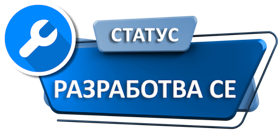
# Упражнения: Делегати и събития – продължение

Можете да проверите решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/4071>

Състояние на настоящия учебен материал:



## Обърнете и изключете

Напишете програма, която **обръща** колекция от **цели числа** и **премахва елементите**, които се делят на дадено цяло число **n**. Използвайте **предикати / ламбда функции**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6  2 | 5 3 1 |
| 20 10 40 30 60 50  3 | 50 40 10 20 |

## Предикат за имена

Напишете програма, която филтрира **списък с имена** в зависимост от тяхната **дължина**. На първия ред ще получите **цяло число n**, което представлява **дължината на името**. На втория ред ще получите имена, разделени с интервал. Напишете функция, която отпечатва **само имената**, чиято **дължина** е **по-малка или равна на n**.

Използвайте **предикат** за филтрирането.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Nathaniel Molly Dan Joe Glen | Dan  Joe  Glen |
| 4  Alexia Cara Robin Lynda | Cara |

## Функция за сравнение

Напишете **сравняваща функция**, която се използва за сортиране на **всички четни числа** преди **всички нечетни** числа в **нарастващ ред**. Подайте я на функцията **Array.Sort()** и **отпечатайте** резултата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6 | 2 4 6 1 3 5 |
| -3 2 | 2 -3 |

## Списък с предикати

Намерете всички числа в диапазона **1...N**, които се **делят едновременно** на числата от **дадена поредица**.

На първия ред ще получите **цяло число N**, което е **края на диапазона**. На втория ред ще получите поредица от цели числа, които са **делителите**. Използвайте предикати / функции.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 10  1 1 1 2 | 2 4 6 8 10 |
| 100  2 5 10 20 | 20 40 60 80 100 |
| 50  3 8 | 24 48 |

## Парти от предикати!

Родителите на Карлос са на ваканция за празниците и той планира парти вкъщи. За съжаление, организационните му способности са слаби, така че вие имате задачата да му помогнете с резервациите.

На първия ред ще получите **списък** с **всички имена** на хората, които ще дойдат. На следващите редове, до получаване на команда **"Party!"**, ще трябва да **удвоите** или да **премахнете всички хора**, които отговарят на даден **критерий**. Има **три различни критерия**:

* Имената трябва да **започват** с **дадения стринг**
* Имената трябва да **завършват** с **дадения стринг**
* Имената трябва да имат **дадената дължина**

Накрая **отпечатайте всички гости**, които ще дойдат на партито, разделени с "," и след това **добавете** към края на стринга **ending** "are going to the party!". Ако **няма гости** на партито, отпечатайте "Nobody is going to the party!".

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Paul Alice Hector  Remove StartsWith P  Double Length 5  Party! | Alice, Alice, Hector are going to the party! |
| Peter  Double StartsWith Pete  Double EndsWith eter  Party! | Peter, Peter, Peter, Peter are going to the party! |
| Philip  Remove StartsWith P  Party! | Nobody is going to the party! |

## Модул за филтър за парти

Имплементирайте филтриращ модул за партито. Първо, на този филтър, който ще наречем PRFM (Party Reservation Filter Module), се **подава списък с покани**. След това PRFM получава **поредица от команди**, които конкретизират дали да се добави или премахне даден филтър.

Всяка PRFM команда има следния формат:

"**{command;filter type;filter parameter}"**

Можете да получите следните **PRFM команди**:

* "**Add filter**"
* "**Remove filter**"
* "**Print**"

Възможните типове **PRFM филтри** са:

* "**Starts with**"
* "**Ends with**"
* "**Length**"
* "**Contains**"

Всички параметри за PRFM филтрите са стрингове (или цяло число за филтъра "**Length"**). Всяка команда ще бъде валидна т.е. няма нужда да проверявате филтрите. Входът ще **приключи** с команда "**Print**", след което трябва да отпечатате **всички посетители** на партито, останали **след филтрирането**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Paul Mark Sandy  Add filter;Starts with;P  Add filter;Starts with;M  Print | Sandy |
| Peter Miles Jorge  Add filter;Starts with;P  Add filter;Starts with;M  Remove filter;Starts with;M  Print | Miles Jorge |

Използвайте **списък от предикати**, за да съхранявате филтрите, които ви се подават.

