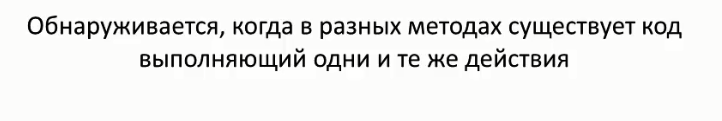
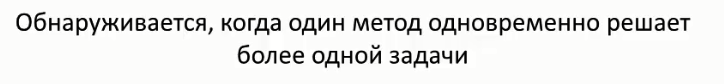


Плохие запахи кода

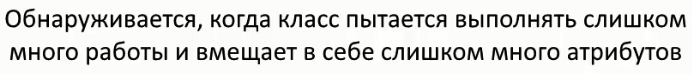
Дублирование кода



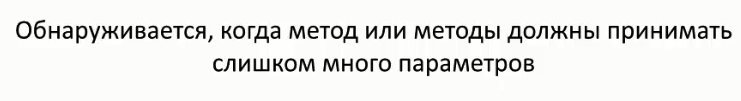
Длинный метод



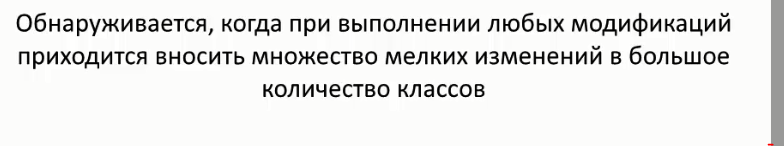
Большой класс



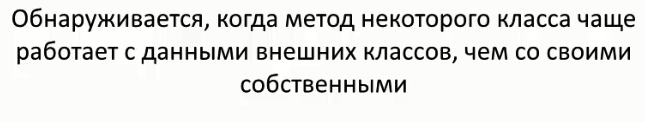
Длинный список параметров



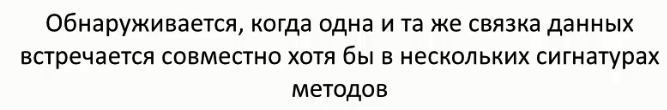
Стрельба дробью, желательно чтобы связь между классами была один к одному!



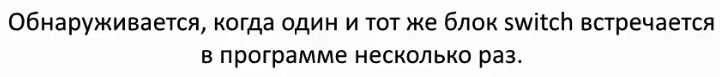
Завистливые функции



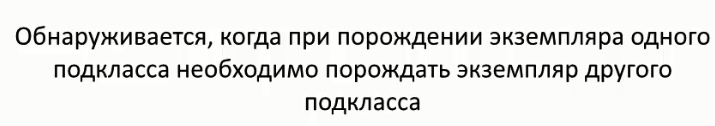
Группы данных



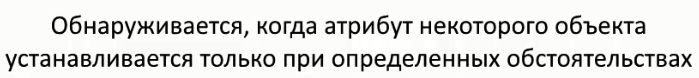
Группы switch



Параллельные иерархии классов

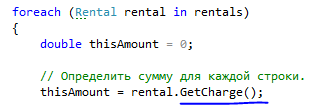


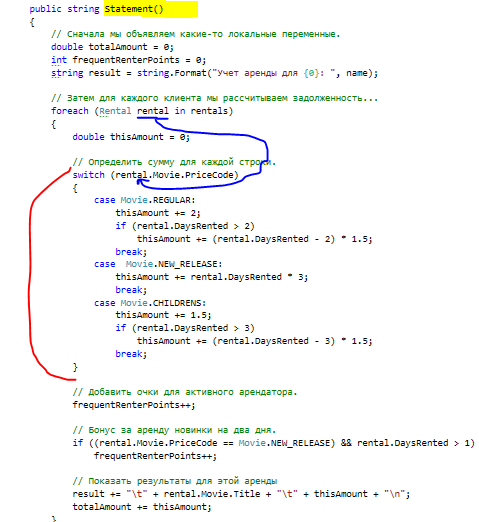
Временное поле



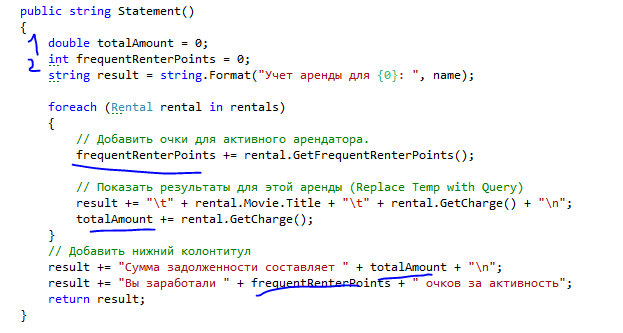
Метод очень большой, его необходимо разбить на несколько методов

