Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Отчет по дисциплине “Новые технологии в программировании”

Студент гр. 586-2:

\_\_\_\_\_\_\_\_Шагаев И.А.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г

Руководители:

Доцент кафедры КСУП ТУСУР

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Томск 2018

**Техническое задание**

**Проблема:**

Бухгалтеры используют неудобную информационную систему для выплаты зарплат рабочим.

**Цель:**

Разработать программу для создания зарплат разных видов. Хранение информации о них, с возможностью редактирования, а также открытие и сохранение ранее созданных таблиц зарплат.

**Задачи:**

**-**Программа должна создавать файлы в формате Json

-Должны предусмотрены следующие типы зарплат: почасовая, по окладу и ставке

-Реализовать добавление, удаление и редактирование данных о зарплатах

-Создать установщик программы

-Предусмотреть любые некорректные действия конечного пользователя

**Критерии качества:**

-Отсутствие непредусмотренных ошибок программы

-Удобный для использования интерфейс

**Функциональные возможности:**

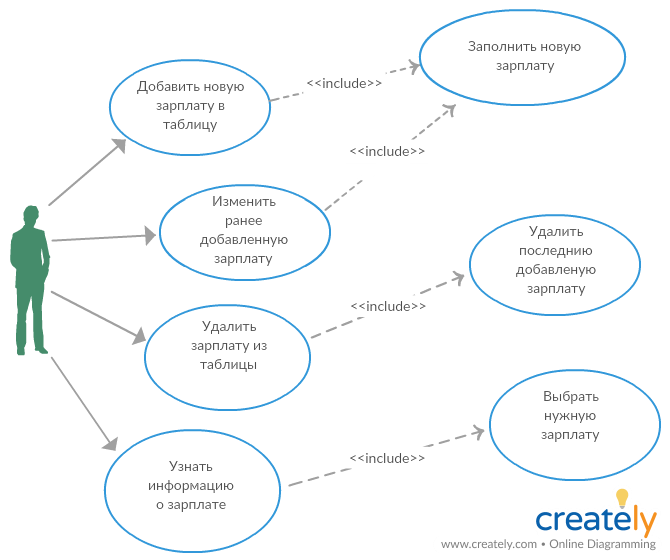
1. Добавление зарплаты в таблицу
2. Удаление зарплаты из таблицы
3. Просмотр информации о зарплате
4. Сохранение и открытие таблицы зарплат

**Введение**

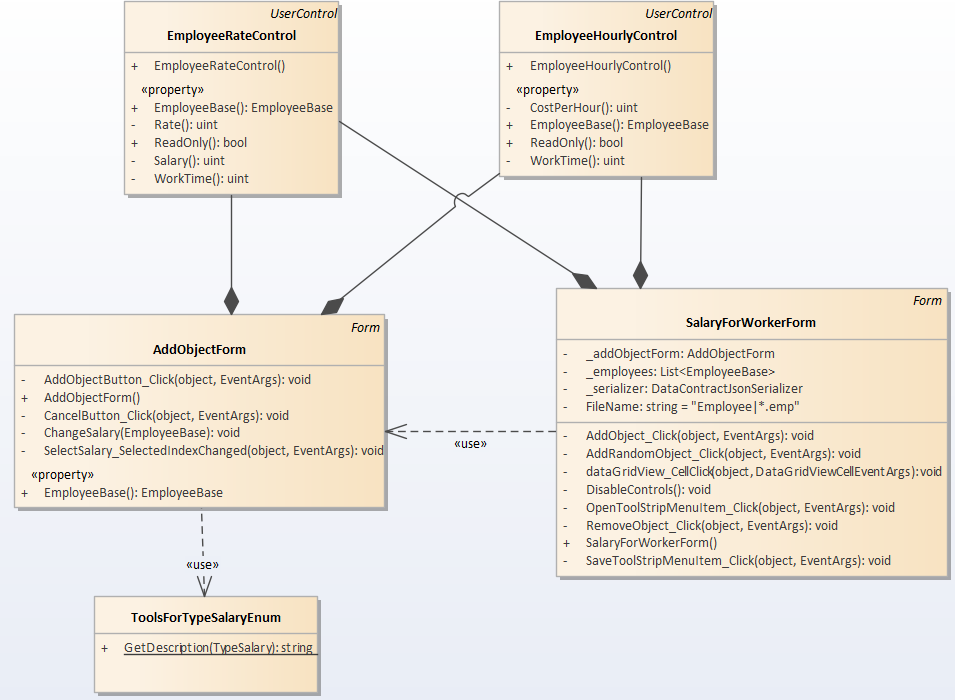
Данная документация предназначена для ознакомления других специалистов и для конечного пользователя. Описание возможностей программы и их реализация позволяет ускорить ознакомление с программой.

