# Упражнения: Прости проверки

Задачи за упражнение в клас и за домашно към курса [„Fundamentals of programming with C#“ @ СофтУни](https://softuni.bg/trainings/1770/programming-basics-with-csharp-october-2017).

## Празно Visual Studio решение (Blank Solution)

Създайте празно решение (**Blank Solution**) във Visual Studio за да организирате решенията на задачите от упражненията – всяка задача ще бъде в отделен проект и всички проекти ще бъдат в общ solution.

1. Стартирайте Visual Studio.
2. Създайте нов **Blank Solution**: [File]🡪 [New] 🡪 [Project].



1. Изберете от диалоговия прозорец [Templates] 🡪 [Other Project Types] 🡪 [Visual Studio Solutions] 🡪 [**Blank Solution**] и дайте подходящо име на проекта, например “Simple-Conditions”:



Сега имате създаден **празен Visual Studio Solution** (без проекти в него):



Целта на този **blank solution** 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** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution 'Simple-Conditions'**. Изберете [Add] 🡪 [New Project…]:



1. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете C# конзолно приложение и задайте име “Excellent-Result”:



Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.

1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



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

 





## 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. Първо създайте **нов C# конзолен проект** в решението “**Simple-Conditions**”.
   * Кликнете с мишката върху решението в Solution Explorer и изберете [Add] 🡪 [**New Project**…].
   * Изберете [Visual C#] 🡪 [Windows] 🡪 [**Console Application**] и задайте име “Excellent-or-Not”.
2. **Напишете кода** на програмата. Може да си помогнете с примерния код от картинката:



1. Включете режим на **автоматично превключване към текущия проект** като кликнете върху главния solution с десния бутон на мишката и изберете **[Set StartUp Projects…]**:



Ще се появи диалогов прозорец, от който трябва да се избере **[Startup Project]** 🡪 [**Current selection**]:



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

 



## Четно или нечетно

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. Първо добавете **нов C# конзолен проект** в съществуващия solution.
2. **Напишете кода** на програмата. 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:



## Изписване на число до 10 с думи

Да се напише програма, която чете **цяло число в диапазона [0…9]**, въведено от потребителя, и го **изписва с думи** на английски език. Ако числото е извън диапазона, изписва “**number too big**”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5 | five | 1 | one | 9 | nine | 10 | number too big |

**Подсказка**: можете да напишете дълга if-else-if-else…else, с която да разгледате възможните **10 случая**.

## Квартално магазинче

Следващата задача има за цел да тренира работата с **вложени проверки** (nested if). Ето го и условието: предприемчив българин отваря **квартални магазинчета** в **няколко града** и продава на **различни цени**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| град / продукт | **coffee** | **water** | **beer** | **sweets** | **peanuts** |
| **Sofia** | 0.50 | 0.80 | 1.20 | 1.45 | 1.60 |
| **Plovdiv** | 0.40 | 0.70 | 1.15 | 1.30 | 1.50 |
| **Varna** | 0.45 | 0.70 | 1.10 | 1.35 | 1.55 |

Напишете програма, която чете **град** (стринг), **продукт** (стринг) и **количество** (десетично число), въведени от потребителя, и пресмята и отпечатва **колко струва** съответното количество от избрания продукт в посочения град. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| coffee  Varna  2 | 0.9 | peanuts  Plovdiv  1 | 1.5 | beer  Sofia  6 | 7.2 | water  Plovdiv  3 | 2.1 | sweets  Sofia  2.23 | 3.2335 |

1. Създайте **нов проект** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution** реда и изберете [Add] 🡪 [New Project…]:



1. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете **C#** **конзолно приложение** и задайте подходящо име, например “Small-Shop”:



Вече имате ново конзолно приложение и остава да напишете кода за решаване на задачата.

1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу. Можете да прехвърлите всички букви в долен регистър с .ToLower() за да сравнявате продукти и градове без значение на малки / главни букви:



1. За **да активирате текущия проект** да стартира при [Ctrl+F5], избере “**Set StartUp Projects…**”:



Изберете първата опция:



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

 

## Точка в правоъгълник

Напишете програма, която проверява дали **точка {x, y}** се намира **вътре в правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда, въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Една точка е вътрешна за даден правоъгълник, ако се намира някъде във вътрешността му или върху някоя от страните му. Отпечатайте “**Inside**” или “**Outside**”. Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside |  | 2  -3  12  3  11  -3.5 | Outside |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| -1  -3  4  1  0.5  1 | Inside |  | -1  -3  4  1  -1.2  1.4 | Outside |  |

\* **Подсказка**: една точка е вътрешна за даден многоъгълник, ако едновременно са изпълнени следните четири условия (можете да ги проверите с if проверка с логическо „**и**“ – оператор &&):

* Точката е надясно от лявата стена на правоъгълника (**x >= x1**)
* Точката е наляво от дясната стена на правоъгълника (**x <= x2**)
* Точката е надолу от горната стена на правоъгълника (**y >= y1**)
* Точката е нагоре от долната стена на правоъгълника (**y <= y2**)

## Плод или зеленчук?

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

* Плодовете "**fruit**" са **banana**, **apple**, **kiwi**, **cherry**, **lemon** и **grapes**
* Зеленчуците "**vegetable**" са **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 |

\* **Подсказка**: използвайте условна if проверка с логическо „**или**“ – operator ||.

