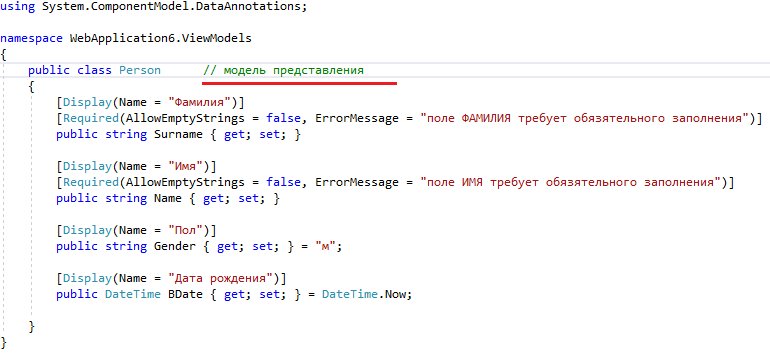
**Лекция**

Программирование интернет-серверов

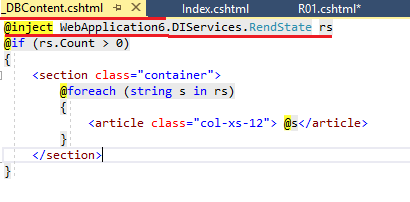
**Работа с представлениями (валидация)**

1. **ModelState**: объект, для хранения состояния представления. Заполнение ModelState при заполнении параметров action.
2. **View Model**: объект, применяемый для передачи данных в представлении.

