# Упражнения: Условни конструкции

Тествайте решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/3896>

## 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 взаимносвързани 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 [Blank 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.

## Обръщение според възраст и пол

Да се напише конзолна програма, която **прочита възраст** (реално 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 |

### Насоки

1. Прочетете **възрастта (age)** от конзолата, на следващия ред прочетете **пола (gender)**.

Graphical user interface, text

Description automatically generated

1. Направете **проверка за пола** и ако върне резултат true,направете **проверка за годините**. В тялото на проверките за възраст принтирайте желаното обръщение.

Graphical user interface, text, application

Description automatically generated 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 generated Text

Description automatically generated

Трябва да получите **100 точки** (напълно коректно решение):

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Квартално магазинче

Предприемчив българин отваря **квартални магазинчета** в **няколко града** и продава на **различни цени според града**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| град / продукт | **coffee** | **water** | **juice** | **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 **продукт** (низ), **град** (низ) 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 | juice  Sofia  6 | 7.2 | water  Plovdiv  3 | 2.1 | sweets  Sofia  2.23 | 3.2335 |

### Насоки

1. Прочетете **входните данни**

Graphical user interface, text

Description automatically generated

1. Създайте **нова променлива**, която ще съхранява **общата цена** на продуктите

Icon

Description automatically generated with low confidence

1. Създайте **switch-конструкция**, за да проверите въведения град

Diagram

Description automatically generated with medium confidence

1. За всеки град добавете **проверки за продукта** и **изчислете цената**

**Text

Description automatically generated**

1. След switch-конструкцията отпечатайте **цената на продуктите**.

****

## GUI App – Квартално магазинче

Реализирайте предходната задача като **Windows Forms GUI app**. Потребителският интерфейс трябва да изглежда така:

Graphical user interface, application

Description automatically generated

### Насоки

1. Създайте нов **Windows Forms проект** и му задайте **смислено име**, например "**SmallShopApp**"
2. Променете **името** на формата: "**Form1**" 🡪 "**FormProductsPrice**"
3. Променете **заглавието** на формата: "**Form 1**" 🡪 "**Products Price**"
4. Добавете необходимите **контроли**:

* **четири** надписа (Label):
  + "labelProduct" с текст "Product: "
  + "labelCity" с текст "City: "
  + "labelQuantity" с текст "Quantity: "
  + "labelPrice" с текст "Price: "
* **две** падащи менюта (ComboBox):
  + "comboBoxProduct"
  + "comboBoxCity"
* **едно** текстово поле (TextBox):
  + "textBoxQuantity"
* **един** бутон (Button):
  + "buttonCalculate" с текст "Calculate price"

Graphical user interface, application

Description automatically generated

1. Добавете възможните **опции** за **продукти** на падащото меню **comboBoxProduct**:

Graphical user interface, text, application

Description automatically generated

1. Добавете възможните **опции** за **град** на падащото меню **comboBoxCity**:

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Добавете **метод-обработчик** на бутона **buttonCalculate** при събитието **Click**:

Graphical user interface, text, application

Description automatically generated

1. **Стартирайте** приложението с **[Ctrl + F5]** и го **тествайте**:

Graphical user interface, application

Description automatically generated

\* Помислете какво се случва, **ако не бъдат попълнени коректно полетата във формата**. Как можете да се справите в този въпрос?

## 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** (реално число), въведени от потребителя, и пресмята **цената** според цените от таблиците по-горе. Резултатът да се отпечата **закръглен до 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 |

### Насоки

1. Прочетете **входните данни**

Graphical user interface, text

Description automatically generated

1. Добавете **нова променлива**, която съхранява **цената на плодовете**

Icon

Description automatically generated

1. Добавете **switch-конструкция**, която проверява дали денят е **работен**, или **почивен**:

Graphical user interface, text, application

Description automatically generated

1. За всеки ден добавете **вложена if-конструкция**, която изчислява **цената** в зависимост от **плода** и **количеството**

Text

Description automatically generated with medium confidence

1. След switch-конструкцията отпечатайте **цената на плодовете**

## Сумиране на секунди

Трима спортни състезатели финишират за някакъв **брой секунди** (между **1** и **50**). Да се напише програма, която чете времената на състезателите в секунди, въведени от потребителя, и пресмята **сумарното им време** във формат "минути:секунди". Секундите да се изведат с **водеща нула** (2 🡪 "02", 7 🡪 "07", 35 🡪 "35").

