# Допълнителни упражнения: Подпрограми

Практически упражнения към курса ["Увод в програмирането" за ученици](https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Basics).

Тествайте решенията си от тази тема в Judge: [https://judge.softuni.bg/Contests/2692/Допълнителни-задачи-за-методи](https://judge.softuni.bg/Contests/2692/%D0%94%D0%BE%D0%BF%D1%8A%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D0%BD%D0%B8-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8-%D0%B7%D0%B0-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8)

## "Hello, Име!"

Да се напише **метод**, който получава като параметър име и принтира на конзолата "*Hello, \!*".

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| Peter | Hello, Peter! |

### Насоки и подсказки

Дефинирайте метод **PrintName(string name)** и го имплементирайте, след което в главната програма прочетете от конзолата име на човек и извикайте метода като му подадете прочетеното име.

## По-малко число

Да се създаде метод GetMin(int a, int b), който връща по-малкото от две числа. Да се напише програма, която чете като входни данни от конзолата три числа и печата най-малкото от тях. Да се използва метода GetMin(…), който е вече създаден.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** |
| 1 2 3 | 1 | -100 -101 -102 | -102 |

### Насоки и подсказки

Дефинирайте метод **GetMin(int a, int b)** и го имплементирайте, след което го извикайте от главната програма както е показано по-долу. За да намерите минимума на три числа, намерете първо минимума на първите две от тях и след това минимума на резултата и третото число:

var min = GetMin(GetMin(num1, num2), num3);

## Повтаряне на низ

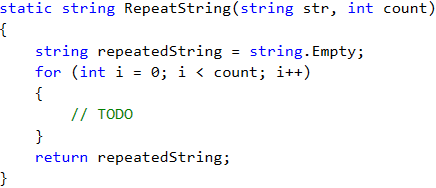
Да се напише метод RepeatString(str, count), който получава като параметри променлива от тип string и цяло число n и връща низа, повторен n пъти. След това резултатът да се отпечата на конзолата.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** |
| str  2 | strstr | roki  6 | rokirokirokirokirokiroki |

### Насоки и подсказки

Допишете метода по-долу като добавите съединяването входния низ към резултата в цикъла:



Имайте предвид, че в езика C# съединяването на низове в цикъл води до лоша производителност и не се препоръчва. Потърсете и пробвайте по-ефективни решения тук: <https://stackoverflow.com/questions/411752>

## N-та цифра

Да се напише метод FindNthDigit(number, index), който получава число и индекс N като параметри и печата N-тата цифра на числото (като се брои от дясно на ляво, започвайки от 1). След това, резултатът да се отпечата на конзолата.

### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** | **вход** | **изход** |
| 83746  2 | 4 | 93847837  6 | 8 | 2435  4 | 2 |

### Насоки и подсказки

За да изпълним алгоритъма, ще използваме **while** цикъл, докато дадено число не стане 0. На всяка итерация на **while** цикъла ще проверяваме дали настоящият индекс на цифрата не отговаря на индекса, който търсим. Ако отговаря, ще върнем като резултат цифрата на индекса (**number % 10**). Ако не отговаря, ще премахнем последната цифра на числото (**number = number / 10**). Трябва да следим коя цифра проверяваме по индекс (от дясно на ляво, започвайки от 1). Когато намерим цифрата, ще върнем индекса.

## Число към бройна система

Да се напише метод IntegerToBase(number, toBase), който получава като параметри цяло число и основа на бройна система и връща входното число, конвертирано към посочената бройна система. След това, резултатът да се отпечата на конзолата. Входното число винаги ще е в бройна система 10, а параметърът за основа ще е между 2 и 10.

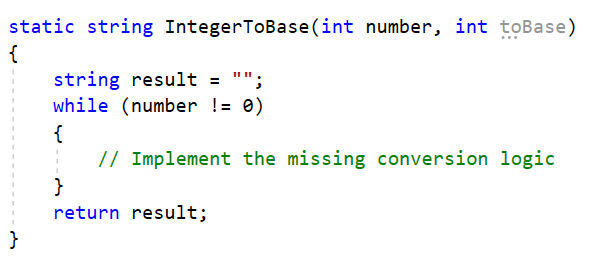
### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** | **вход** | **изход** |
| 3  2 | 11 | 4  4 | 10 | 9  7 | 12 |

### Насоки и подсказки

За да решим задачата, ще декларираме стрингова променлива, в която ще пазим резултата. След това трябва да изпълним следните изчисления, нужни за конвертиране на числото.

* Изчисляваме **остатъка** от числото, разделено на основата.
* **Вмъкваме остатъка** от числото в началото на низа, представящ резултата.
* **Разделяме** числото на основата.
* **Повтаряме** алгоритъма, докато входното число не стане 0.

Допишете липсващата логика в метода по-долу:

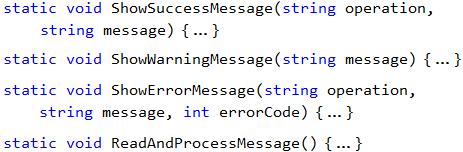
## Известия

Да се напише програма, която прочита цяло число **n** и на следващите редове въвежда **n** **съобщения** (като за всяко съобщение се прочитат по няколко реда). Всяко съобщение започва с **messageType**: **success**, **warning** или **error**:

* Когато **messageType** е **success** да се четат **operation** + **message** (всяко на отделен ред).
* Когато **messageType** е **warning** да се чете само **message**.
* Когато **messageType** е **error** да се четат **operation** + **message** + **errorCode** (всяко на отделен ред).