## Бонус точки

Дадено е **цяло число** – брой точки. Върху него се начисляват **бонус точки** по правилата, описани по-долу. Да се напише програма, която пресмята **бонус точките** за това число и **общия брой точки** с бонусите.

* Ако числото е **до 100** включително, бонус точките са **5**.
* Ако числото е **по-голямо от 100**, бонус точките са **20%** от числото.
* Ако числото е **по-голямо от 1000**, бонус точките са **10%** от числото.
* Допълнителни бонус точки (начисляват се отделно от предходните):
  + За **четно** число 🡪 + 1 т.
  + За число, което **завършва на 5** 🡪 + 2 т.

Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 20 | 6  26 | 175 | 37  212 | 2703 | 270.3  2973.3 | 15875 | 1589.5  17464.5 |

Ето как би могло да изглежда решението на задачата в действие:



**Подсказка**:

* Основните бонус точки можете да изчислите с if-else-if-else-if конструкция (имате 3 случая).
* Допълнителните бонус точки можете да изчислите с if-else-if конструкция (имате още 2 случая).

## Конвертор за мерни единици

Да се напише програма, която **преобразува разстояние** между следните 8 **мерни единици**: m, mm, cm, mi, in, km, ft, yd. Използвайте съответствията от таблицата по-долу:

|  |  |
| --- | --- |
| **входна единица** | **изходна единица** |
| **1** meter (**m**) | **1000** millimeters (**mm**) |
| **1** meter (**m**) | **100** centimeters (**cm**) |
| **1** meter (**m**) | **0.000621371192** miles (**mi**) |
| **1** meter (**m**) | **39.3700787** inches (**in**) |
| **1** meter (**m**) | **0.001** kilometers (**km**) |
| **1** meter (**m**) | **3.2808399** feet (**ft**) |
| **1** meter (**m**) | **1.0936133** yards (**yd**) |

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

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

Резултатът **да се форматира** до **осмия знак след десетичната запетая**.

Примерен вход и изход:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 12  km  ft | 39370. 07886932 | 150 mi in | 9503999.99393599 | 450  yd  km | 0.41148000 |

## 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().

## Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична** фигура и пресмята лицето й. Фигурите са четири вида: квадрат (**square**), правоъгълник (**rectangle**), кръг (**circle**) и триъгълник (**triangle**). На първия ред на входа се чете вида на фигурата (square, rectangle, circle или triangle). Ако фигурата е **квадрат**, на следващия ред се чете едно число – дължина на страната му. Ако фигурата е **правоъгълник**, на следващите два реда четат две числа – дължините на страните му. Ако фигурата е **кръг**, на следващия ред чете едно число – радиусът на кръга. Ако фигурата е **триъгълник**, на следващите два реда четат две числа – дължината на страната му и дължината на височината към нея. Резултатът да се закръгли до **3 цифри след десетичната точка**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| square  5 | 25 | rectangle  7  2.5 | 17.5 | circle  6 | 113.097 | triangle  4.5  20 | 45 |

**Подсказка**: използвайте серия от if-else-if-else**-**… конструкции, за да обработите 4-те вида фигури.

## Време + 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.

## Обръщение според възраст и пол

Първата задача от тази тема е да се напише **конзолна програма**, която **прочита възраст** (десетично число) и **пол** (“m” или “f”), въведени от потребителя, и отпечатва **обръщение** измежду следните:

* “Mr.” – мъж (пол “m”) на 16 или повече години
* “Master” – момче (пол “m”) под 16 години
* “**Ms.**” – жена (пол “f”) на 16 или повече години
* “**Miss**” – момиче (пол “f”) под 16 години

Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 12  f | Miss | 17  m | Mr. | 25  f | Ms. | 13.5  m | Master |

1. Създайте **нов проект** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution** реда и изберете [Add] 🡪 [New Project…]:



1. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете **C#** **конзолно приложение** и задайте подходящо име, например “Personal-Titles”:



Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.

1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



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

 





## Точка върху страната на правоъгълник

Напишете програма, която проверява дали **точка {x, y}** се намира **върху някоя от страните на правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Да се отпечата “Border” (точката лежи на някоя от страните) или “Inside / Outside” (в противен случай). Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside / Outside |  | 2  -3  12  3  12  -1 | Border |  |

\* **Подсказка**: използвайте една или няколко условни if проверки с логически операции. Точка **{x, y}** лежи върху някоя от страните на правоъгълник **{x1, y1} – {x2, y2}**, ако е изпълнено едно от следните условия:

* **x** съвпада с **x1** или **x2** и същевременно **y** е между **y1** и **y2**
* **y** съвпада с **y1** или **y2** и същевременно **x** е между **x1** и **x2**

Можете да проверите горните условия с една по-сложна if-else конструкция или с няколко по-прости проверки или с **вложени** if-else проверки.