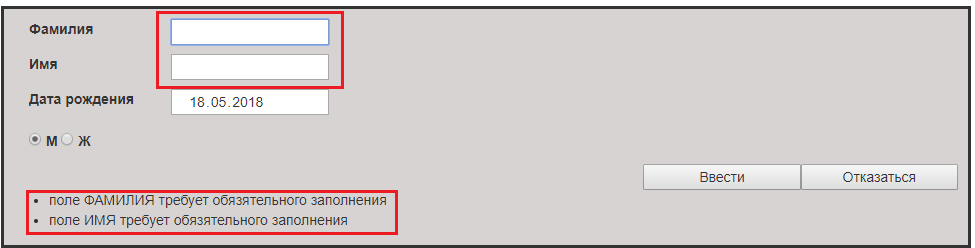






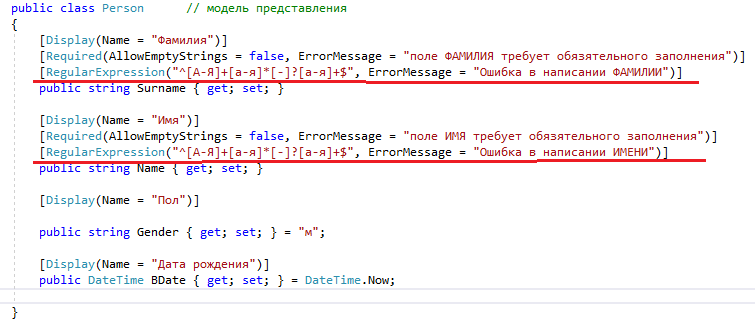


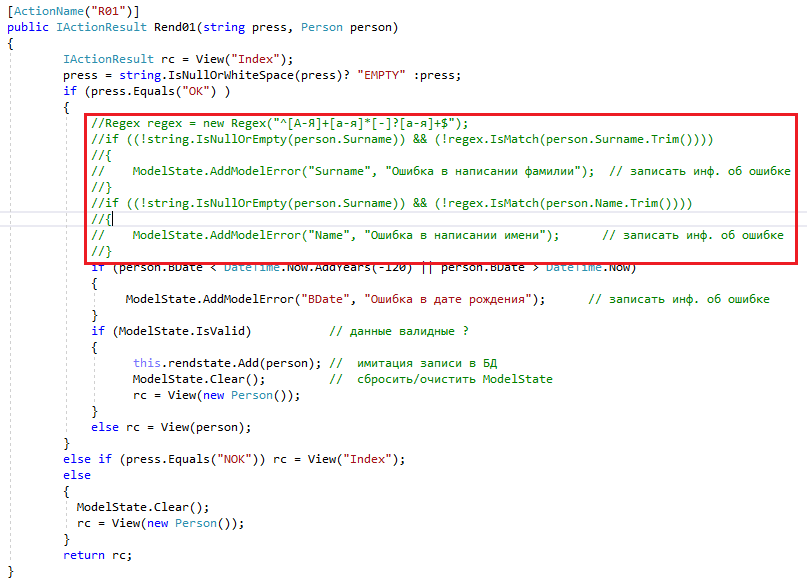




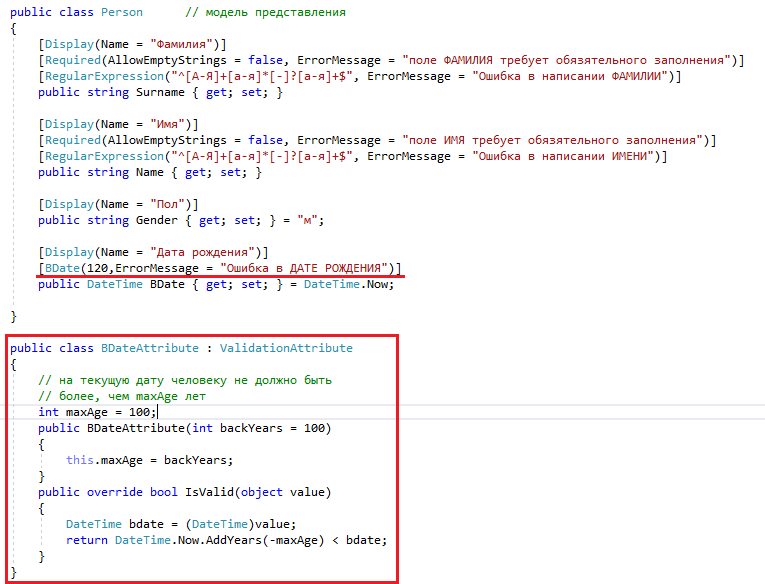


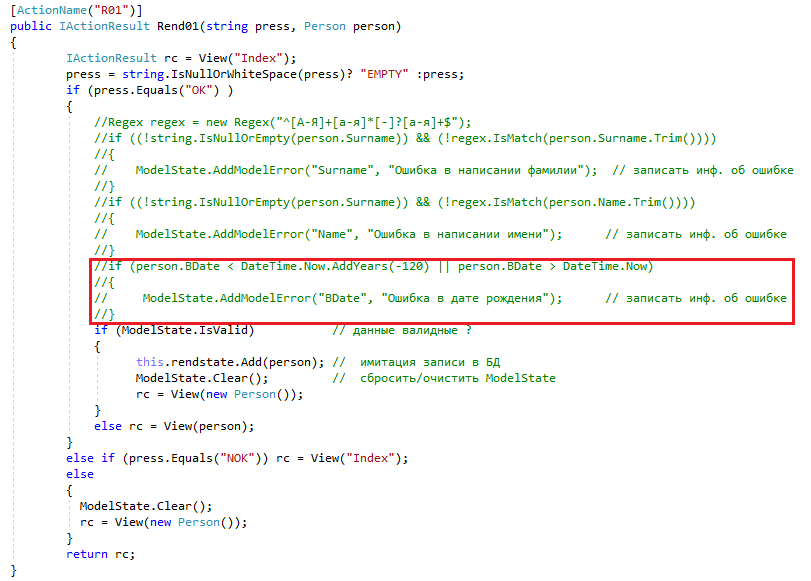
1. **ModelState**: можно для проверки написания фамилии и имени можно было использовать стандартный атрибут валидации [RegularExpression]





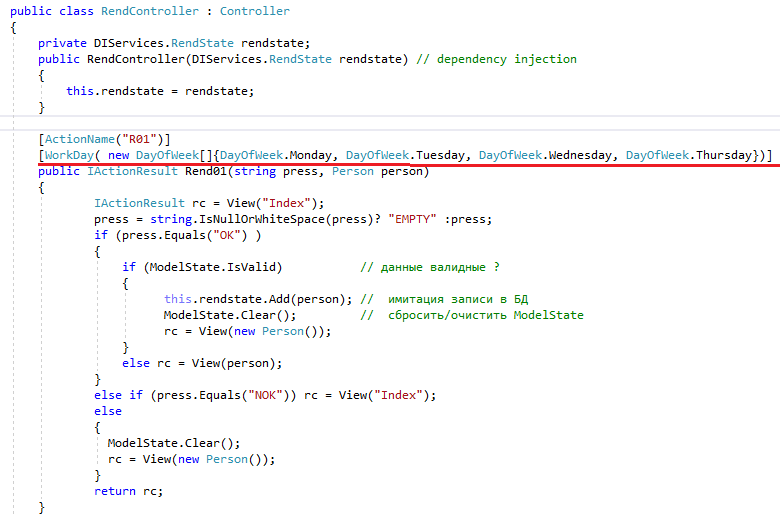
1. **ModelState**: можно для проверки даты рождения разработать собственный атрибут валидации

