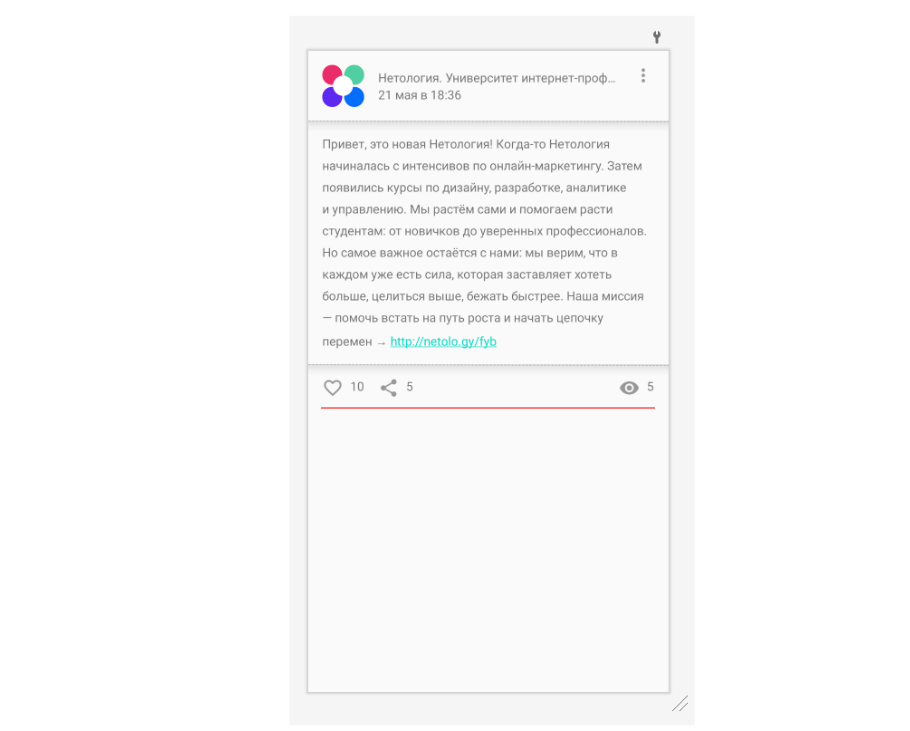
**Домашнее задание к занятию «2.1. Обработка событий в Android»**

**Задача Like, Share**

**Легенда**

Поскольку вы делаете приложение с гораздо бóльшими функциональными возможностями, чем делают лекторы, то придётся дописать часть логики (по отношению к лекции).

Ваше приложение должно выглядеть так:

[](https://github.com/netology-code/and2-homeworks/blob/master/04_events/pic/layout.png)

**Задача**

1. При клике на like должна меняться не только картинка, но и число рядом с ней: лайкаете — увеличивается на 1, дизлайкаете — уменьшается на 1.
2. При клике на share должно увеличиваться число рядом: 10 раз нажали на share — +10.
3. Если количество лайков, share или просмотров перевалило за 999, должно отображаться 1K и т. д., а не 1 000. Предыдущие функции должны работать: если у поста было 999 лайков и вы нажали like, то должно стать 1К, если убрали лайк, то снова 999.

Обратите внимание:

1. 1.1К отображается по достижении 1 100.
2. После 10К сотни перестают отображаться.
3. После 1M сотни тысяч отображаются в формате 1.3M.
4. Подумайте, можно ли это вынести в какую-то функцию вместо того, чтобы хранить эту логику в Activity.

Опубликуйте изменения в вашем проекте на GitHub. Убедитесь, что apk собирается с помощью GitHub Actions и при установке в эмуляторе приложение работает корректно.

В качестве результата пришлите ссылку на ваш GitHub-проект в личном кабинете студента на сайте [netology.ru](https://netology.ru/).

**Задача Parent Child**

**Легенда**

Исследование поведения системы — важная часть работы разработчика.

До этого мы устанавливали OnClickListener только на один View.

А если мы установим разные Listener на View, которые пересекаются: на ConstraintLayout, который содержит все остальные View, и на кнопку Like?

**Задача**

1. Установите обработчики OnClickListener на binding.root и binding.like.
2. Поставьте внутрь обработчиков точки останова.
3. Запустите приложение в режиме отладки.
4. Кликните на кнопку Like, на кнопку с тремя точками (на ней пока нет обработчика), на текст, на аватар.
5. Установите обработчик OnClickListener на аватар и кликните по нему снова.

В качестве результата напишите ответы на следующие вопросы:

1. Какой из обработчиков сработал при клике на кнопку Like?
2. Сработал ли обработчик на binding.root при клике на кнопку с тремя точками?
3. Сработал ли обработчик на binding.root при клике на текст?
4. Сработал ли обработчик на binding.root при клике на аватар до установки на avatar собственного обработчика?
5. Сработал ли обработчик на binding.root при клике на аватар после установки на avatar собственного обработчика?

Попробуйте выявить закономерность: когда срабатывает обработчик на контейнере, а когда нет.

Если не получается работать с отладчиком или ставить точки останова, добавьте в каждый обработчик по println с разным текстом. Если при нажатии текст будет выводиться в LogCat, то обработчик был вызван. Если текста не будет, значит обработчик вызван не был.