МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЗО

Кафедра Автоматизированных систем управления

**ОТЧЁТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №11**

по дисциплине

«Программирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Яковлев В.А..* | Преподаватель | *Эстрайх И.В.* | |
| Факультет | *ЗО* |  |  | |
| Направление (специальность) подготовки | *09.03.03*  *Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *ДЭ–220* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр | *140602737* |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Подпись студента | | Подпись преподавателя | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | |

Новосибирск 2024

**Оглавление**

[I. Задание 3](#_Toc157199979)

[II. Этапы решения задачи 4](#_Toc157199980)

[Решение пункта 1 4](#_Toc157199981)

[Решение пункта 2 5](#_Toc157199982)

[Решение пункта 3 7](#_Toc157199983)

[Решение пункта 4 8](#_Toc157199984)

## I. Задание

1. Разработайте классы, моделирующие работников с почасовой (класс I) и комиссионной оплатой труда (класс II). У каждого работника есть ФИО и пол. Добавьте в класс I метод Работать, в который передаётся количество часов, отработанных работником. Добавьте в класс II метод Продать, в который передаётся сумма, на которую были проданы товары. Объекты накапливают объёмы отработанных часов и суммы проданных товаров соответственно. Добавьте в оба класса метод Рассчитать зарплату, который обнуляет накопленный объём/сумму и возвращает зарплату к выплате работнику. Работник с почасовой оплатой труда получает фиксированную сумму за каждый час работы до некоторого нормативного значения, а потом увеличенную сумму за каждый час переработки. Работник с комиссионной оплатой труда получает фиксированный оклад, а также некоторый процент с продаж.
2. Создайте класс предприятие с возможностью приема на работу и увольнения сотрудников. Добавьте метод МоделироватьРаботу, принимающий количество дней, в течение которых моделируется работа компании. Каждый сотрудник каждый день выполняет случайный объём работы. Два раза в месяц каждый сотрудник получает ЗП.
3. Необходимо реализовать сериализацию объектов ваших классов из предыдущей лабораторной работы, т.е. их преобразование в поток (последовательность) байтов и обратно для возможности передачи по сети, записи в БД и т.д. В качестве формата можете использовать любой, например, JSON, XML, BSON, свой собственный.
4. Классы должны быть обеспечены модульными тестами.

Необходимо использовать язык C#

## II. Этапы решения задачи

### Решение пункта 1

Абстрактный класс рабочего:

Свойства:

* Name: string; // Имя рабочего
* Sex: Sex; // Пол рабочего

Методы:

* Payoff() // Получить зарплату
* GetInfo() // Получть информацию о читателе
* CalculateSalary() // Рассчитать зарплату без обнуления

Класс рабочего с почасовой оплатой труда:  
Наследует:

* Абстрактный рабочий

Свойства:

* StandardWorkHoursPerDay: int; // Стандартное количество рабочих часов в день
* WorkedHours: int; // Отработано часов
* SalaryPerHour: decimal; // Оклад
* OvertimeMultiplier: decimal; // Коэффициент переработки
* DaysWorked: int // Отработано дней

Методы:

* Work() // Работать
* Payoff() // Получить зарплату
* GetInfo() // Получить информацию о рабочего
* ResetWorker() // Сбросить отработанное время рабочего
* CalculateSalary() // Рассчитать зарплату без обнуления

Класс рабочего с комиссионной оплатой труда

Наследует:

* Абстрактный рабочий

Свойства:

* SoldPrice: decimal; // Сумма проданных товаров
* BaseSalaryPerDay: decimal; // Базовый оклад в день
* Comission: decimal; // Комиссия с продаж
* DaysWorked: int; // Отработано дней

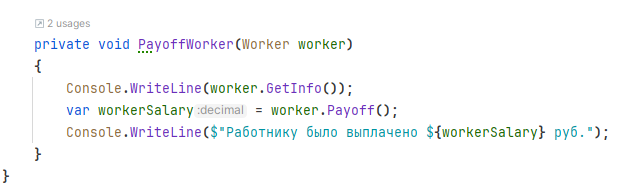
Методы:

* Sell(decimal soldPrice) // Работать
* Payoff() // Получить зарплату
* GetInfo() // Получить информацию о рабочего
* ResetWorker() // Сбросить отработанное время рабочего
* CalculateSalary() // Рассчитать зарплату без обнуления

