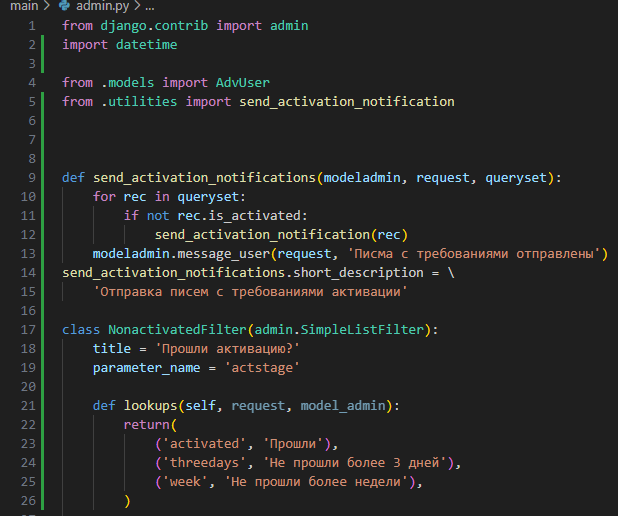
**Инструменты для администрирования пользователей**

Добавим возможность фильтрации пользователей по именам, адресам электронной почты, настоящим именам и фамилиям. Также реализуем вывод пользователей, уже выполнивших активацию, не выполнивших её в течение трёх дней и недели, и действие по отправке выбранным пользователям писем с требованиями пройти активацию.

Код класса редактора **AdvUserAdmin**, вспомогательного класса и функции следует занести в модуль **admin.py** пакета приложения, заменив им имеющийся там код.





В списке записей указываем выводить строковое представление записи (имя пользователя – как реализовано в модели **AbstractUser**, от которой наследуется наша модель), поле признака, выполнил ли пользователь активацию, временНую отметку его регистрации. Также разрешаем выполнять фильтрацию по полям имени, адреса электронной почты, настоящих имени и фамилии.

Для выполнения фильтрации пользователей, выполнивших активацию, не выполнивших её в течении трёх дней и недели, используем класс **NonactivatedFilter**. Обратим внимание на код, непосредственно фильтрирующий пользователей по значению даты их регистрации.

Мы явно указываем список полей, которые должны выводится в формах для правки пользователей, чтобы выстроить их в удобном для работы порядке. Поля даты регистрации пользователя и последнего его входа на сайт делаем доступными только для чтения.

Наконец регистрируем действие, которое разошлёт пользователям письма с предписаниями выполнить активацию. Это действие реализовано функцией **send\_activation\_notifications().** В ней мы преобразуем всех выбранных пользователей и для каждого, кто не выполнил активацию, вызываем функцию **send\_activation\_notification()**, объявленную ранее в модуле **utilities.py** и непосредственно производящую отправку писем.