместо цикъл може да използвате командата "new string", която създава нов текст, съставен от символ, повторен определен брой пъти:

Graphical user interface, text

Description automatically generated

1. Сега създайте метод, който ще отпечатва средните редове. Е, ясно е, ще го наречете предполагам "PrintMiddleRow" ☺

Text, letter

Description automatically generated

1. Използвайте методите, които току-що създадохте, за изчертаването на квадрата:

Graphical user interface, text, application

Description automatically generated

# Връщане на стойности и предефиниране

## Конвертор за температури

Създайте метод, който конвертира температура от Фаренхайт в Целзий. Форматирайте резултата до втория десетичен знак.

Използвайте формулата: **(fahrenheit - 32) \* 5 / 9**.

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| 95 | 35.00 |
| 33.8 | 1.00 |
| -40 | -40.00 |

### **Подсказки**

1. Прочетете входните данни
2. Създайте метод, който **връща стойност от тип** double:  
   Text

   Description automatically generated
3. **Извикайте** метода в main и **запишете върнатата стойност в нова променлива**:  
   Text

   Description automatically generated

## Пресмятане на лице на триъгълник

Създайте метод, който изчислява и връща лицето на триъгълник по дадени основа и височина:

### **Пример**

|  |  |
| --- | --- |
| Вход | Изход |
| 3  4 | 6 |

### **Подсказки**

1. Първо прочетете входните данни
2. После създайте метод, но този път вместо да пишете "static void" преди името му, напишете "static double", така че да го накараме да върне стойност от тип double:

Text

Description automatically generated with low confidence

1. Извикайте метода в main и съхранете върната стойност в нова променлива:

Graphical user interface, text, application

Description automatically generated

## Повдигане на степен

Създайте метод, който пресмята и връща стойността на число, повдигнато на указаната степен:

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| 2  8 | 256 |
| 3  4 | 81 |

### **Подсказки**

1. Както обикновено, прочетете входните данни
2. Създайте метод, който ще има два параметъра - числото и степента, и ще връща резултат от тип double:

Graphical user interface, text, application

Description automatically generated

1. Отпечатайте резултата

## По-голямата от две стойности

Имате подадени като входни данни две стойности от един и същи тип. Стойностите може да са от тип int, char или string. Създайте метод GetMax() който връща по-голямата от двете стойности:

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| int  2  16 | 16 |
| char  a  z | z |
| string  Ivan  Todor | Todor |

### **Подсказки**

1. За тази задача ще трябва да създадете три метода с едно и също име и с различни сигнатури
2. Създайте метод, който ще сравнява цели числа:

Graphical user interface, text, application

Description automatically generated

1. Създайте втори метод със същото име, който ще сравнява символи. Следвайте логиката на предния метод:

A picture containing graphical user interface

Description automatically generated

1. И накрая създайте метод за сравняване на низове. Той ще е малко по-различен, тъй като низовете не може да бъдат сравнявани с операторите > и <

Text

Description automatically generated

Трябва да използвате метода "CompareTo()", който връща целочислена стойност (положителна ако сравняваният обект е по-голям, отрицателна, ако е по-малък и нула, ако двата са равни).

1. Последната стъпка е да прочетете входните данни, да използвате променливи от подходящ тип и да извикате GetMax() от вашия Main():

Text

Description automatically generated