**Описание программной системы**

**UML диаграмма вариантов использования:**



UML диаграмма классов Приложения:



|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
|  | Описание класса |  |
| Класс *EmployeeBase* – сущность для описания абстрактного рабочего с зарплатой в программе | | |
|  | Свойства |  |
| + WorkTime | uint | Время работы |
| + Payroll | uint | Подсчет зарплаты |
| + Type | string | Тип зарплаты |
|  | Методы |  |
| # CheckingData | void | Проверка при присвоение вводимых значений полям |
| # EmployeeBase (uint) |  | Конструктор для создания зарплаты.  *Uint* – время работы |

**Описание классов зарплат**

Таблица 1 – Описание класса EmployeeBase

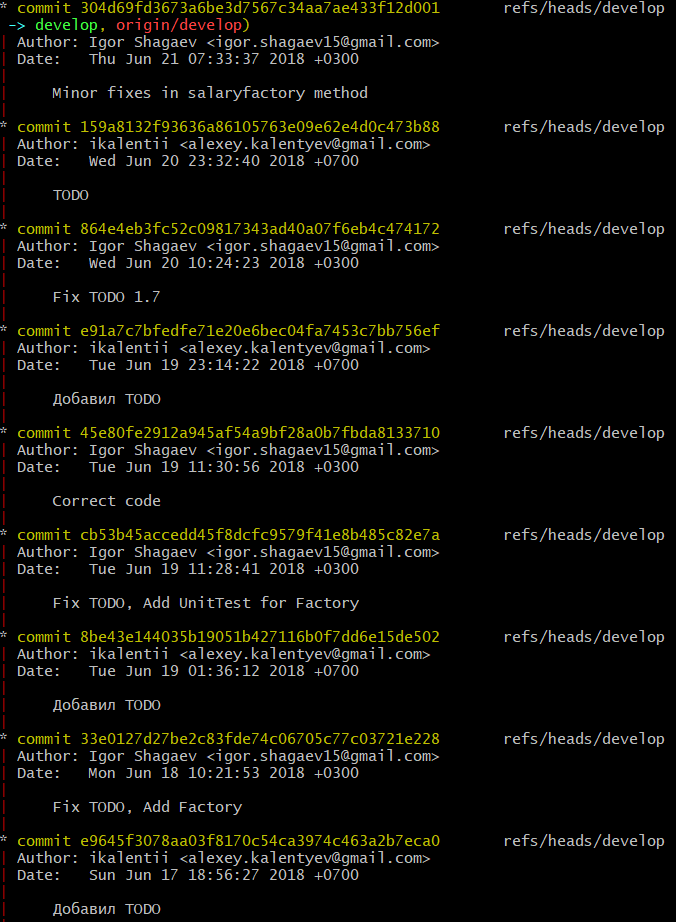
Таблица 2 – Описание класса EmployeeHourly