## Калкулатор за събития

Напишете програма, която събира две числа. Ще получите **числата**, разделени с **интервал**. Отпечатайте **сумата** от числата и ако тя е **нечетно число**, активирайте **събитие**, което отпечатва съобщението "\*\*\*\*\*\*\*\*Event Executed : This is Odd Number\*\*\*\*\*\*\*\*\*\*", използвайки **делегати**.

Създайте **публичен** клас AddTwoNumbers, дефинирайте в него **събитие** OddNumberFound и създайте метод Add(num1, num2). Методът трябва да пресмята сумата и да **извика събитието** в случай на нечетна сума (ако не е null). В класа StartUp създайте обект от класа AddTwoNumbers, абонирайте се към събитието и използвайте метода Add().

TODO: да се опише по-добре за този StartUp клас.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3 4 | 7  \*\*\*\*\*\*\*\*Event Executed : This is Odd Number\*\*\*\*\*\*\*\*\*\* |
| 2 2 | 4 |

### Насоки

Graphical user interface, text, application

Description automatically generated

Text

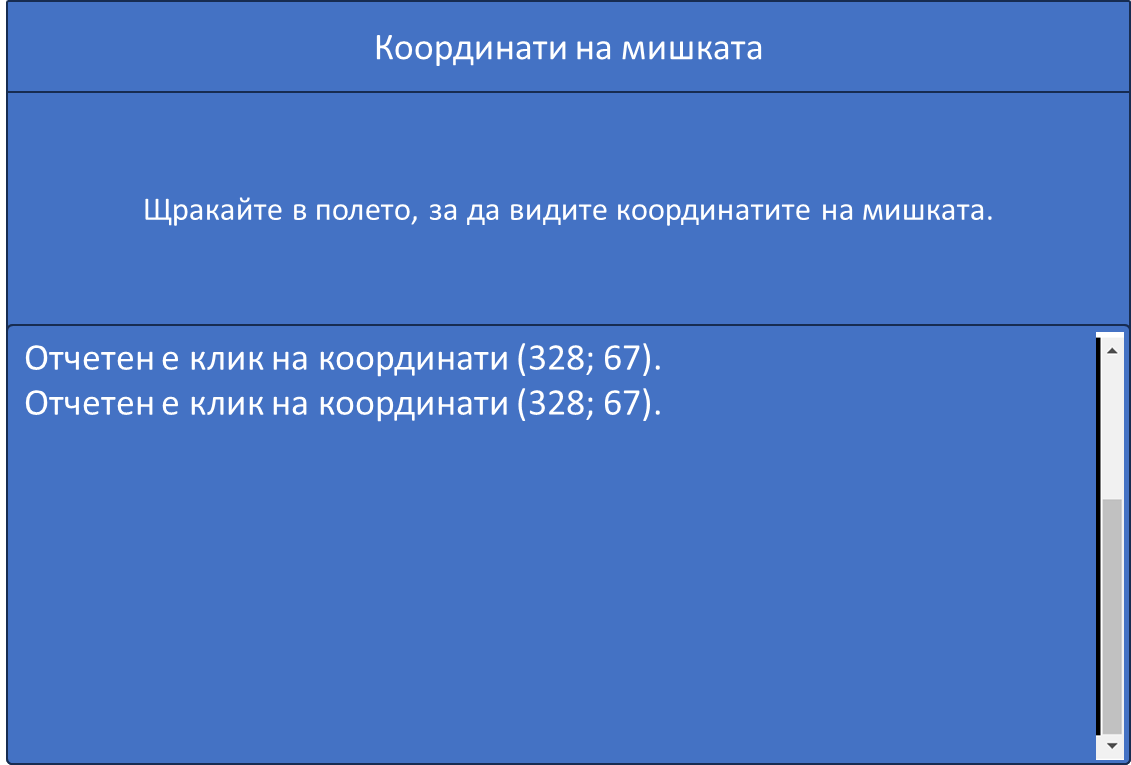
Description automatically generated

## Координати на мишката

Важно: Тази задача не може да бъде тествана в Judge!

Направете Windows Form приложение, което прихваща щракване от мишката и отпечатва кооедринатите, където се е случило щракването:

TODO: screenshot



TODO: инструкции

1. Направете Windows Forms app
2. Сложете TextBoxClickLog (multi line)
3. Хванете MainForm.Click
4. При клик добавете текст в TextBoxClickLog