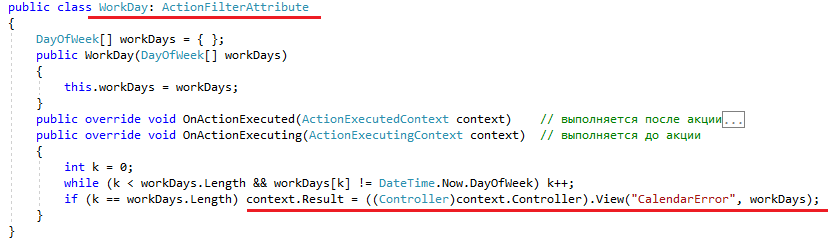




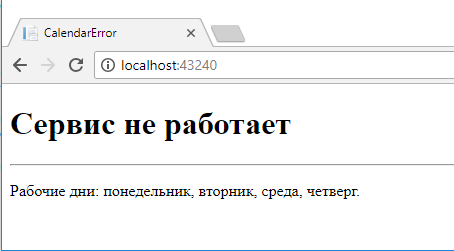
1. **MVC Filters**: механизм автоматического выполнения кода в рамках контроллера; фильтры: действий, результатов, исключений, авторизации. Фильтры могут быть в форме атрибутов акции (действует для одну акцию) или атрибутов контроллера (действует для всех акций контроллера). Для однородны фильтров может быть установлен приоритет (Order, чем больше, тем приоритетней)

1. **Action Filter**: позволяет автоматически выполнить методы класса до и после выполнения акции (действия) контроллера.

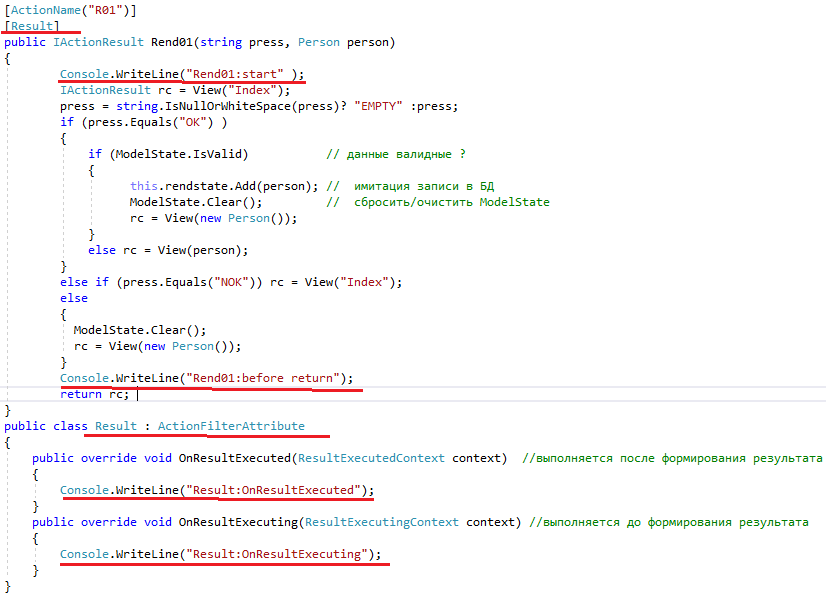


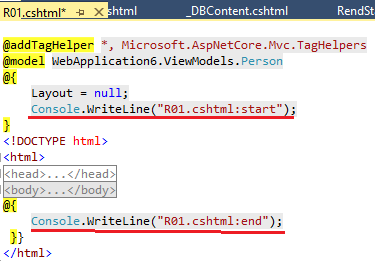




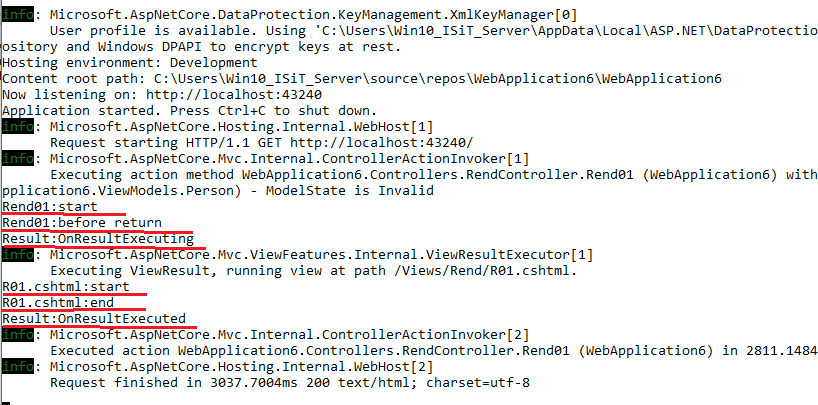


1. **Result Filter**: позволяет выполнить методы класса, до и после формирования объекта-результата (IActionResult), возвращаемого акцией.









1. **Authorization Filter**: проверяет наличие авторизации текущего пользователя
2. **Resource Filter:** выполняетсяпосле фильтра авторизации и после всех остальных фильтров.
3. **Exception Filter:** обработка исключений в акциях
4. **MVC Filters**: общая схема выполнения фильтров



1. **MVC Filters:** Для однородных фильтров может быть установлен приоритет (Order, чем больше, тем приоритетней). Для этого фильтр должен реализовывать IOrdersFilter (иначе Order = 0).
2. **Service Filters**:
3. k
4. k
5. k
6. k
7. k
8. k