На конзолата да се отпечата **всяко прочетено съобщение**, форматирано в зависимост от неговия **messageType**. Като след заглавния ред за всяко съобщение да се отпечатат толкова на брой символа **=**, **колкото е дълъг** съответният **заглавен ред** и да се сложи по един **празен ред** след всяко съобщение (за по-детайлно разбиране погледнете примерите).

Задачата да се реши с дефиниране на четири метода: **ShowSuccessMessage()**, **ShowWarningMessage()**, **ShowErrorMessage()** и **ReadAndProcessMessage()**, като само последният метод да се извиква от главния **Main()** метод:



### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| 4 error credit card purchase Invalid customer address 500 warning Email not confirmed success user registration User registered successfully warning Customer has not email assigned | Error: Failed to execute credit card purchase. ============================================== Reason: Invalid customer address. Error code: 500.  Warning: Email not confirmed. =============================  Successfully executed user registration. ======================================== User registered successfully.  Warning: Customer has not email assigned. ========================================= |

### Насоки и подсказки

**Дефинирайте и имплементирайте посочените четири метода**.

В **ReadAndProcessMessage()** прочетете типа съобщение от конзолата и според прочетения тип прочетете останалите данни (още един два или три реда). След това извикайте съответния **метод за печатане на съответния тип съобщение**.

## \*Числа към думи

Да се напише метод **Letterize(number)**, който **прочита цяло число** и го **разпечатва с думи на английски език** според условията по-долу:

* **Да се отпечатат с думи стотиците, десетиците и единиците** (и евентуални минус) според правилата на английския език.
* Ако числото е **по-голямо от 999**, трябва да се принтира **"too large".**
* Ако числото е **отрицателно**, трябва да се **принтира "minus" преди него**.
* Ако числото **не е съставено от три цифри**, **не трябва да се принтира**.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** |
| 3 999 -420 1020 | nine-hundred and ninety nine minus four-hundred and twenty too large | 2 15 350 | fifteen three-hundred and fifty |
| **вход** | **изход** | **вход** | **изход** |
| 4 311 418 509 -9945 | three-hundred and eleven four-hundred and eighteen five-hundred and nine too small | 3 500 123 9 | five-hundred one-hundred and twenty three nine |

### Насоки и подсказки

Можем първо да отпечатаме **стотиците** като текст - (числото / 100) % 10, след тях **десетиците** - (числото / 10) % 10 и накрая **единиците** - (числото % 10).

Първият специален случай е когато числото е точно **закръглено на 100** (напр. 100, 200, 300 и т.н.). В този случай отпечатваме **"one-hundred", "two-hundred", "three-hundred**" и т.н.

Вторият специален случай е когато числото, формирано от последните две цифри на входното число, е **по-малко от 10** (напр. 101, 305, 609 и т.н.). В този случай отпечатваме **"one-hundred and one", "three-hundred and five", "six-hundred and nine" и т.н.**

Третият специален случай е когато числото, формирано от последните две цифри на входното число, е **по-голямо от 10 и по-малко от 20** (напр. 111, 814, 919 и т.н.). В този случай отпечатваме **"one-hundred and eleven", "eight-hundred and fourteen", "nine-hundred and nineteen" и т.н.**

## \*Криптиране на низ

Да се напише **метод Encrypt(char letter)**, който **криптира дадена буква** по следния начин:

* Вземат се **първата и последна цифра** от **ASCII кода на буквата** и се **залепят** една за друга в низ, който ще представя резултата.
* Към **началото на стойността на низа**, който представя резултата, се залепя **символа**, който отговаря на следното условие:
  + **ASCII кода на буквата + последната цифра от ASCII кода на буквата.**
* След това към **края на стойността на низа**, който представя **резултата**, се залепя **символа**, който отговаря на следното условие:
  + **ASCII кода на буквата - първата цифра от ASCII кода на буквата.**
* Методът трябва да върне **като резултат криптирания низ**.

Използвайки метода, описан по-горе, да се напише програма, която чете **поредица от символи**, **криптира ги** и **отпечатва резултата на един ред**.

Приемаме, че входните данни винаги ще бъдат валидни. Главният метод трябва да прочита входните данни, подадени от потребителя – цяло число **n**, следвани от по един символ на всеки от следващите **n** реда.

Да се **криптират символите** и да се добавят към **криптирания низ**. Накрая като резултат трябва да се отпечата **криптиран низ от символи** като в следващия пример.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **обяснения** |
| j | **p16i** | ASCII кодът на j e 106 → Първа цифра - 1, последна цифра - 6.  Залепяме първата и последната цифра → 16.  Към началото на стойността на низа, който представя резултата, залепяме символа, който се получава от сбора на ASCII кода + последната цифра → 106 + 6 → 112 → p.  • Към края на стойността на низа, който представя резултата, залепяме символа, който се получава от разликата на ASCII кода - първата цифра → 106 - 1 → 105 → i. |

|  |  |  |  |
| --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** |
| 7 S o f t U n i | V83Kp11nh12ez16sZ85Mn10mn15h | 7 B i r a H a x | H66<n15hv14qh97XJ72Ah97xx10w |

### Насоки и подсказки

На променливата от тип **string**, в която ще се **пази стойността на резултата**, ще присвоим първоначална стойност **string.Empty**. Трябва да се завърти цикъл **n** пъти, като на всяка итерация към променливата, в която пазим стойността на резултата, **ще прибавяме криптирания символ**.

За да намерим първата и последната цифри от ASCII кода, ще използваме алгоритъма, който използвахме за решаване на задача "N-та цифра", а за да създадем низа, ще процедираме както в задачата "Число към бройна система".

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

