**Фитнес**

**Общ преглед**

Във вашата фирма постъпва проект за създаване на приложение, обслужващо фитнес.

Вашият софтуер трябва да описва **спортисти** (Athlete) и **треньори (**Coach**).**

Tрябва да реализирате функционалност, която да позволява добавяне на **спортисти** и **треньори**, а в последствие и бизнес логика включваща трениворки, генериране на статистики и други – всичко това ще работи чрез **команди**, които вие ще получавате. Поредицата от команди приключва с „**End**”. За ваше удобство ще получите готов Program.cs файл, а вие ще трябва да реализирате само необходимите класове в Athlete.cs и Coach.cs

**Подзадача 1: Структура на Спортист – 15 точки**

**Athlete**

Всички спортисти имат име, възраст и сила:

name – низ, съставен от малки и големи латински букви, без други специални знаци

                Уникална стойност, която различава всички спортисти

**age** – цяло число

**strength** – цяло число

Реализирайте конструктор:

* public Athlete(string name, int age, int strength)– този конструктор трябва да приема името, възрастта и силата на спортиста.

За справка вижте следната схема за Athlete.cs:

|  |
| --- |
| Athlete.cs |
| public Athlete(string name, int age, int strength)  {      //TODO: Добавете вашия код тук …  } |

**Команди за добавяне на спортисти**

Трябва да довършите следната команда за добавяне на спортисти:

* **AddAthlete** <**име**> <**възраст**> <**сила**> - тази команда има за цел да добави спортист с неговите име, възраст и сила. Командата е описана в Main метода.

**За да сработи, командата, се нуждае от горепосочения конструктор в класа Athlete!**

**Команда за извеждане на информация**

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за спортист. Командата за това е следната:

* **PrintAthleteInfoByName** <**име**> - отпечатва информация за спортист във формат:  
  **Athlete: name - <име>, age - <възраст> strength <сила>.**

Тази команда ще получава винаги валидни и съществуващи имена на спортисти. За успешна реализация трябва да реализирате ваша версия на **ToString()** метода за класа **Athlete.**

**За справка виж нулевия тест!**

**Подзадача 2: Структура на Треньор – 15 точки**

**Coach**

Всеки треньор има: име, опит, списък от клиенти и пари

name – низ, съставен от малки и големи латински букви, без други специални знаци.

Уникална стойност, която различава всички треньори

**experience** – цяло число

**clients** – списък от **Athlete**

**money** – число с плаваща запетая

Трябва да реализирате **два** конструктора:

* public Coach(string name, int experience) – този конструктор трябва да приема име, опит и да постави дефолтна стойност в променливата **money** да бъде 0.
* public Coach(string name, int experience, double money) – този конструктор трябва да приема име, опит и пари

 За справка вижте следната схема за Coach.cs:

**Погрижете се за валидните състояния на всички полета в класа!**

|  |
| --- |
| Coach.cs |
| public Coach(string name, int experience) {        //TODO: Добавете вашия код тук …    }      public Coach(string name, int experience, double money) {        //TODO: Добавете вашия код тук …    } |

**Команда за добавяне на треньор**

* **AddCoach** <**name**> <**experience**>- тази команда има за цел да добави нов треньор.
* **AddCoach** <**name**> <**experience**> **<money>**- тази команда има за цел да добави нов треньор.

**Командата е описана в Main метода. За да сработи се нуждае от горепосочените конструктори в класа Coach!**

**Команда за извеждане на информация**

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за треньор. Командата за това е следната:

* **PrintCoachInfoByName <име> -** отпечатва информация за треньор във формат:  
  Coach: **<име>** has experience **<опит>** and his current money balace is: **<пари>**

**Парите следва да бъдат форматирани до 2-ри знак след десетичния разделител!**

Тази команда ще получава винаги валидни и съществуващи имена на треньори. За успешна реализация трябва да реализирате ваша версия на **ToString()** метода за класа **Coach.** Очаква се да заместите стойностите подадени в диамантени скоби с реални такива.

