**Village**

**Преглед на задачата**

Като Junior C# разработчици получавате първата си  обучителна задача във фирмата. Трябва да създадете програма, която пресъздава развитието на едно селце. Основните функционалности са да се развива и да може да бъде нападано от бунтовници. Трябва да приложите знанията си по ООП, за да завършите тази задача. Успех!!!

**Забележка**: Преди да започнете със задачата, за да нямате проблеми със системата премахнете **namespaces** от всеки клас.

* **Структура**

**Person**

Класът **Person** е **базов** **клас** и **не** **трябва** да може да се **инстанцира**.

**Data**

Всеки човек има име и години.

**Name** – символен низ

**Age**– цяло число

* Името на човека **трябва** да бъде между 3 и 30 символа включително**.**

**Съобщение:** **Name should be between 3 and 30 characters!**

* Годините на човека **не** **може** да бъдат **отрицателни.**

**Съобщение:** **Age should be 0 or positive!**

* При подаване на **невалидна** стойност хвърлете изключение – **ArgumentException с конкретното съобщение**.

**Constructor**

Това са параметрите за конструктора:

string name, int age

**Behavior**

**override string ToString()**

**Name: {Name}**

**Age: {Age}**

**Child Classes**

**Peasant**

**Data**

Всеки селянин има име, години и продуктивност.

**Name** – символен низ

**Age**– цяло число

**Productivity**– цяло число

* Годините **не трябва** да бъдат повече от **110.**

**Съобщение:** **Age cannot be greater than 110!**

* Продуктивността **не** **може** да бъде **отрицателна или 0.**

**Съобщение:** **Productivity cannot be less or equal to 0!**

* При подаване на **невалидна** стойност хвърлете изключение – **ArgumentException с конкретното съобщение**.

**Забележка**: Първо изпълнете валидацията от базовия клас след това текущата.

**Constructor**

Това са параметрите за конструктора:

string name, int age, int productivity

**Behavior**

**override string ToString()**

**Name: {Name}**

**Age: {Age}**

**Productivity: {Productivity}**

**Rebel**

**Data**

Всеки бунтовник има име, години и щета.

**Name** – символен низ

**Age**– цяло число

**Harm**– цяло число

* Годините **не трябва** да бъдат повече от **50.**

**Съобщение:** **Age should be less or equal to 50!**

* Щетата **не** **може** да бъде **отрицателна или 0.**

**Съобщение:** **Harm should be greater than 0!**

* При подаване на **невалидна** стойност хвърлете изключение – **ArgumentException с конкретното съобщение**.

**Забележка**: Първо изпълнете валидацията от базовия клас след това текущата.

**Constructor**

Това са параметрите за конструктора:

string name, int age, int harm

**Behavior**

**override string ToString()**

**Name: {Name}**

**Age: {Age}**

**Harm: {Harm}**

**Village**

**Data**

Всяко селце има име, локация, ресурси и **списък**със селяни:

**Name**– символен низ

**Location**– символен низ

**Resource**– цяло число

* Името **трябва** да бъде **между 2 и 40 символа включително.**

**Съобщение:** **Name should be between 2 and 40 characters!**

* Локацията **трябва** да бъде **между 3 и 45 символа включително.**

**Съобщение:** **Location should be between 3 and 45 characters!**

* При подаване на **невалидна** стойност хвърлете изключение – **ArgumentException с конкретното съобщение**.

**Забележка**: Не правете **СВОЙСТВО**за списъка. **Ще бъде проверен!!!**

**Constructor**

Това са параметрите за конструктора:

string name, string location

**Behavior**

**void AddPeasant(Peasant peasant)**

В метода трябва да добавите новия заселник.

**int PassDay()**

В метода трябва да сумирате продуктивността на всички заселници и да ги добавите към ресурсите на селцето. Върнете произведените ресурси за деня.

**List<Peasant> KillPeasants(int count)**

В метода трябва да премахнете първите N на брой заселника и да ги върнете като резултат. В случай, че няма достатъчно хора, върнете всички.

**override string ToString()**

**Village - {Name} ({Location})**

**Resources - {Resource}**

**Peasants – ({peasant count})**

**{peasant1.ToString()}**

**{peasant2.ToString()}**

**…**

**{peasantN.ToString()}**

**Забележка:** Уверете се, че за всеки от методите задавате правилен модификатор за достъп.

* **Бизнес логика**

**The Controller Class**

Бизнес логиката на програмата трябва да бъде концентрирана около няколко команди. Внедрете клас, наречен **Controller**, който ще реализира главната функционалност, представена от тези публични методи:

|  |
| --- |
| **Controller.cs** |
| public string ProcessVillageCommand(List<string> args)  {      //TODO: Add some logic here …  }    public string ProcessSettleCommand(List<string> args)  {      //TODO: Add some logic here …  }    public string ProcessRebelCommand(List<string> args)  {      //TODO: Add some logic here …  }    public string ProcessDayCommand(List<string> args)  {      //TODO: Add some logic here …  }    public string ProcessAttackCommand(List<string> args)  {      //TODO: Add some logic here …  }  public string ProcessInfoCommand(List<string> args)  {      //TODO: Add some logic here …  }    public string ProcessEndCommand()  {      //TODO: Add some logic here …  } |