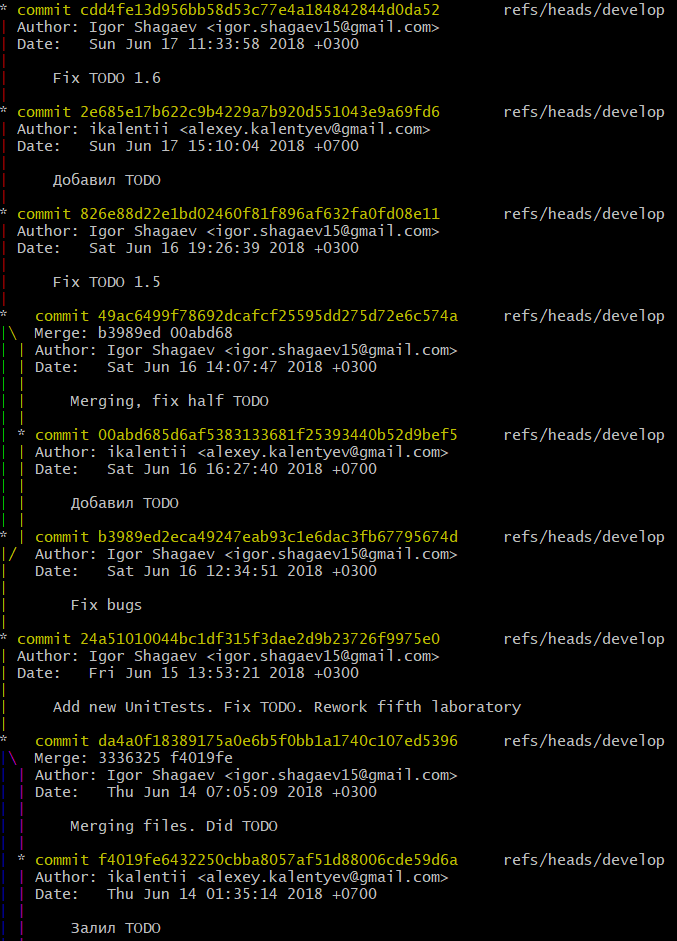
|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
|  | Описание класса |  |
| Класс *EmployeeHourly* – сущность для описания зарплаты по ставке и окладу | | |
| Свойства | | |
| + CostPerHour | uint | Оплата в час |
| + Payroll | uint | Подсчет почасовой зарплаты |
| + Type | string | Тип зарплаты – почасовая |
| Методы | | |
| + EmployeeHourly (uint, uint) |  | Конструктор для создания почасовой зарплаты.  *Uint* – время работы.  *Uint* – оплата в час. |

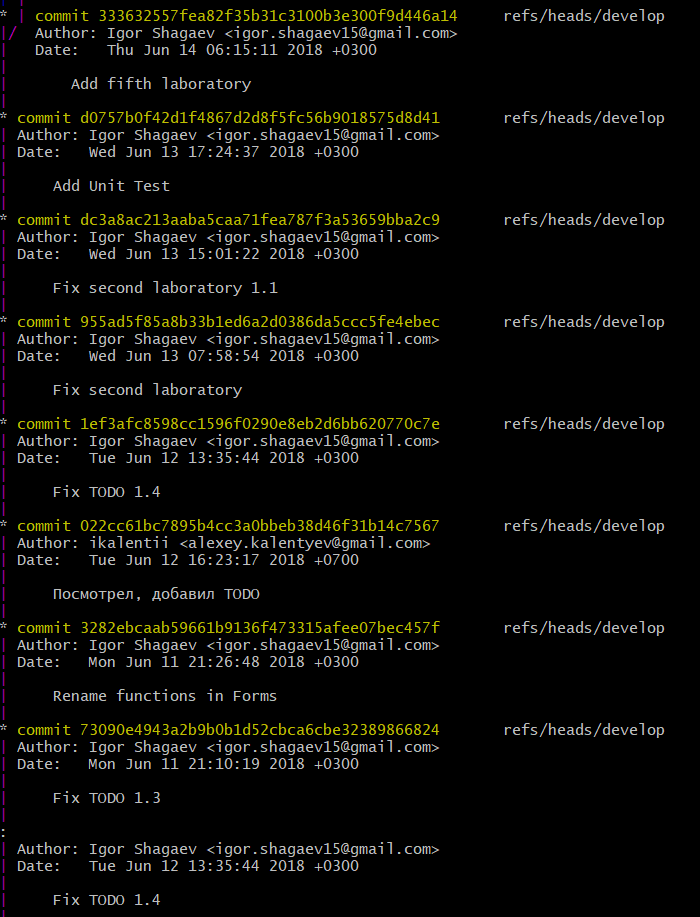
Таблица 3 – Описание класса EmployeeRate

