**JobBoard**

**Available from**: Sunday, 30 October 2022, 10:00  
 **Requested files**: Program.cs, JobOffer.cs, OnSiteJobOffer.cs, RemoteJobOffer.cs, Category.cs, Controller.cs ( [Download](https://it-kariera.mon.bg/e-learning/mod/vpl/views/downloadrequiredfiles.php?id=244))  
**Type of work**:  Individual work  
 **Настройки на оценките**: Максимална оценка: 100  
 **Run**: Не.  **Evaluate**: Да. **Evaluate just on submission**: Да  
**Automatic grade**: Да.

**JobBoard**

Намирането на работа е доста важна задача за всеки човек. Днес вие ще направите софтуер, който да помага именно в тази нелека задача. Основните му функционалности са свързани със създаване на обяви за работа и търсене на обяви за работа по даден критерий.

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

**Структура**

**JobOffer**

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

**Данни**

Всяка оферта има име на позиция, име на работодател и заплата.

JobTitle – символен низ

Company – символен низ

Salary – реално число

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

Съобщение: JobTitle should be between 3 and 30 characters!

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

Съобщение: Company should be between 3 and 30 characters!

* Заплатата не може да бъде отрицателна.

Съобщение: Salary should be 0 or positive!

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

**Конструктор**

Това са параметрите за конструктора:

string jobTitle, string company, double salary

**Методи за класа JobOffer**

override string ToString()

Job Title: {JobTitle}

Company: {Company}

Salary: {Salary} BGN

Заплатата трябва да се извежда с точност 2 знака след запетаята.

**Child Classes**

**OnSiteJobOffer**

**Данни**

Всяка присъствена оферта за работа има адрес на месторабота

City – символен низ

Съобщение: City should be between 3 and 30 characters!

* Името на града трябва да бъде между 3 и 30 символа включително.
* При подаване на невалидна стойност хвърлете изключение – ArgumentException с конкретното съобщение.

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

**Конструктор**

Това са параметрите за конструктора:

string jobTitle, string company, double salary, string city

**Методи за класа OnSiteJobOffer**

override string ToString()

Job Title: {JobTitle}

Company: {Company}

Salary: {Salary} BGN

City: {City}

**RemoteJobOffer**

**Данни**

**Забележка: Офертите за дистанционна работа могат да бъдат и частични - т.е. Част от времето се работи присъствено, част дистанционно.**

FullyRemote – булева стойност

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

**Конструктор**

Това са параметрите за конструктора:

string jobTitle, string company, double salary, bool fullyRemote

**Методи за класа RemoteJobOffer**

override string ToString()

Job Title: {JobTitle}

Company: {Company}

Salary: {Salary} BGN

Fully Remote: {yes|no}

**Забележка: Ако FullyRemote е True, то в ToString() трябва да се изведе “yes” за стойност и “no”, в противен случай. Обърнете внимание на примерния вход и изход.**

**Category**

**Данни**

Всяка категория има име и списък с обяви в нея:

Name – символен низ

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

Съобщение: Name should be between 2 and 40 characters!

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

**Конструктор**

Това са параметрите за конструктора:

string name

**Методи за класа Category**

**void AddJobOffer(JobOffer offer)**

В метода трябва да добавите новата оферта към категорията.

**double AverageSalary()**

В метода трябва да намерите средната заплата от всички оферти в категорията и да я върнете.

**List<JobOffer> GetOffersAboveSalary(double salary)**

В метода трябва да върнете тези оферти, които са със заплата по-голяма или равна на подадената като параметър стойност. Сортирайте тези оферти в низходящ ред по тяхната обявена заплата.

**List<JobOffer> GetOffersWithoutSalary()**

В метода трябва да върнете тези оферти, които са без обявена заплата. За такива оферти се считат всички оферти, чиято стойност за Salary e 0. Подредете офертите по име на компанията в азбучен ред.

**override string ToString()**

Category {Name}

Total Offers: {Брой оферти}

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

**Бизнес логика**

Управлението на програмата се случва благодарение на клас, който се казва Controller. Той ще обработва команди, които отговарят на нужната функционалност, представена от тези публични методи:

|  |
| --- |
| Controller.cs |
| public string AddCategory(List<string> args)  {  //TODO: Add some logic here …  }    public string AddJobOffer(List<string> args)  {  //TODO: Add some logic here …  }    public string GetAverageSalary(List<string> args)  {  //TODO: Add some logic here …  }    public string GetOffersAboveSalary(List<string> args)  {  //TODO: Add some logic here …  }    public string GetOffersWithoutSalary(List<string> args)  {  //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.