## Магазин за плодове

Магазин за плодове през **работните дни** работи на следните **цени**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **плод** | banana | apple | orange | grapefruit | kiwi | pineapple | grapes |
| **цена** | 2.50 | 1.20 | 0.85 | 1.45 | 2.70 | 5.50 | 3.85 |

**Събота** и **неделя** магазинът работи на **по-високи** **цени**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **плод** | banana | apple | orange | grapefruit | kiwi | pineapple | grapes |
| **цена** | 2.70 | 1.25 | 0.90 | 1.60 | 3.00 | 5.60 | 4.20 |

Напишете програма, която чете от конзолата **плод** (banana / apple / orange / grapefruit / kiwi / pineapple / grapes), **ден от седмицата** (Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / Sunday) и **количество** (десетично число) , въведени от потребителя, и пресмята **цената** според цените от таблиците по-горе. Резултатът да се отпечата **закръглен с 2 цифри** след десетичната точка. При невалиден ден от седмицата или невалидно име на плод да се отпечата “**error**”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| apple  Tuesday  2 | 2.40 | orange  Sunday  3 | 2.70 | kiwi  Monday  2.5 | 6.75 | grapes  Saturday  0.5 | 2.10 | tomato  Monday  0.5 | error |

\* **Подсказки**:

* Прочетете входа и обърнете името на плода и деня от седмицата в **малки букви**:



* Първоначално задайте цена **-1**:



* Използвайте вложени if проверки, за да изчислите цената за дадения плод и ден от седмицата:



* Накрая проверете цената. Ако все още е **-1**, значи даденият плод или денят от седмицата е **невалиден**. За да отпечатате точно **2 цифри след десетичната точка** (със закръгляне), използвайте форматиращ низ “{0:f2}”. Кодът може да е подобен на следния:



## Търговски комисионни

Фирма дава следните **комисионни** на търговците си според **града**, в който работят и обема на **продажбите** s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Град** | **0 ≤ s ≤ 500** | **500 < s ≤ 1 000** | **1 000 < s ≤ 10 000** | **s > 10 000** |
| Sofia | 5% | 7% | 8% | 12% |
| Varna | 4.5% | 7.5% | 10% | 13% |
| Plovdiv | 5.5% | 8% | 12% | 14.5% |

Напишете **конзолна програма**, която чете име на **град** (стринг) и обем на **продажби** (десетично число) , въведени от потребителя, и изчислява и извежда размера на търговската **комисионна** според горната таблица. Резултатът да се изведе закръглен с **2 цифри след десетичната точка**. При **невалиден** град или обем на продажбите (отрицателно число) да се отпечата “**error**”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| Sofia  1500 | 120.00 | Plovdiv  499.99 | 27.50 | Varna  3874.50 | 387.45 | Kaspichan  -50 | error |

\* **Подсказки**:

* Прочетете входа и **обърнете града в** **малки букви** (като в предходната задача).
* Първоначално задайте **комисионна -1**. Тя ще бъде променена, ако градът и ценовият диапазон бъдат намерени в таблицата с комисионните.
* Използвайте вложени if проверки, за **да изчислите комисионната** според града и според обема на продажбите. Може да си помогнете с кода по-долу:



* Накрая проверете комисионната. Ако все още е **-1**, значи въведеният град или обем продажби не се срещат в таблицата с комисионните и трябва да се отпечата “**error**”. В противен случай трябва да се изчисли комисионната (процент комисионна по обем на продажбите) и да се отпечата със закръгляне с точно **2 цифри след десетичната точка**. Може да използвате Console.WriteLine("{0:f2}", …).

## Кино

В една кинозала столовете са наредени в правоъгълна форма в r реда и c колони. Има три вида прожекции с билети на различни цени:

* Premiere – премиерна прожекция, на цена 12.00 лева.
* Normal – стандартна прожекция, на цена 7.50 лева.
* Discount – прожекция за деца, ученици и студенти на намалена цена от 5.00 лева.

Напишете програма, която чете **тип прожекция** (стринг), брой **редове** и брой **колони** в залата (цели числа), въведени от потребителя, и изчислява общите приходи от билети при пълна зала. Резултатът да се отпечата във формат като в примерите по-долу, с 2 знака след десетичната точка. Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| Premiere  10  12 | 1440.00 leva | Normal  21  13 | 2047.50 leva | Discount  12  30 | 1800.00 leva |

\* **Подсказка**: използвайте прости проверки и елементарни изчисления. За да изведете резултата с точно 2 цифри след десетичната точка, използвайте Console.WriteLine("{0:f2}", result).