### Примери

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 35  45  44 | 2:04 | 22  7  34 | 1:03 | 50  50  49 | 2:29 | 14  12  10 | 0:36 |

### Насоки

1. Прочетете входните данни - **секундите на състезателите**

Graphical user interface, text, application

Description automatically generated

1. Създайте **нова променлива**, в която да съхранявате **сбора от секундите на тримата състезатели**



1. След като сте намерили **сбора от секундите**, трябва да ги **превърнете в минути и секунди** (например, ако сборът е **85 секунди**,това са **1 минута и 25 секунди**, защото **1 минута** има **60 секунди**). Създайте **две нови променливи**. В първата изчислете **колко минути е сборът от секунди**, като **разделите сбора на 60**. Във втората променлива **изчислете секундите** с помощта на **деление с остатък (%).** Използвайте деление с остатък (%), за да вземете **остатъка при деление с 60**, което са останалите секунди. Например имате общ сбор от 134 секунди (2 минути и 14 секунди) след целочисленото деление (/) на 60 ще получим 2, а след делението с остатък (%) ще получим оставащите секунди - 14.

A picture containing text

Description automatically generated

1. След като вече знаете **колко минути и секунди** е общия сбор, трябва да ги принтираме в правилния формат **(минути : секунди**), като ако секундите са **по-малки от 10** трябва да отпечатаме **0 преди секундите**, в противен случай просто си отпечатваме **резултата в дадения формат**. За целта **направете проверка** (if). За принтирането може да използвате **шаблон**.

## Бонус точки

Дадено е цяло число – **начален брой точки**. Върху него се начисляват **бонус точки** по правилата, описани по-долу. Да се напише програма, която пресмята **бонус точките**, които получава числото, и **общия брой точки** (числото + бонуса).

* 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 |

### Насоки

1. Прочетете входните данни – **началния брой точки**
2. Създайте си **нова променлива от тип double**, в която ще си изчислявате **натрупаните бонус точки**, като ѝ дадете **начална стойност 0.0**
3. Направете **if-else-if конструкция** за първите три проверки, за да проверите **големината на числото** и да **изчислите бонуса**
4. Направете **нова if-else-if конструкция**, за да извършите останалите проверки и да **изчислите допълнителния бонус**
   * ако числото **е четно** към до момента натрупания бонус **добавете 1**
   * ако **завършва на 5, към бонуса добавете 2**

За да проверите дали едно число **е четно**, трябва да го **разделите на 2** и ако получавате **остатък при делението 0**, то значи числото е **четно**. Ако получите **остатък 1**, това означава, че числото е **нечетно**. Например числото 34 е четно, защото 34 / 2 = 17 и остатъкът е 0, а числото 35 е нечетно, защото 35 / 2 = 17 с остатък 1.

За да проверите дали едно число **завършва на 5**, трябва **да разделите числото на 10** и ако **получите остатък при делението 5**, то значи числото **завършва на 5**. Например числото 245 / 10 = 24 с остатък 5.

1. Принтирайте **на два реда** резултатите. На първия ред **натрупания бонус**, а на втория **крайното число**, което ще намерите, като **съберете началния брой точки и бонуса**.

## Търговски комисиони

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** (реално число), въведени от потребителя, изчислява и извежда **размера на търговската** **комисиона** според горната таблица. Резултатът да се изведе **форматиран до** **2 цифри** след десетичната запетая. 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 |

## Годзила срещу Конг

Снимките за дългоочаквания филм "Годзила срещу Конг" започват. Сценаристът Адам Уингард ви моли да напишете програма, която да изчисли **дали предвидените средства са достатъчни** за снимането на филма. За снимките ще бъдат нужни **определен брой статисти**, **облекло за всеки статист** и **декор.**

Известно е, че:

* Декорът за филма е на стойност **10% от бюджета.**
* При **повече от 150 статисти**,има **отстъпка** **за облеклото** на стойност **10%.**

### Вход

От конзолата се четат **3 реда**:

1. **Бюджет за филма** – реално число в интервала [1.00 … 1000000.00]
2. **Брой на статистите** – цяло число в интервала [1 … 500]
3. **Цена за облекло на един статист** – реално число в интервала [1.00 … 1000.00]

### Изход

На конзолата трябва да се отпечатат **два реда**:

* Ако парите за декора и дрехите **са повече от бюджета**:
  + "Not enough money!"
  + "Wingard needs {парите недостигащи за филма} leva more."
* Ако парите за декора и дрехите са **по малко или равни на бюджета**:
  + "Action!"
  + "Wingard starts filming with {останалите пари} leva left."

