# Упражнения: Условни конструкции

Тествайте решенията си в **Judge системата**: <https://judge.softuni.bg/Contests/3155/Conditional-Statements>

## Празно Visual Studio решение (Blank Project)

Създайте празно решение (**Blank Solution**) във Visual Studio. Решенията (solutions) във Visual Studio обединяват **група проекти**. Тази възможност е изключително удобна, когато искаме да работим по няколко проекта и бързо да превключваме между тях или искаме да обединим логически няколко взаимосвързани проекта.

В настоящото практическо занимание ще използваме **Blank Solution с няколко проекта,** за да организираме решенията на задачите от упражненията – всяка задача в отделен проект и всички проекти в общ solution.

1. Стартирайте Visual Studio.
2. Създайте нов проект: [Create a new project].

Graphical user interface, application

Description automatically generated

1. Изберете [Black solution], ако не го виждате, в търсачката изпишете ["Blank solution"].

Graphical user interface, text, application, email

Description automatically generated

1. Задайте подходящо име на проекта, например "Conditional-Statements":

Graphical user interface, text, application, email

Description automatically generated

Сега имате създаден **празен Visual Studio Solution** (с 0 проекта в него):

A screenshot of a social media post

Description automatically generated

Целта на този blank solution e да добавяте в него **по един проект за всяка задача** от упражненията.

## 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.

### Примери

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

### Насоки

1. Създайте **нов C# конзолен проект** с име "ExcellentResult" в **Blanc Project** с име " Conditional-Statements ", като натиснем с десен бутон на мишката в/у **Solution Condition-Statements->**  **add -> New Project...**:

A screenshot of a social media post

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Вече имате създаден клас със Main **метод**

A screenshot of a cell phone

Description automatically generated

1. Отидете в тялото на метода Main(string[] args) (между къдравите скоби). Създайте една променлива, в която да запазите **реално** **число** – оценката, което ще прочетете от конзолата:

A screenshot of a cell phone

Description automatically generated

1. Направете проверка за стойността на оценката. Ако тя е по-голяма или равна на 5.50 отпечатайте изхода по условие:

Graphical user interface, text, application

Description automatically generated

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

Graphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generated

## 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 |

### Насоки

1. Прочетете две цели числа от конзолата:

A screenshot of a cell phone

Description automatically generated

1. Сравнете, дали първото число num1 e по-голямо от второто num2. Отпечатайте по-голямото число.

A screenshot of a cell phone

Description automatically generated

## Четно или нечетно число

Да се напише програма, която чете **цяло число**, въведено от потребителя, и отпечатва на конзолата дали е **четно** или **нечетно**.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 2 | even | 3 | odd | 25 | odd | 1024 | even |

### Насоки

1. Първо добавете **нов конзолен проект** към съществуващия проект
2. Прочетете едно цяло число от конзолата:

A screenshot of a cell phone

Description automatically generated

1. Проверете дали числото е четно, като използвате модул оператора с 2 и проверите, дали има остатък от целочисленото деление. Отпечатайте изхода по условие – текста "even":

A screenshot of a cell phone

Description automatically generated

1. В противен случай отпечатайте "odd":

A screenshot of a cell phone

Description automatically generated

## Число от 100 до 200

Да се напише програма, която **чете цяло число**, въведено от потребителя и проверява, дали е **под 100**, **между 100 и 200** или **над 200**. Да се отпечатат съответно съобщения, като в примерите по-долу:

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 95 | Less than 100 | 120 | Between 100 and 200 | 210 | Greater than 200 |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете **едно цяло число** от конзолата:

Text

Description automatically generated with low confidence

1. Добавете **if-конструкция**, за да проверите дали числото е **по-малко от 100**, между **100 и 200** или **по-голямо от 100**:

Graphical user interface, text, application

Description automatically generated

## Информация за скоростта

Да се напише програма, която **чете скорост** **(реално число)**, въведена от потребителяи отпечатва **информация за скоростта**. При скорост **до 10** (включително) отпечатайте “**slow**”. При скорост **над 10** и **до 50** отпечатайте “**average**”. При скорост **над 50** и **до 150** отпечатайте “**fast**”. При скорост **над 150** и **до 1000** отпечатайте “**ultra fast**”. При по-висока скорост отпечатайте “**extremely fast**”. Примери:

### Примери

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 8 | slow | 49.5 | average | 126 | fast | 160 | ultra fast | 3500 | extremely fast |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете входните данни (**скоростта**):

Text

Description automatically generated with low confidence

1. Използвайте if-else-if конструкция, за да проверите **стойностите за скоростта**.

