# Упражнения: Логически изрази и проверки

## Празно Visual Studio решение (Blank Solution)

Създайте празно решение от Microsoft Visual Studio, за да организирате решенията на задачите от упражненията, като всяка задача ще бъде в отделен проект и всички проекти ще бъдат в общото решение.

1. Стартирайте Visual Studio.
2. Създайте нов проект [Create New Project].

Graphical user interface, text, application

Description automatically generated

1. Търсете и изберете: Blank Solution

Graphical user interface, text, application, email

Description automatically generated

1. Именувайте решението: Simple-Calculations

Graphical user interface, text, application, email

Description automatically generated

Сега имате създаденo празно решение без проекти в него.

Целта e да добавяте в него по един проект за всяка задача от упражненията.

## Проверка за отлична оценка

Първата задача от тази тема е да се напише конзолна програма, която чете оценка (десетично число), въведена от потребителя, и отпечатва “Excellent!”, ако оценката е 5.50 или по-висока.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 6 | Excellent! | 5 | *(няма изход)* | 5.50 | Excellent! | 5.49 | *(няма изход)* |

1. Създайте нов проект от менюто [File] 🡪 [New] 🡪 [New Project …]
2. Изберете шаблон на проекта [C#] 🡪 [Windows] 🡪 [Console] 🡪 [Console App].
3. Именувайте проекта: Excellent-Result.
4. Напишете кода на програмата от картинката:

Graphical user interface, text, application, email

Description automatically generated

1. Стартирайте програмата с [Ctrl+F5] и я тествайте с различни входни стойности:

Text

Description automatically generated

## Отлична оценка или не

Следващата задача от тази тема е да се напише конзолна програма, която чете оценка (десетично число), въведена от потребителя, и отпечатва “Excellent!”, ако оценката е 5.50 или по-висока, или “Not excellent.” в противен случай.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 6 | Excellent! | 5 | Not excellent. | 5.50 | Excellent! | 5.49 | Not excellent. |

1. Създайте нов проект от менюто [File] 🡪 [New] 🡪 [New Project …]
2. Изберете шаблон на проекта [C#] 🡪 [Windows] 🡪 [Console] 🡪 [Console App].
3. Именувайте проекта: Excellent-Or-Not.
4. Напишете кода на програмата от картинката:



1. Включете автоматично превключване на проектите, като кликнете върху решението с десния бутон на мишката и изберете [Set StartUp Projects…], от диалоговият прозорец изберете [Startup Project] 🡪 [Current selection]:

Graphical user interface, text, application, email

Description automatically generated

1. Сега стартирайте програмата, както обикновено с [Ctrl+F5] и я тествайте:

Text

Description automatically generated

## Четно или нечетно число

Да се напише програма, която чете цяло число, въведено от потребителя, и печата дали е четно или нечетно. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 2 | even | 3 | odd | 25 | odd | 1024 | even |

1. Добавете нов конзолен проект към решението, като именувайте проекта: Odd-Or-Even.
2. Напишете кода на програмата, като проверката за четност може да се реализира чрез проверка на остатъка при деление на 2 по следния начин: var even = (num % 2 == 0).
3. Стартирайте програмата с [Ctrl+F5] и я тествайте:

Text

Description automatically generated

## Намиране на по-голямото число

Да се напише програма, която чете две цели числа, въведени от потребителя, и отпечатва по-голямото от двете. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 5  3 | 5 | 3  5 | 5 | 10  10 | 10 | -5  5 | 5 |

1. Добавете нов конзолен проект към решението, като именувайте проекта: Bigger-Number.
2. Напишете кода на програмата. Необходима е единична if-else конструкция.
3. Стартирайте програмата с [Ctrl+F5] и я тествайте:

Text

Description automatically generated

Подсказка: може да си помогнете частично с кода от картинката, който е нарочно замъглен, за да помислите как да си го напишете сами:

Text

Description automatically generated

## Познай паролата

Да се напише програма, която чете парола (един ред с произволен текст), въведена от потребителя, и проверява дали въведеното съвпада с фразата “s3cr3t!P@ssw0rd”. При съвпадение да се изведе “Welcome”. При несъвпадение да се изведе “Wrong password!”. Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |
| qwerty | Wrong password! | s3cr3t!P@ssw0rd | Welcome | s3cr3t!p@ss | Wrong password! |

Подсказка: използвайте if-else конструкцията.

## Еднакви думи

Да се напише програма, която чете две думи, въведени от потребителя, и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе “yes” или “no”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| Hello  Hello | yes | SoftUni  softuni | yes | Soft  Uni | no | beer  vodka | no | HeLlO  hELLo | yes |

Подсказка: използвайте if-else конструкция. Преди сравняване на думите ги обърнете в долен регистър: word = word.ToLower().

## Време плюс 15 минути

Да се напише програма, която чете час и минути от 24-часово денонощие, въведени от потребителя, и изчислява колко ще е часът след 15 минути. Резултатът да се отпечата във формат hh:mm. Часовете винаги са между 0 и 23, а минутите винаги са между 0 и 59. Часовете се изписват с една или две цифри. Минутите се изписват винаги с по две цифри, с водеща нула когато е необходимо. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 1  46 | 2:01 | 0  01 | 0:16 | 23  59 | 0:14 | 11  08 | 11:23 | 12  49 | 13:04 |

Подсказка: добавете 15 минути и направете няколко проверки. Ако минутите надвишат 59, увеличете часовете с 1 и намалете минутите със 60. По аналогичен начин разгледайте случая, когато часовете надвишат 23. При печатането на минутите проверете за водеща нула.

## \* Три еднакви числа

Да се напише програма, която чете 3 числа и отпечатва дали са еднакви (yes / no).