**Подзадача 3: Логика – 40 точки**

**Kоманди**

Вашето приложение трябва да реализира следните команди:

* **AddAthleteToSpecificCoach <име на спортист> <име на треньор> -** Тази команда добавя дадения спортист към списъка от **клиенти** на треньора. *Командата използва методът***void AddNewClient(Athlete a)** *от класа Coach.cs*

* **RemoveAthleteFromSpecificCoach <име на спортист> <име на треньор> -** Тази команда премахва клиент (спортист) от списъка с клиенти на треньора. *Командата използва* **bool LooseClient(string name)** *от класа Coach.cs.* **Методът трябва да връща true стойност, ако успее да изтрие клиент и false, ако не е същестувал такъв спортист в списъка с клиенти на треньора.**

* **GetClientsCount <име на треньор> -** Тази команда следва да преброи всички клиенти. *Командата използва методът* **int GetClientsCount()** *от класа Coach.cs.*

**Методът трябва да връща целочислена стойност, което представлява броя на всички спортисти в  списъка от клиенти на треньора.**

* **GetAllClientsTougherThan <име на треньор> <сила> -** Тази команда следва да върне всички клиенти, които имат по-голяма стойност в полето **strength** от подадената на метода. *Командата използва методът* **List<Athlete> GetAllClientsTougherThan(int strength)** *от класа Coach.cs.*

**Методът трябва да връща списък от Athlete, който представлява част от всички клиенти на треньора**

* **SpendMoneyOnSupplements <име на треньор> <сума> -** Тази команда следва да намали стойността в полето **money** с подадената стойност на метода. *Командата използва методът*

**void SpendMoneyOnSupplements(double amount)** *от класа Coach.cs.*

* **TrainHard <име на треньор>** - Тази команда има за цел да повиши „опита“ на треньора. Стойността в полето **experience** следва **да бъде повишена с 5 единици**! *Командата използва методът* **void TrainHard()** *от класа Coach.cs.*

**Подзадача 4: Бонус логика – 20 точки**

Трябва да реализирате още няколко команди:

* **TrainAthlete** <**име на спортист**> <**име на треньор**> **- *Командата използва методът***

**void TrainClient(Athlete athlete)** *от класа Coach.cs****.***  Методът следва да изпълни логика обсужваща една тренировка на клиент.

1. **1.** Спортистът, който бива трениран получава увеличение на стойността в полето **strength** с **5 единици.**
2. **2.** Опитът на треньора се увеличава, съответно в полето **experience** добавете една единица.
3. **3.** Тъй като тренировките не са безплатни, треньорът получава сума, която трябва да прибави към текущата стойност на полето **money.** Сумата, която трябва да **прибави** се изчислява по формулата: **Athlete.Strength \* 0.75.**

**Важно: Спазете поредността на стъпките, описани по-горе!**

* **RemoveAllClientsLeakerThan** <**име на треньор**> **<сила> -** *Командата използва методът*  **int RemoveAllClientsLeakerThan(int strength)** *от класа Coach.cs.* Тази команда следва да премахне всички клиенти, които имат по-ниска стойност в полето **strength** от подадената на метода. Трябва да върнете целочислена стойност, равняваща се на броя премахнати клиенти от списъка.

* **GetToughestClient** <**име на треньор**> **- *Командата използва методът*** Athlete GetToughestClient() ***от класа Coach.cs.*** Методът трябва върне спортистът с най-висока стойност в полето **strength**. Не се очаква да има двама или повече спортисти с една и съща стойност в съответното поле.

**За безпроблемната работа на всички изброени команди от 3 и 4 подзадача трябва да реализирате Athlete.cs и Coach.cs, по аналогичен начин на показаното по-долу:**

|  |
| --- |
| Coach.cs |
| public Coach(string name, int experience)  {      //TODO: Добавете вашия код тук …  }    public Coach(string name, int experience, double money)  {      //TODO: Добавете вашия код тук …  }    public string Name  {      //TODO: Добавете вашия код тук …  }    public int Experience  {      //TODO: Добавете вашия код тук …  }    public double Money  {      //TODO: Добавете вашия код тук …  }    public void AddNewClient(Athlete a)  {      //TODO: Добавете вашия код тук …  }    public bool LooseClient(string name)  {      //TODO: Добавете вашия код тук …  }    public int GetClientsCount()  {      //TODO: Добавете вашия код тук …  }    public List<Athlete> GetAllClientsTougherThan(double strength)  {      //TODO: Добавете вашия код тук …  }  public void SpendMoneyOnSupplements(double amount)  {      //TODO: Добавете вашия код тук …  }    public void TrainHard()  {      //TODO: Добавете вашия код тук …  }    public void TrainClient(Athlete athlete)  {      //TODO: Добавете вашия код тук …  }    public int RemoveAllClientsLeakerThan(int strength)  {      //TODO: Добавете вашия код тук …  }    public Athlete GetToughestClient()  {      //TODO: Добавете вашия код тук …  }    public override string ToString()  {       //TODO: Добавете вашия код тук …  } |