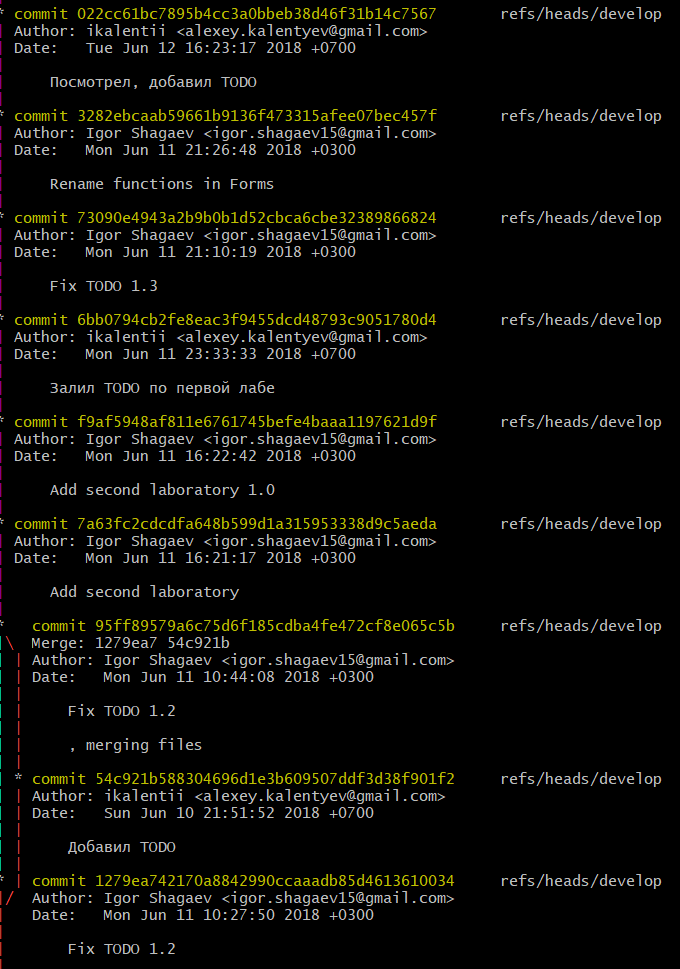
|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *EmployeeRate –* сущность для описания зарплаты по ставке и окладу | | |
| Свойства | | |
| + Rate | uint | Ставка (норма работы) |
| + Salary | uint | Оклад |
| +Payroll | uint | Подсчет зарплаты по ставке и окладу |
| + Type | string | Тип зарплаты – по ставке и окладу |
| Методы | | |
| + EmployeeRate (uint, uint, uint) |  | Конструктор для создания зарплаты по ставке и окладу.  *Uint* – время работы.  *Uint* – оклад.  *Uint* – ставка (норма работы). |

