*Данное задание разработано для подписчиков ютюб канала*

[*https://www.youtube.com/c/SBeregovoyRU/*](https://www.youtube.com/c/SBeregovoyRU/)

*Спасибо всем кто заинтересовался и собирается выполнить данный проект.*

# Приложение - Расчет зарплаты ООО “Разработка софта”

# Цель:

Ваша задача разработать приложение, которое позволит вести учет отработанного времени сотрудников и рассчитывать зарплату. Приложение должно уметь строить отчет по времени и зарплате по всем сотрудникам.

# Требования:

1. Для хранения информации следует использовать текстовые файлы. Базы данных использовать нельзя.
2. Должно быть поддержка трех видов ролей: руководитель, сотрудник на зарплате, внештатный сотрудник (фрилансер). При этом авторизация отсутствует т..е при входе в приложение определение роли сотрудника основано на честном выборе. Как он себя называет - тем и является. Мы доверяем его выбору.
3. Интерфейс должен быть построен на базе консольного приложения
4. Код должен быть покрыт юнит тестами. Рекомендуемый процент покрытия кода не менее 50%. Т.е. если у вас есть 10 методов в коде - должно быть хотя бы 5 тестов на эти методы. Если вам совсем не знакома тема юнит тестирования - это требование можно игнорировать
5. Рекомендуется использовать систему контроля версий GIT и фиксировать промежуточные результаты по мере разработки приложения. Комментарии к коммитам нужно давать осмысленные. Подробнее про то как использовать GIT и Visual Studio можно посмотреть тут: <https://youtu.be/WgEeF-XAJZA>
6. Ввод данных должен проверяться на корректность т.е. очевидно не существующие в реальном мире данные не могут быть сохранены в систему. В таких случаях следует выдавать предупреждения.

# Роли и особенности функционала

## Руководитель

1. Может добавлять сотрудников с ролью руководитель, сотрудник на зарплате, внештатный сотрудник
2. Может добавлять время любому сотруднику. Возможно добавление времени задним числом.
3. Может просматривать отчет за выбранный период по всем сотрудникам (периоды могут быть за день, за неделю, за месяц)
4. Может просматривать часы работы за выбранный период по конкретному сотруднику

**Замечание:** имена сотрудников считаются уникальными т.е. если уже есть сотрудник в базе с именем “Иванов Иван”, то второго такого же добавить нельзя. При добавлении часов сотрудника указывается его имя и по имени и осуществляется связь(привязка данных) с сотрудником.

### 

### Пример отчета по сотрудникам

*Отчет за период с [дата начала] по [дата окончания]*

*Иван отработал [n] часов и заработал за период [сумма] руб*

*Петр отработал [n] часов и заработал за период [сумма] руб*

*...*

*Всего часов отработано за период [N], сумма к выплате [сумма]*

*Желаете посмотреть еще один отчет? Нажмите (Д)а для продолжения. (Н)ет для выхода на главный экран программы.*

### Пример отчета по конкретному сотруднику

*Отчет по сотруднику: [Имя сотрудника] за период за период с [дата начала] по [дата окончания]*

*10.10.2020, 8 часов, исправлял работу модуля отчетов*

*11.10.2020, 8 часов, разработка новой функциональности модуля интеграции*

*12.10.2020, 10 часов, срочные исправления модуля интеграции*

*Итого: 26 часов, заработано: 2000 руб*

### Свойства руководителя

Месячная зарплата: 200 000 руб (Для отчета по дням расчет зарплаты производится так 8/160\*200 000 = 10000)

Месячная премия: 20 000 руб (выплачивается в случае если у руководителя были переработки). Для отчета по дням расчет премии производится так 8/160\*20000 = 1000

**Примечание**: количество рабочих часов в месяц считаем равным 160 для всех сотрудников.

## Сотрудник на зарплате

1. Может добавлять отработанные часы в список сотрудников. Добавлять время может только за себя. Возможно добавление времени задним числом.
2. Может просматривать свои отработанные часы и зарплату за период

### Свойства сотрудника на зарплате

Месячная зарплата: 120 000

Переработки свыше 160 часов оплачиваются в двойном размере.

## Внештатный сотрудник (фрилансер)

1. Может добавлять отработанные часы в список внештатных сотрудников. Добавлять время может только за себя. Возможно добавление времени задним числом не ранее чем за два дня от текущего времени.Т.е. если между текущим днем и тем днем в какой фрилансер хочет вписать часы прошло больше двух дней - не даем это сделать.
2. Может просматривать свои отработанные часы и зарплату за период

Месячная зарплата отсутствует. Оплата почасовая 1000 руб час. Оплата за переработки отсутствует.

# Технические рекомендации

Для хранения данных используем csv файлы (это простой текстовый файл) с разделителем запятая.

Пример информации из файла со списком отработанных часов сотрудников:

*10.10.2020,Иванов,8,работал весь день над задачей такой-то*

*10.10.2020,Петров,9,фича №1*

*11.10.2020,Иванов,8,исправлял баги в модуле отчетов*

Список сотрудников будет выглядеть так:

*Иванов,руководитель*

*Петров,сотрудник*

*Сидоров,фрилансер*

Хранение информации по ролям должно быть разделено на файлы. В конечном итоге всего будет 4 файла

1. Список сотрудников
2. Список отработанных часов руководителей
3. Список отработанных часов сотрудников на зарплате
4. Список отработанных часов внештатных сотрудников

При входе в программу на экране приветствия нам нужно

1. Представиться - т.е. ввести имя. По имени будет определена роль. И в зависимости от роли будут возможные опции. Если сотрудник с таким именем не найден - выдаем сообщение об ошибке.
2. Дальше показывается список действий доступных для текущей роли.
3. На первом экране также доступен выход из программы.

## Пример экрана для руководителя

*Здравствуйте, Иванов!*

*Ваша роль: руководитель*

*Выберите желаемое действие:*

*(1). Добавить сотрудника*

*(2). Просмотреть отчет по всем сотрудникам*

*(3). Просмотреть отчет по конкретному сотруднику*

*(4). Добавить часы работы*

*(5). Выход из программы*

**Важно:** Программа не предполагает действий по удалению сотрудников, времени работы и т.п.

*Данное задание разработано для подписчиков ютюб канала*

[*https://www.youtube.com/c/SBeregovoyRU/*](https://www.youtube.com/c/SBeregovoyRU/)

*Спасибо всем кто заинтересовался и собирается выполнить данный проект.*