**AddCategory Команда**

Създава категория (Category) и я добавя към системата.

Не всички данни ще бъдат валидни!!!

**Параметри**

* name – символен низ (винаги ще бъде уникално)

**AddJobOffer Команда**

Създава обява за работа (JobOffer) и я добавя към категорията (Category) с подаденото име.

Не всички данни ще бъдат валидни!!!

**Параметри**

* name – символен низ, име на категорията
* jobTitle – символен низ
* company – символен низ
* salary – реално число
* type – символен низ
  + Ако type e “onsite”, то това е оферта за работа на място, тогава ще се приеме още един параметър:
    - city – символен низ
  + Ако type e “remote”, то това е оферта за дистанционна работа, тогава ще се приеме още един параметър:
    - fullyRemote – булева стойност

**GetAverageSalary Команда**

При нейното извикване се изчислява средната заплата в дадена категория и се извежда съобщение:

"The average salary is: {averageSalary} BGN"

**Параметри**

* name – символен низ

Ако няма оферти, да се приеме, че средната заплата е 0. В изчислението участват всички оферти.

**GetOffersAboveSalary Команда**

Намира категорията с подаденото име от системата и извиква GetOffersAboveSalary() метода ѝ.

**Параметри**

* name – символен низ
* salary – реално число

**GetOffersWithoutSalary Команда**

Намира категорията с подаденото име от системата и извиква GetOffersWithoutSalary() метода ѝ.

**Параметри**

* name – символен низ

**End Команда**

Тази команда прекратява изпълнението на програмата.

**Вход / Изход**

**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:

* **AddCategory {name}**
* **AddJobOffer {name} {jobTitle} {company} {salary} {type} {city или fullyRemote}**
* **GetAverageSalary {name}**
* **GetOffersAboveSalary {name} {salary}**
* **GetOffersWithoutSalary {name}**
* **End**

**Изход**

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

**AddCategory Команда**

При успех върнете:

Съобщение: Created Category {name}!

**AddJobOffer Команда**

При успех върнете:

Съобщение: Created JobOffer {jobTitle} in Category {name}!

**GetAverageSalary Команда**

При успех върнете:

Съобщение: The average salary is: {averageSalary} BGN

**GetOffersAboveSalary Команда**

Изведете информация за всички оферти с дадената или по-висока заплата от дадена категория. Използвайте ToString() метода за съответния клас-наследник на JobOffer.

**GetOffersWithoutSalary Команда**

Изведете информация за всички оферти с дадената или по-висока заплата от дадена категория. Използвайте ToString() метода за съответния клас-наследник на JobOffer.

**End Команда**

При изпълнение програмата приключва своята работа.

**Всички заплати трябва да се извеждат с точност 2 цифри след запетаята.**

**Ограничения**

* Имената в програмата ще бъдат символни низове, които може да съдържат всеки ASCII символ, с изключение на интервал.
* Винаги ще получавате команда за приключване на програмата.
* Входните данни ще бъдат валидни от страна на типове данни.
* Имената на категориите винаги ще са уникални
* В командите в които се търси категория по подадено име, винаги ще се подава име на съществуваща категория.

**Примери**

|  |  |
| --- | --- |
| Input | Output |
| AddCategory Restaurants AddCategory It AddJobOffer Restaurants Waiter Romance 1000 onsite Sofia GetAverageSalary It AddJobOffer It TechSupport MyTech 1500 remote True AddJobOffer It OnSiteTechSupport MyTech 1700 onsite Burgas AddJobOffer It Tester YourTest 2500 onsite Varna GetAverageSalary It GetOffersAboveSalary It 1600 End | Created Category Restaurants! Created Category It! Created JobOffer Waiter in Category Restaurants! The average salary is: 0.00 BGN Created JobOffer TechSupport in Category It! Created JobOffer OnSiteTechSupport in Category It! Created JobOffer Tester in Category It! The average salary is: 1900.00 BGN Job Title: Tester Company: YourTest Salary: 2500.00 BGN City: Varna Job Title: OnSiteTechSupport Company: MyTech Salary: 1700.00 BGN City: Burgas |

**Точкуване**

Всяка различна задача Ви дава точки:

* Структура - 30 точки
  + За успешната реализация тук трябва да имате работещи команди за създаване на оферта и на категория, като реализирате и съответните им ToString() методи
* Бизнес логика - 50 точки
* Вход/Изход и валидиране - 20 точки