**ЗАБЕЛЕЖКА: Не трябва да променяте нищо по методите. Трябва да имплементирате логиката на самите методи. Не прихвайщайте никакви изключения!**

**Команди**

Има няколко команди, които контролират бизнес логиката на приложението и трябва да ги имплементирате.

Те са посочени по-долу. За всяка от командите връщайте текстов низ в съответствие с указанията в частта „Изход“ на задачата.

**Village Команда**

Създава **Village**и го добавя към системата.

**Не всички данни ще бъдат валидни!!!**

**Параметри**

* **name** – символен низ (винаги ще бъде уникално)
* **location** – символен низ

**Settle Команда**

Създава **Peasant**и го добавя към **Village**-а с **подаденото име**.

**Не всички данни ще бъдат валидни!!!**

**Параметри**

* **name** – символен низ
* **age** – цяло число
* **productivity** – цяло число
* **villageName** – символен низ (**винаги валидно** и **съществуващо**)

**Rebel Команда**

Създава **Rebel**и го добавя към **системата**.

**Не всички данни ще бъдат валидни!!!**

**Параметри**

* **name** – символен низ
* **age** – цяло число
* **harm** – цяло число

**Day Команда**

Намира **Village**-а с подаденото име от системата и извиква **PassDay()** метода му.

**Параметри**

* **villageName** – символен низ

**Attack Команда**

Взима **първите** **N** **бунтовника** (**Rebel**), където N се равнява на **първия параметър** от командата (**rebelCount**). След като ги вземе, сумира тяхната вреда (**Harm**) и намира **Village**-а с подаденото име от системата. След това **намалява ресурсите** на **Village**-а със **сумираната вреда** (**Harm**) на бунтовниците. Също така, всеки **2-ма** бунтовника (**Rebel**), **убиват** по **1** **селянин**(**Peasant**), което означава, че трябва да се извика, на намерения **Village**, **KillPeasants()** метода с  
**броя**на**бунтовниците / 2**.

Ако **няма никакви бунтовници**в системата, тук не би трябвало да се случи нищо.

**Параметри**

* **rebelCount** – цяло число
* **villageName** – символен низ

**Info Команда**

Ако **side** параметъра се равнява на "**Rebel**", трябва да се изведе информация за всички бунтовници (**Rebel**).

Ако **side** параметъра се равнява на "**Village**", трябва да се изведе информация за всички села (**Village**).

**Параметри**

* **side** – символен низ

**End Команда**

Тази команда прекратява изпълнението на програмата и връща **общия брой** на селата (**Village**) в системата, **общия брой** на бунтовниците (**Rebel**) в системата и **общия брой** на **атаките** (**Attack command**), които са се случили до момента.

* **Вход / Изход**

Преди да продължите с тази секция от изпита, трябва да разясним, че **Startup** и **Engine** класовете не се тестват по никакъв начин. Задачата ви е да довършите приложението, така че да заработи. Няма правила какво трябва да бъде изписано в изброените по-горе класове.

**Вход**

* Четете редове с различни команди, докато не получите команда за приключване на програмата.

По-долу можете да видите формата, в който всяка команда ще бъде дадена във входа:

* **Village {name} {location}**
* **Settle {name} {age} {productivity} {villageName}**
* **Rebel {name} {age} {harm}**
* **Day {villageName}**
* **Attack {rebelCount} {villageName}**
* **Info {side}**
* **End**

**Изход**

По – долу може да видите кой изход трябва да бъде предоставен от командите.

**Village Команда**

При успешно добавяне върнете:

**Съобщение**: **Created Village {villageName}!**

**Settle Команда**

При успешно добавяне върнете:

**Съобщение**: **Settled Peasant {peasantName} in Village {villageName}!**

**Rebel Команда**

При успешно добавяне върнете:

**Съобщение**: **Rebel {rebelName} joined the gang!**

**Day Команда**

При изпълнение върнете:

**Съобщение**: **Village {villageName} resource increased with {dailyResource}!**

**Attack Команда**

При успешно изпълнение върнете:

**Съобщение**: **Village {villageName} lost {takenResource} resources and {deadPeasants} peasants!**

**Забележка**: **takenResources** се **равнява** на **взетите ресурси** от **бунтовниците** (тоест **сумираната** **вреда** на бунтовниците (**Harm**)), а **deadPeasants** на **убитите селяни**.

