

Создайте приложение для конвертации валют.

Приложение должно состоять из трех экранов - "Валюты", "Обмен" и "Транзакции".

## Экран "Валюты"

Экран "Валюты" должен работать в двух режимах.

### Валюты: список

Каждую секунду нужно вызывать метод `getRates` из `RemoteRatesServiceImpl`, передавая туда желаемую сумму с кодом выбранной валюты и показывать в виде списка возвращаемые кросс-курсы.

В этом режиме по щелчку на любую другую валюту из списка она становится выбранной, перемещается вверх списка и кросс-курсы теперь пересчитываются по отношению к ней.

В этом режиме кросс-курсы должны пересчитываться по отношению к 1 единице выбранной валюты.



Figure 1: Экран "Валюты: Список" – пример, из источника данных будет возвращаться больше валют

Для каждого элемента в списке нужно показать флаг, название валюты и ее аббревиатуру, кросс-курс и количество этой валюты на балансе (если есть, подробнее ниже).

## Валюты: ввод суммы

“Валюты: ввод суммы” - при щелчке на 1 в элементе выбранной валюты должна появляться возможность ввести нужную сумму цифрами, при этом суммы элементах других валют в списке должны автоматически пересчитываться в зависимости от введенного количества выбранной валюты (количество выбранной валюты умноженное на кросс-курс).

Кроме того, теперь в списке нужно показывать только те валюты, которые присутствуют на счету в базе данных в достаточном количестве для покупки введенной пользователем суммы выбранной валюты (таблица accounts в базе данных, класс AccountDb). То есть, если пользователь хочет купить 200 евро а на его рублевом счету имеется только 500 рублей, рубли в списке валют показывать не нужно.

После того, как пользователь ввел нужную сумму валюты которую он хочет купить, он щелкает на любую валюту в списке и перемещается в экран “Обмен”.

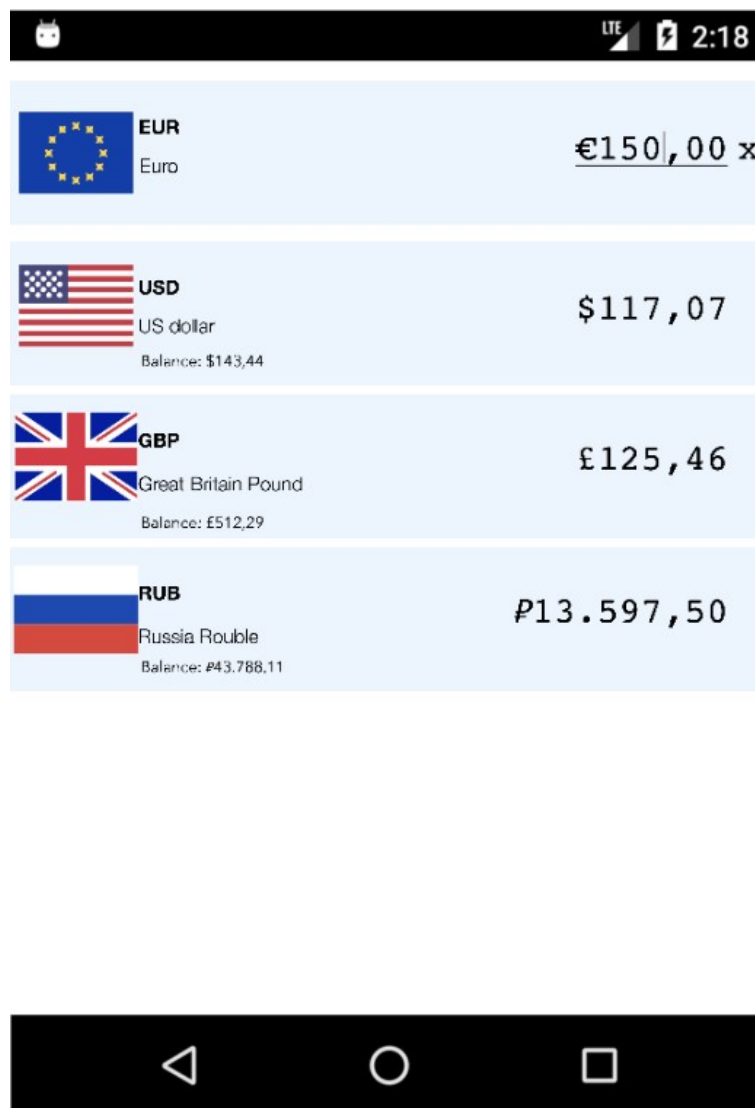


Figure 2: Экран "Валюты: ввод суммы" – пользователь выбрал “евро” и ввел сумму 150

Если пользователь начал вводить сумму для покупки, справа от суммы нужно показать кнопку “х”, по нажатию на которую сумма выбранной валюты сбрасывается в 1, “х” убирается и экран опять переключается в режим “Валюты: список”.

При установке приложения у пользователя на рублевом счету уже имеется 75000 рублей.

## Экран “Обмен”

В экране “Обмен” присутствуют сумма валюты для покупки, соответствующая ей сумма в валюте которой покупка должна быть оплачена, обменный курс и кнопка “Купить”.

По нажатию на кнопку “Купить” транзакция сохраняется в таблицу transactions, в таблице accounts обновляются строки для пары валют для которых произошел обмен и пользователь возвращается на экран “Валюты” в режим “Валюты: список”.



Figure 3: Экран “Обмен”

## Экран “Транзакции”

На экране “Транзакции” показать список совершенных транзакций с указанием пар валют, количества и даты/времени операции из таблицы transactions (класс TransactionDbO). Добавьте иконку для открытия экрана “Транзакции” на экран “Валюты” как удобно.

## Дополнительная информация

Вам могут понадобиться флаги стран - пример для рубля. При желании флаги государств можно сохранить в ресурсы.

В заготовку приложения уже добавлены compose, retrofit2, coil, hilt, room, recyclerview, navigation библиотеки, но вы можете добавить любые другие.

Вносить изменения в класс RemoteRatesServiceImpl запрещается.

При создании пользовательского интерфейса используйте фрагменты или compose.

Если при работе над приложением были приняты интересные нестандартные дизайн-решения, опишите их в файле README.md

Если есть задор, можно:

- Всячески улучшать пользовательский опыт.
- Воспользоваться библиотеками для внедрения зависимостей.
- Создать приложение в соответствии с чистой архитектурой.
- Написать тесты для бизнес-логики.
- Воспользоваться навигацией.

Каждое улучшение помимо базовой функциональности будет учтено.

Решением будет являться ссылка на архив с исходным кодом приложения выполняющего задачи выше в облачном хранилище (google drive, yandex диск и тп).

По решению будет предоставлена обратная связь.