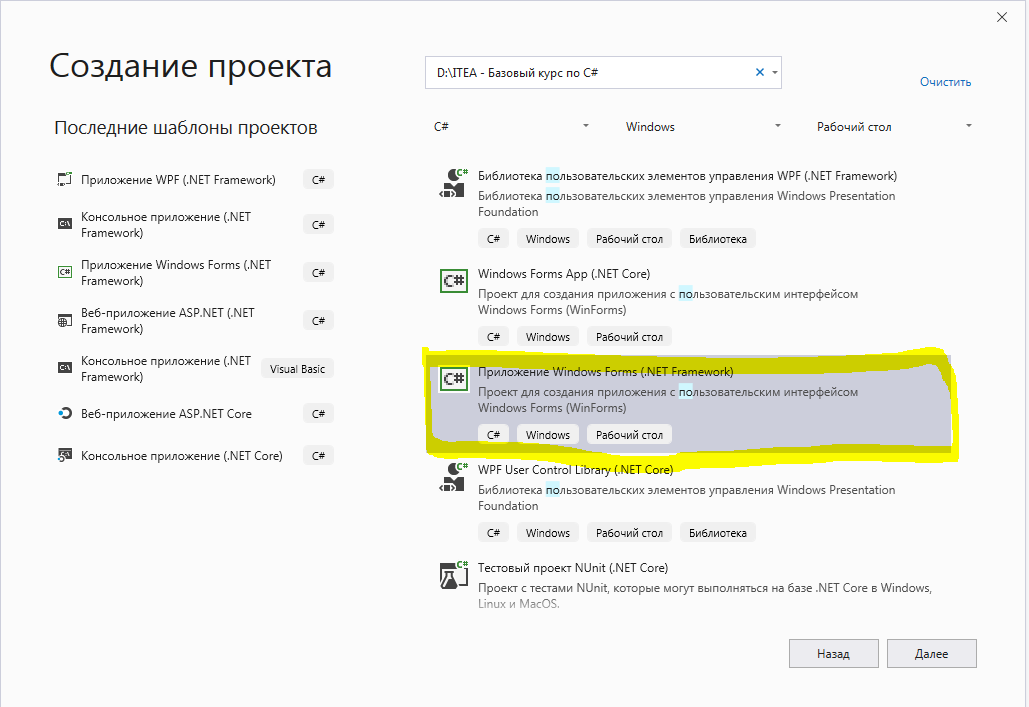
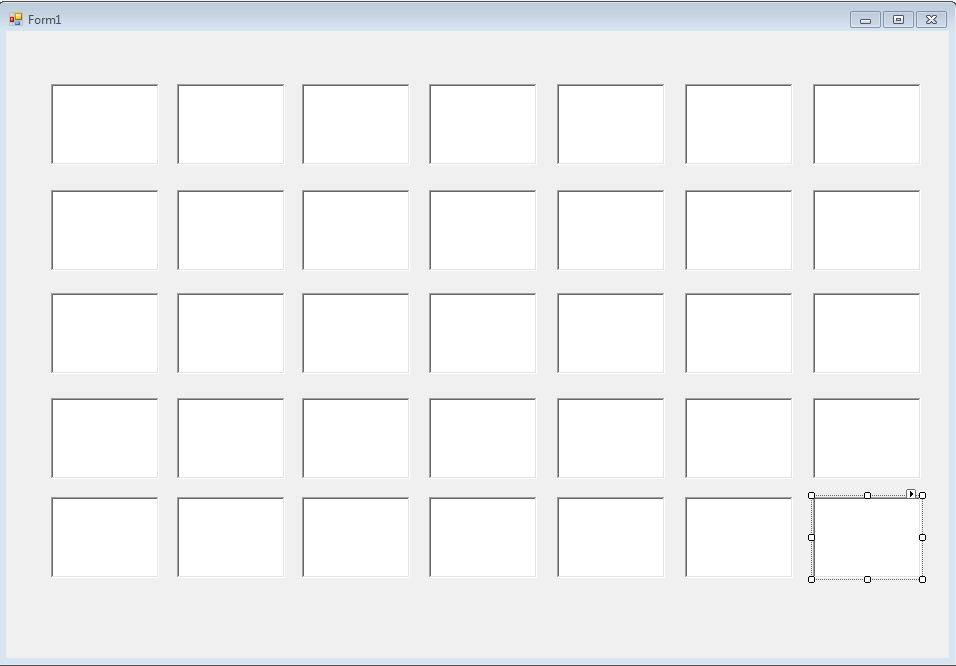
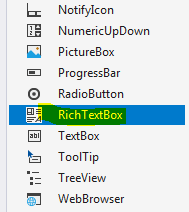
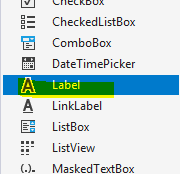
Финальный проект.

Используя Visual Studio, создайте проект по шаблону Windows Forms.



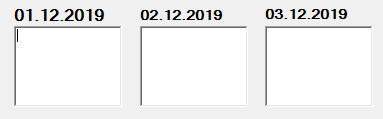
Общий вид окна приложения должен иметь следующий вид: 

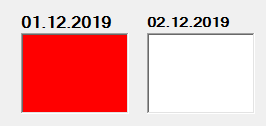
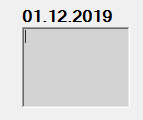
Для добавления контейнеров на форму использовать RichTextBox.

. 

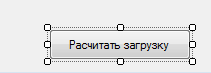
Далее добавить над каждым RichTextBox добавить метку Label, организовать логику по 

заполнению меток, таким образом, чтоб каждая метка заполнялась датой, в общем она должна иметь следующий вид:

, метки должны заполнятся динамически, в зависимости от месяца и года, по умолчанию брать текущий год и месяц. Также если на форме больше RichTextBox и меток, чем в месяце который отображается на форме, то лишние элементы должны скривятся. Также необходимо реализовать, заливку RichTextBox, красным цветом, если день выходной:

, и заливку серым цветом всех дней, если день уже прошел, заливка по прошедшим дням должны перекрывать выходные дни. В каждый RichTextBox пользователь должен иметь возможность вносить свои данные. После заполнения данными всеx полей, необходимо проанализировать загрузку пользователя. Для анализа загрузки, необходимо реализовать кнопку на форме:

, данная кнопка должна расчитывать загрузку по следующему алгоритму:

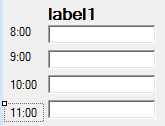
1. Берем количество всех дней в месяце и умножаем их на 8 = (31\*8 = 248 часов) - весь фонд времени за месяц.
2. Анализируем заполнения RichTextBox, считаем, если в нем есть текст, тогда этот день занят на 100%, если записи нету , тогда свободен.
3. После расчета программа должна выдавать отчет, о том, сколько дней свободно, сколько занято, также информация должна быть и в часах.

Для реализации данного алгоритма необходимо создать отдельный класс, в котором прописать логику.

Также на форму необходимо добавить обычный TextBox , в который пользователь может вводить год и месяц, и в зависимости от введенного значения, форма должны заполнятся соответственными значениями дат, реализовать проверку с помощью try...catch...finally, если вводится значение неправильного формата.

**Дополнительное задание (не сложное).**

Реализовать возможность разбивки дня на часы следующим образом:

, модифицировать класс для расчета загрузки соответственно добавленной разбивки.

**Дополнительное задание (сложное, не обязательное).**

Создать текстовый файл, продумать формат хранения данных, используя разделители. Реализовать возможность сохранения данных при выходе из календаря, и восстановления данных при открытии приложения. Использовать классы StreamWriter и StreamReader, для записи и чтения данных.