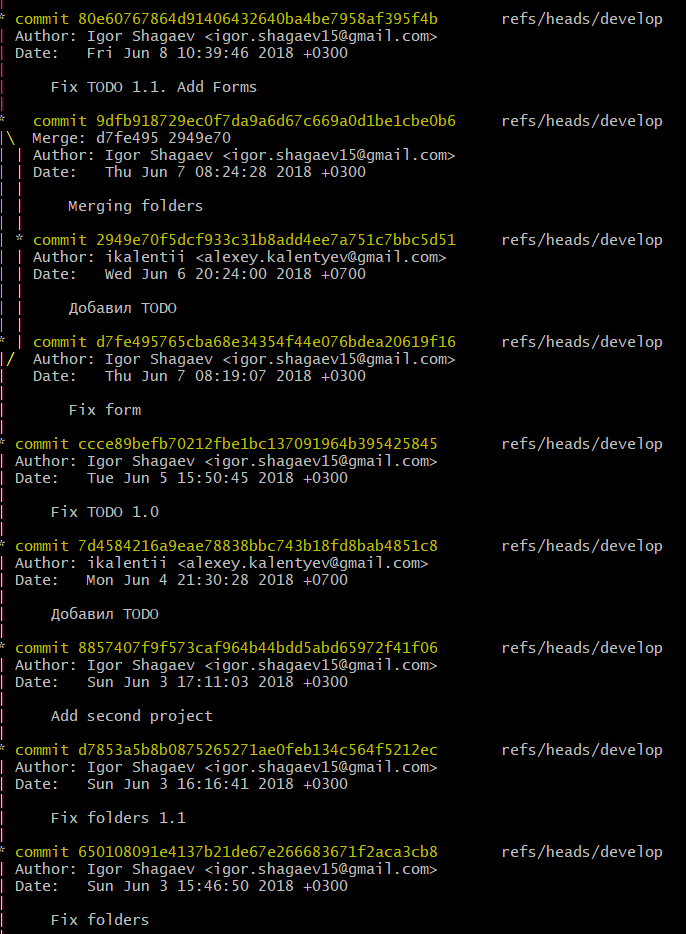
**Дерево ветвлений git**

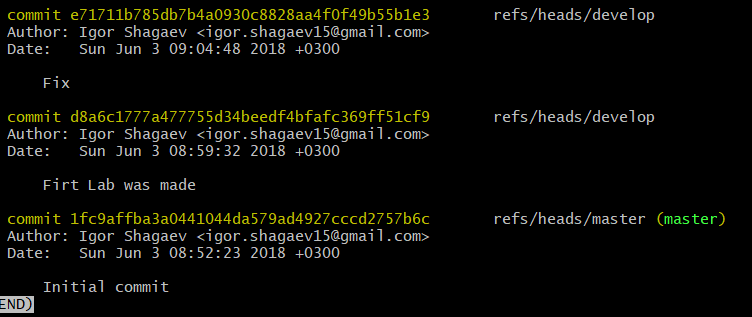
****

****

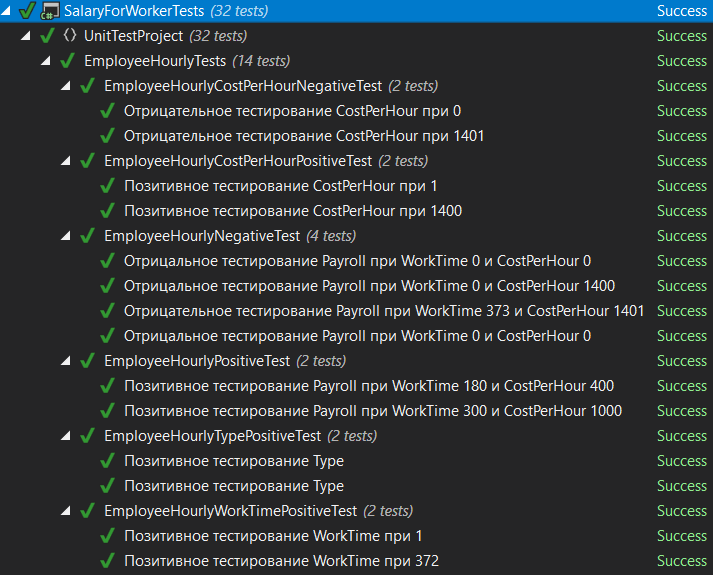
****

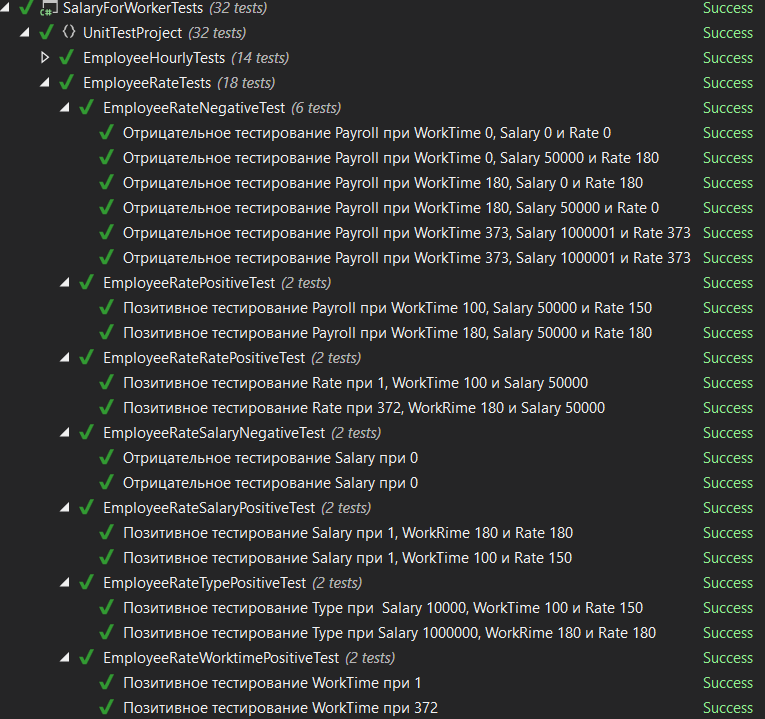
****

****

****

**Тестирование**

****

****

**Заключение**

В результате работы была разработана информационная система для подсчета зарплаты двух видов: по часам, по окладу и ставке.

Была разработана .dll библиотека классов, работа которой продемонстрирована в консоле. После данная библиотека добавлена в Windows Forms. С помощью элементов в Windows Forms был разработан пользовательский интерфейс. Для тестирования использовался NUnit для Visual Studio. Были проведены положительные и отрицательные тесты открытых свойств классов.