Резултатът трябва да е форматиран до втория знак след десетичната запетая.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 20000  120  55.5 | Action!  Wingard starts filming with 11340.00 leva left. | Сума за декор: 10% от 20000 = 2000 лв.  Сума за облекло: 120 \* 55.5 = 6660 лв.  Обща сума за филма: 2000 + 6660 = 8660 лв.  20000 – 8660 = 11340 лева остават. |
| 15437.62  186  57.99 | Action!  Wingard starts filming with 4186.33 leva left. | Сума за декор: 10% от 15437.62 = 1543.762 лв.  Сума за облекло: 186 \* 57.99 = 10786.14 лв.  Статистите са повече от 150 следователно има 10% отстъпка на облеклото.  10% от 10786.14 е 1078.614  10786.14 – 1078.614 = 9707.526 лв. за облекло  Обща сума за филма: 1543.762 + 9707.526 = 11251.288  15437.62 – 11251.288 = 4186.331 лева остават |
| 9587.88  222  55.68 | Not enough money!  Wingard needs 2495.77 leva more. | Сума за декор: 10% от 9587.88 = 958.788 лв.  Сума за облекло: 11124.864 лв.  Обща сума за филма: 958.788 + 11124.864 = 12083.652  9587.88 – 12083.652 = 2495.77 лева не достигат |

## Нов дом

Марин и Нели си купуват къща недалеч от София. Нели толкова много обича цветята, че ви убеждава да **напишете програма**, която да **изчисли колко ще им струва**, да си засадят определен брой цветя и **дали наличният бюджет ще им е достатъчен**. Различните цветя са с различни цени.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Цвете** | **Роза** | **Далия** | **Лале** | **Нарцис** | **Гладиола** |
| **Цена на брой в лева** | 5 | 3.80 | 2.80 | 3 | 2.50 |

**Съществуват следните отстъпки:**

* Ако Нели купи **повече от 80 рози** ➡ **10% отстъпка** от крайната цена
* Ако Нели купи **повече от 90 далии** ➡ **15% отстъпка** от крайната цена
* Ако Нели купи **повече от 80 лалета** ➡ **15% отстъпка** от крайната цена
* Ако Нели купи **по-малко от 120 нарциса** ➡ цената се **оскъпява с 15%**
* Ако Нели купи **по-малко от 80 гладиоли** ➡цената се **оскъпява с 20%**

От конзолата се **четат 3 реда:**

* **Вид цветя** - текст с възможности - "Roses", "Dahlias", "Tulips", "Narcissus", "Gladiolus"
* **Брой цветя** - цяло число в интервала [10…1000]
* **Бюджет** - цяло число в интервала [50…2500]

Да се **отпечата** на конзолата **на един ред:**

* **Ако бюджетът е достатъчен** ➡"Hey, you have a great garden with {броя цвета} {вид цветя} and {останалата сума} leva left."
* **Ако бюджетът НЕ е достатъчен** ➡ "Not enough money, you need {нужната сума} leva more."

Сумата да бъде форматирана **до втория знак** след десетичната запетая.

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| Roses  55  250 | Not enough money, you need 25.00 leva more. | Нели иска 55 броя Рози. Цената на една роза е 5лв., следователно за 55 броя Нели ще трябва да плати: 55 \* 5 = 275.  Тя обаче разполага с 250 лв. бюджет.  Понеже 275 > 250 , то не и достигат 25 лв. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| Tulips  88  260 | Hey, you have a great garden with 88 Tulips and 50.56 leva left. | Narcissus  119  360 | Not enough money, you need 50.55 leva more. |