|  |
| --- |
| Athlete.cs |
| public Athlete(string name, int age, int strength)  {  //TODO: Добавете вашия код тук …  }    public string Name  {       //TODO: Добавете вашия код тук …  }    public int Age  {       //TODO: Добавете вашия код тук …  }    public int Strength  {       //TODO: Добавете вашия код тук …  }    public override string ToString()  {       //TODO: Добавете вашия код тук …} |

**Забележка:** Освен горепосочените методи трябва да реализирате и необходимите свойства за всеки от класовете. Възможно е, да е удачно, да реализирате допълнителни полета, свойства и методи, по ваша преценка.

**Подзадача 5: Валидация – 10 точки**

Освен всичко останало вие трябва да направите и валидация!

При невалидни данни хвърлете грешка от тип **ArgumentException(“message”)**

* Методът **SpendMoneyOnSupplements** не бива да допуска харчене на сума **по-голяма** от текущата стойност в полето **money**, message => **"Insufficient balance"**
* Не създавайте треньор с **дължина на името** по-малка или равна на 2 символа, message => **"Invalid coach name!"**
  + Програмата ще получава множество редове с информация. Всеки ред представлява команда. Самият вход се обработва изцяло от примерния Program.cs.
  + Всички команди приключват с въвеждането на End

**Вход / Изход**

**Вход**

**Изход**

За някои от командите не е нужно да извеждате нищо. За всички останали изпечатването ще ви бъде дадено в Program.cs, освен ToString() методите, които са оставени на вас.

**Ограничения**

* Всички цели числа ще бъдат в диапазона **–10000** до **+10000**
* Всички числа с плаваща запетая ще бъдат въвеждани с до **15** знака след запетаята.
* Имената няма да съдържат интервал

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| AddAthlete Pesho 20 3 |  |
| AddAthlete Martin 1 1234 |  |
| PrintAthleteInfoByName Pesho | Athlete: name - Pesho, age - 20, strength 3. |
| AddCoach Mario 50 |  |
| AddCoach Ivan 20 100.15 |  |
| AddAthleteToSpecificCoach Pesho Ivan |  |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 20 and his current money balace is: 100.15 |
| AddAthleteToSpecificCoach Martin Ivan |  |
| RemoveAthleteFromSpecificCoach Pesho Ivan | Successfully removed athlete: Pesho from coach Ivan! |
| GetClientsCount Ivan | Coach Ivan has total count of clients: 1. |
| AddAthlete Pesho2 20 51 |  |
| AddAthleteToSpecificCoach Pesho1 Ivan |  |
| AddAthleteToSpecificCoach Pesho2 Ivan |  |
| GetAllClientsTougherThan Ivan 50 | Tough Guys:  0: Athlete: name - Martin, age - 1, strength 1234.  1: Athlete: name - Pesho2, age - 20, strength 51. |
| SpendMoneyOnSupplements Ivan 80 | Coach Ivan spent on supplements 80.00 |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 20 and his current money balace is: 20.15 |
| TrainHard Ivan |  |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 25 and his current money balace is: 20.15 |
| TrainAthlete Pesho1 Ivan |  |
| TrainAthlete Pesho1 Ivan |  |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 27 and his current money balace is: 106.40 |
| RemoveAthletesLeakerThan Ivan 52 | Coach Ivan removed 1 loosers. |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 27 and his current money balace is: 106.40 |
| GetToughestAthlete Ivan | Toughest client of coach Ivan is Athlete: name - Martin, age - 1, strength 1234. |
| SpendMoneyOnSupplements Ivan 5555 | Insufficient balance |
| PrintCoachInfoByName Ivan | Coach: Ivan has experience 27 and his current money balace is: 106.40 |
| AddCoach I 15 500.53 | Invalid coach name! |
| End |  |