1 выделить в отдельный метод оператор switch, определить принимающие параметры, но при этом оператор будет больше интересоваться другим классом а не классом в теле которого он находится ПЕРЕНОСИМ ЕГО В ДРУГОЙ КЛАСС

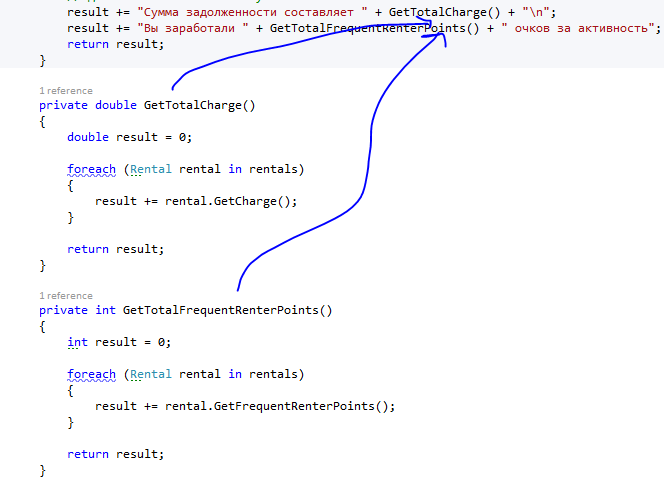




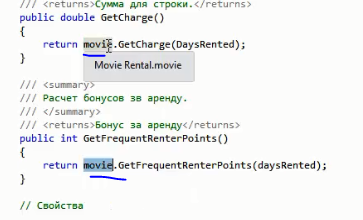
Теперь проблема много локальных переменных



Нужно выделить их в отдельные методы

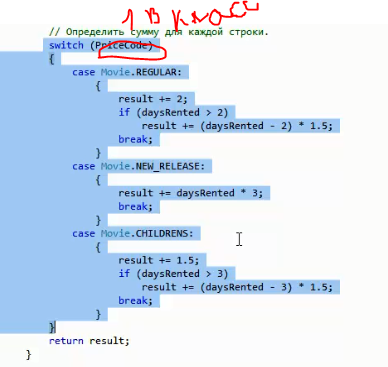


В классе рендел switch (его нужно перенести в класс муви в связи с тем что при добавлении новых жанров необходимо будет изменять switch в ином классе (стрельба дробью один ко многим зависимость) ) переносим в муви так как он больше интересуется его параметрами а так же перенесем методы расчета

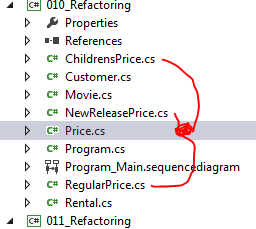


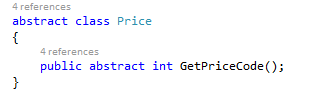
Теперь оператор switch нужно разбить на классы (применить паттерн состояние)

Поле прайс выделить в класс

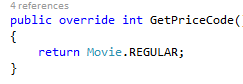


Создаем абстрактный класс Price и от него наследуем три класса

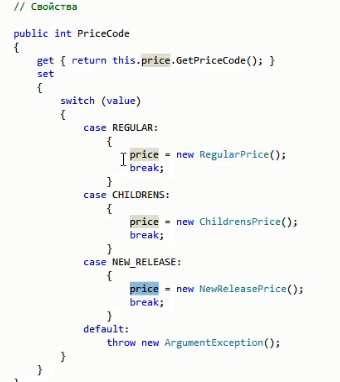




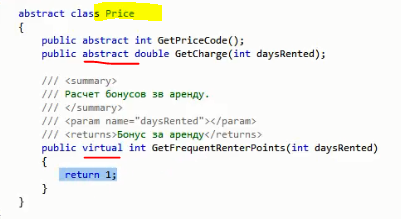
И каждый класс теперь возвращает перечислитель

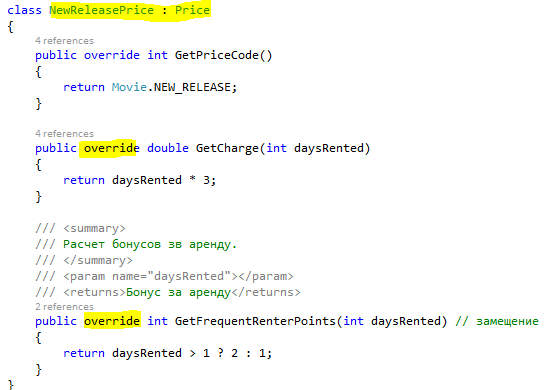


Теперь инициализируем Price



И теперь мы применяем полиморфизм





И теперь вызываем полиморфизмом метод