### Решение пункта 2

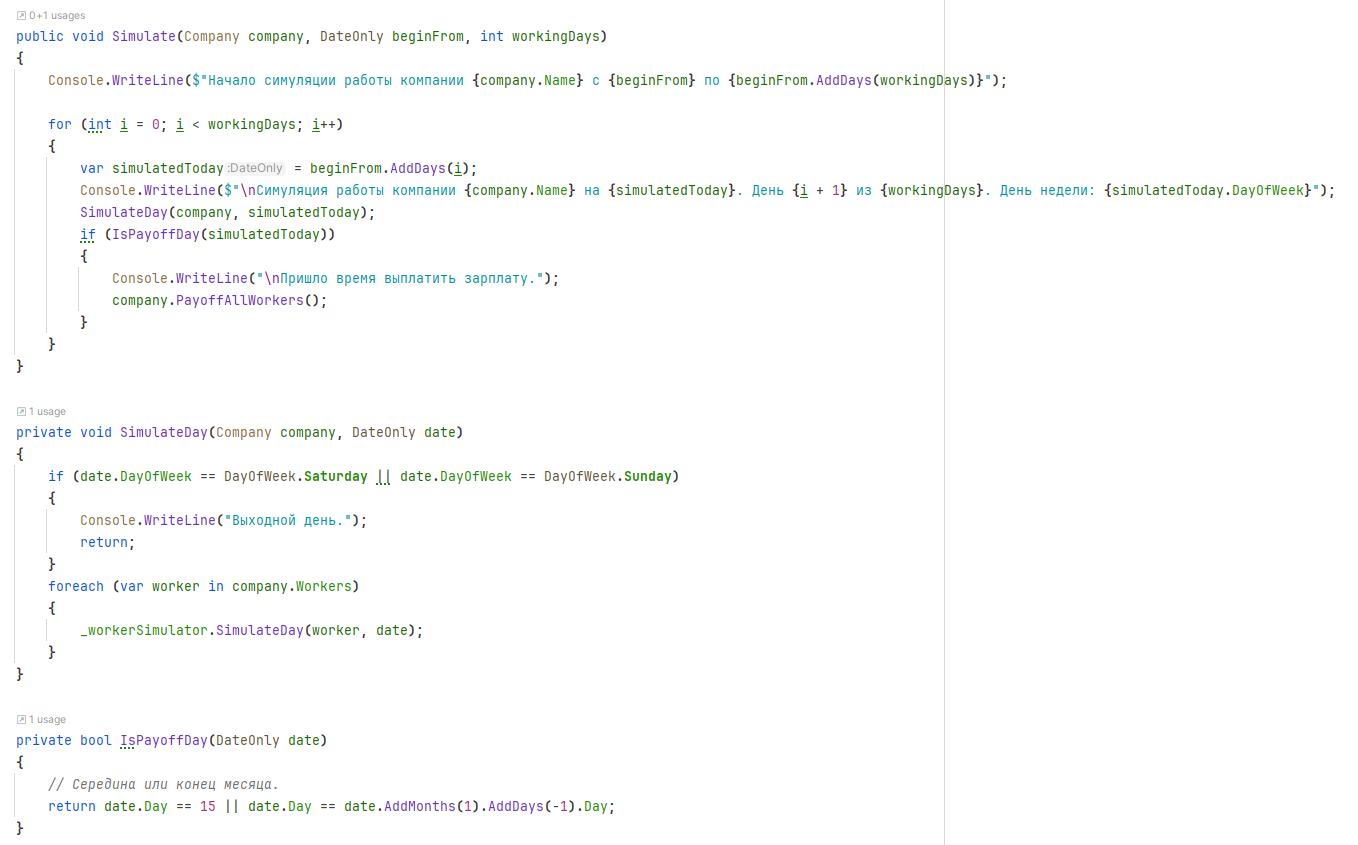
Класс “Предприятие”:



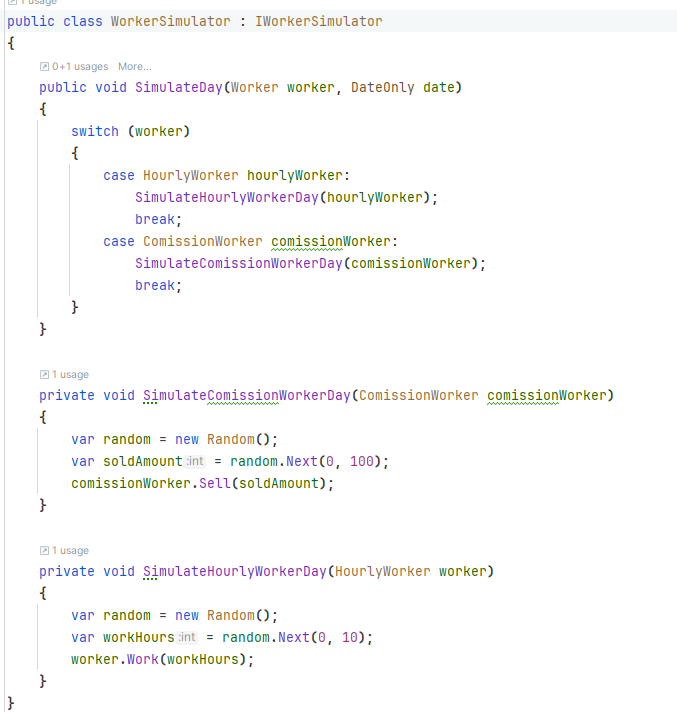


Для моделирования работы были созданы специальные классы, симулирующие работу компании. Оба класса реализуют соответствующие интерфейсы. Это было сделано с учётом возможности изменения логики работы компании.

CompanySimulator:

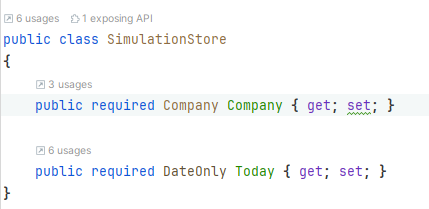


WorkerSimulator:



### Решение пункта 3

Для сериализации было принято решение сохранять файл в формате JSON. Для удобного сохранения, был создан класс состояния приложения:



Для чтения и записи в файл был создан класс записи. Сериализация и десериализация выполняется при помощи библиотеки System.Test.Json.



### Решение пункта 4

В качестве основы среды тестирования была выбрана библиотека xUnit

Во время написания тестов симуляции работы предприятия, были разработан Mock-класс симулятора рабочего.