Graphical user interface, text, application

Description automatically generated

## Число в интервала

Да се напише програма, която проверява дали въведеното от потребителя число е в интервала [-100, 100] и е различно от 0 и извежда "**Yes**", ако отговаря на условията, или "**No**" ако е извън тях.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| -25 | Yes | 0 | No | 25 | Yes |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете **едно цяло число** от конзолата
3. Добавете **if-конструкция**, която проверява дали числото е **по-голямо или равно на -100**

Graphical user interface, text, application

Description automatically generated

1. Добавете **вложена** **if-конструкция**, която проверява дали числото е **по-малко или равно на 100**

Graphical user interface, text, application, email

Description automatically generated

## Билет за кино

Да се напише програма която чете ден от седмицата (текст) – въведен от потребителя и принтира на конзолата цената на билет за кино според деня от седмицата:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| 12 | 12 | 14 | 14 | 12 | 16 | 16 |

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Monday | 12 | Friday | 12 | Sunday | 16 |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете **един стринг** от конзолата – въведения ден
3. Създайте **switch-конструкция**, за да отпечатате цената на билета в зависимост от деня

Graphical user interface, text, application

Description automatically generated

## Невалидно число

TODO: Да се променят тестовете в Judge така че при вход 0 да се отпечатва invalid.

Дадено **число е валидно**, ако е в диапазона [**100**…**200**]. Да се напише програма, която **чете цяло число**, въведено от потребителя, и печата "invalid" ако въведеното число **не е валидно**.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 75 | invalid | 150 | *(няма изход)* | 220 | invalid | 199 | *(няма изход)* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| -1 | invalid | 100 | *(няма изход)* | 200 | *(няма изход)* | 0 | invalid |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете **едно цяло число** от конзолата
3. Добавете **вложени if-конструкции**, за да проверите дали числото е в диапазона между 100 и 200.
4. Отпечатайте **“invalid”**, ако числото **не е валидно**.

## 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! |

### Насоки

1. Добавете **нов конзолен проект** към съществуващия проект
2. Прочетете **един стринг** от конзолата – въведената парола
3. Добавете **if-конструкция**, която проверява дали въведената парола **е равна** на "s3cr3t!P@ssw0rd" и отпечатва съответния резултат.

## Ден от седмицата

Напишете програма, която чете **цяло** **число**, въведено от потребителя, и отпечатва **ден от седмицата** (на английски език), в граници [1...7] или отпечатва "Error" в случай, че въведеното число е **невалидно**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |
| 7 | Sunday |
| -1 | Error |

## Почивен или работен ден

Напишете програма която, чете ден от седмицата (**текст**), на английски език - въведен от потребителя. Ако денят е работен отпечатва на конзолата - "**Working day**", ако е почивен - "**Weekend**". Ако се въведе текст различен от ден от седмицата да се отпечата - "**Error**".

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Monday | Working day |  | Sunday | Weekend |  | April | Error |

## Плод или зеленчук

Да се напише програма, която **чете име на продукт**, въведено от потребителя, и проверява дали е **плод** или **зеленчук**.

* Плодовете "fruit" имат следните възможни стойности: **banana**, **apple**, **kiwi**, **cherry**, **lemon** и **grapes**
* Зеленчуците "vegetable" имат следните възможни стойности: **tomato**, **cucumber**, **pepper** и **carrot**
* Всички останали са "unknown"

Да се изведе "fruit”, "vegetable" или "unknown" според въведения продукт.

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| banana | fruit | apple | fruit | tomato | vegetable | water | unknown |

## Конвертор за мерни единици

Да се напише програма, която **преобразува разстояние** между следните **3 мерни единици**: mm, cm, m. Използвайте съответствията от таблицата по-долу:

|  |  |
| --- | --- |
| **Входна единица** | **Изходна единица** |
| **1** meter (**m**) | **1000** millimeters (**mm**) |
| **1** meter (**m**) | **100** centimeters (**cm**) |

Входните данни се състоят от три реда, въведени от потребителя:

* **Първи ред:** число за преобразуване - **реално число**
* **Втори ред:** входна мерна единица - **текст**
* **Трети ред:** изходна мерна единица (за резултата) - **текст**

На конзолата да се отпечата **резултатът от преобразуването** на мерните единици **форматиран** до **третия знак след десетичната запетая**.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 12  mm  m | 0.012 | 150  m  cm | 15000.000 | 45  cm  mm | 450.000 |