В случай, че няма никакви бунтовници в системата, върнете:  
**Съобщение**: **No rebels to perform raid...**

**Info Команда**

**Rebel**

При изпълнение с параметър **side** = "**Rebel**" върнете:

**Съобщение**:

**Rebels:  
Name: {Rebel1.Name}  
Age: {Rebel1.Age}  
Harm: {Rebel1.Harm}  
Name: {Rebel2.Name}  
Age: {Rebel2.Age}  
Harm: {Rebel2.Harm}  
...**

В случай, че няма бунтовници в системата, изведете:

**Съобщение**: **No Rebels**

**Village**

При изпълнение с параметър **side** = "**Village**" върнете:

**Съобщение**:

**Villages:  
Village - {Village1.Name} ({Village1.Location})  
Resources: {Village1.Resource}  
Peasants: ({Village1.PeasantsCount})**

**Name: {Peasant1.Name}**

**Age: {Peasant1.Age}**

**Productivity: {Peasant1.Productivity}**

**Name: {Peasant2.Name}**

**Age: {Peasant2.Age}**

**...**

**Village - {Village2.Name} ({Village2.Location})  
Resources: {Village2.Resource}  
Peasants: ({Village2.PeasantsCount})  
...**

В случай, че няма селца в системата, изведете: **Съобщение**: **No Villages**

**End Команда**

При изпълнение върнете:

**Съобщение**:

**Villages: {totalVillagesCount}  
Rebels: {totalRebelsCount}  
Attacks on Villages: {totalAttacksPerformed}**

**Ограничения**

* Имената на селата, селяните и бунтовниците ще бъдат символни низове, които може да съдържат всеки ASCII символ, с изключение на двоеточие(:).
* Винаги ще получавате команда за приключване на програмата.
* Входните данни ще бъдат валидни от страна на типове данни.
* Имената на селата винаги ще са уникални
* В командите в които се търси село по подадено име, винаги ще се подава име на съществуващо село.

**Примери**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Village Rashford Downshire  Village Silkhaven Uphills  Settle Joseph 37 50 Rashford  Settle Jack 25 100 Rashford  Settle Michaila 25 35 Rashford  Settle Teresa 23 110 Rashford  Settle Igor 40 150 Silkhaven  Rebel Donald 30 25  Day Rashford  Day Rashford  Day Silkhaven  Attack 1 Silkhaven  Attack 1 Silkhaven  End | Created Village Rashford!  Created Village Silkhaven!  Settled Peasant Joseph in Village Rashford!  Settled Peasant Jack in Village Rashford!  Settled Peasant Michaila in Village Rashford!  Settled Peasant Teresa in Village Rashford!  Settled Peasant Igor in Village Silkhaven!  Rebel Donald joined the gang!  Village Rashford resource increased with 295!  Village Rashford resource increased with 295!  Village Silkhaven resource increased with 150!  Village Silkhaven lost 25 resources and 0 peasants!  Village Silkhaven lost 25 resources and 0 peasants!  Villages: 2  Rebels: 1  Attacks on Villages: 2 |
| Info Rebel  Info Village  Village Rashford Downshire  Settle Joseph 37 50 Rashford  Settle Jack 25 100 Rashford  Settle Michaila 25 35 Rashford  Settle Teresa 23 110 Rashford  Settle Igor 40 150 Rashford  Attack 1 Rashford  Rebel Donald 30 25  Rebel Zack 31 50  Day Rashford  Day Rashford  Attack 1 Rashford  Attack 2 Rashford  Info Rebel  Info Village  End | No Rebels  No Villages  Created Village Rashford!  Settled Peasant Joseph in Village Rashford!  Settled Peasant Jack in Village Rashford!  Settled Peasant Michaila in Village Rashford!  Settled Peasant Teresa in Village Rashford!  Settled Peasant Igor in Village Rashford!  No rebels to perform raid...  Rebel Donald joined the gang!  Rebel Zack joined the gang!  Village Rashford resource increased with 445!  Village Rashford resource increased with 445!  Village Rashford lost 25 resources and 0 peasants!  Village Rashford lost 75 resources and 1 peasants!  Rebels:  Name: Donald  Age: 30  Harm: 25  Name: Zack  Age: 31  Harm: 50  Villages:  Village - Rashford (Downshire)  Resources - 790  Peasants - (4)  Name: Jack  Age: 25  Productivity: 100  Name: Michaila  Age: 25  Productivity: 35  Name: Teresa  Age: 23  Productivity: 110  Name: Igor  Age: 40  Productivity: 150  Villages: 1  Rebels: 2  Attacks on Villages: 2 |

* **Точкуване**

Всяка различна задача Ви дава точки:

* Структура - 30 точки
* Бизнес логика - 50 точки
* Вход/Изход - 20 точки