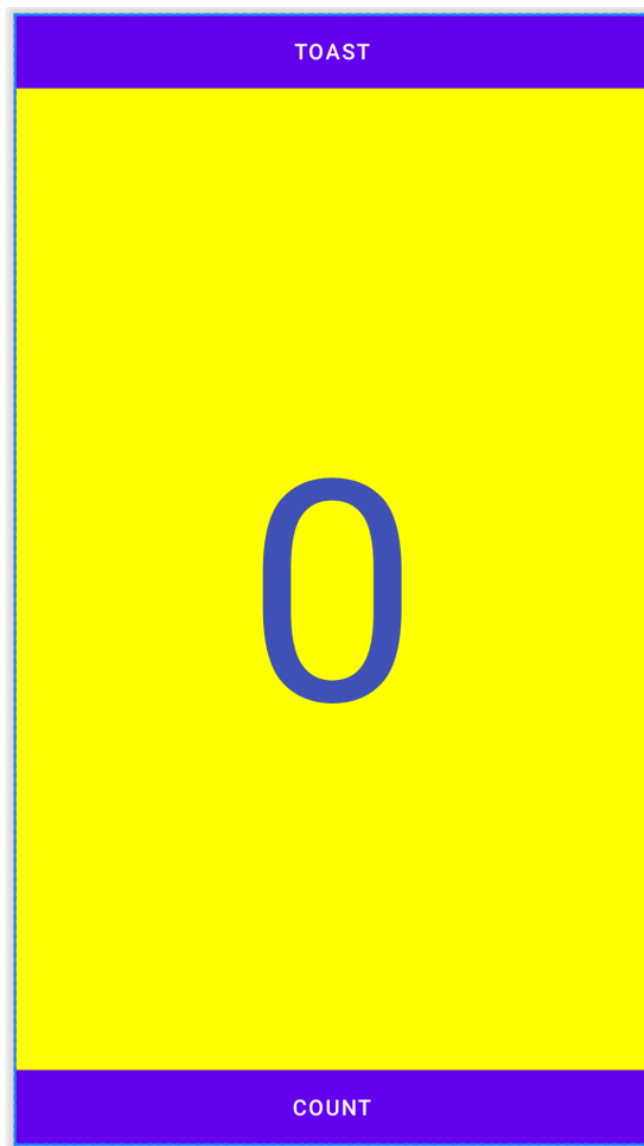


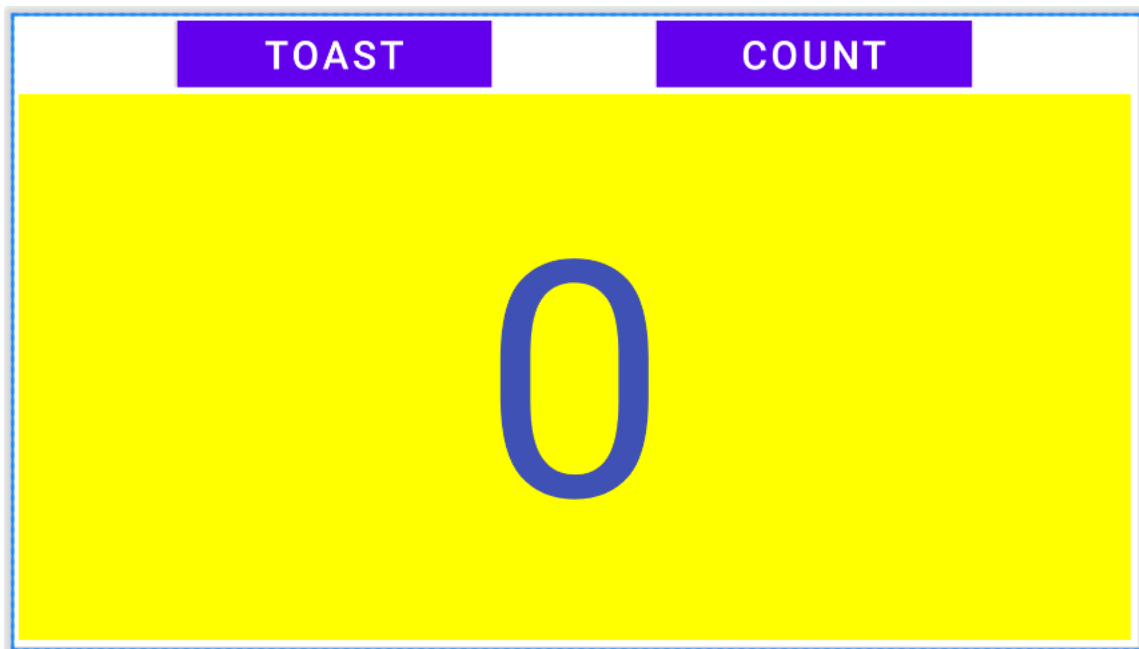
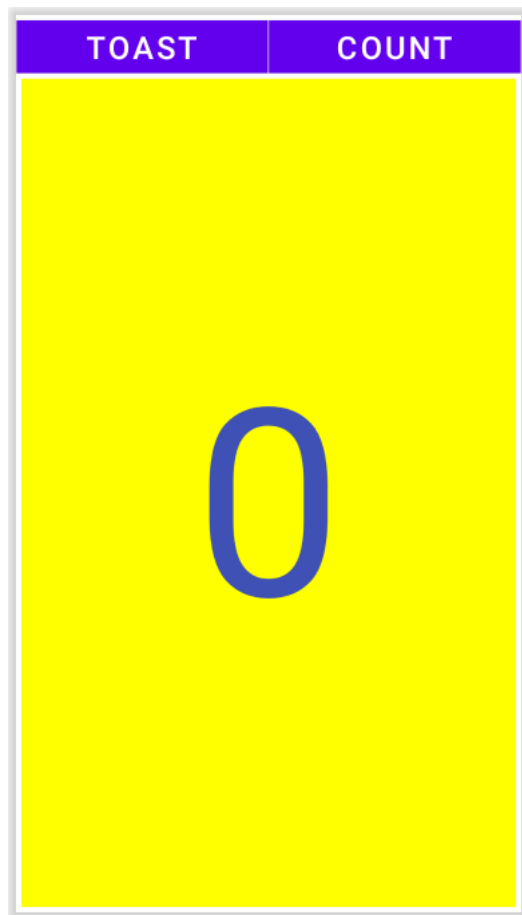
Phát triển ứng dụng di động (Tuần 01: Introduction)

Bài 01

Ứng dụng HelloToast bao gồm hai Button và một TextView. Khi người dùng chạm vào Nút đầu tiên, nó sẽ hiển thị một thông báo (Toast) trên màn hình. Nhấn vào Nút thứ hai sẽ tăng bộ đếm được hiển thị trong TextView, bắt đầu từ 0.

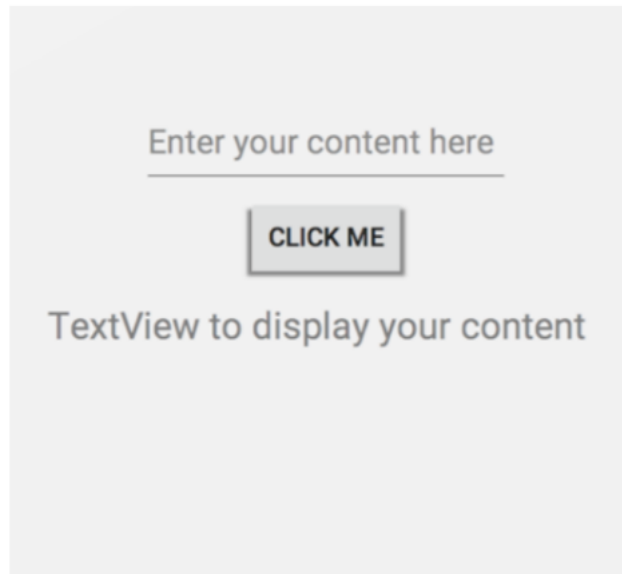


Tạo biến thể cho giao diện trên kích thước màn hình lớn (tablet) cho ứng dụng HelloToast theo mô tả như trong hình minh họa bên dưới.



Bài 02

Tạo ứng dụng như hình bên dưới



Khi click nút **CLICK ME**:

- Nếu EditText đang rỗng, hiển thị thông báo Toast “vui lòng nhập thông tin”.
- Ngược lại, nội dung trong EditText sẽ được hiển thị ở TextView bên dưới, nội dung của EditText sau đó sẽ được clear.

Nâng cao:

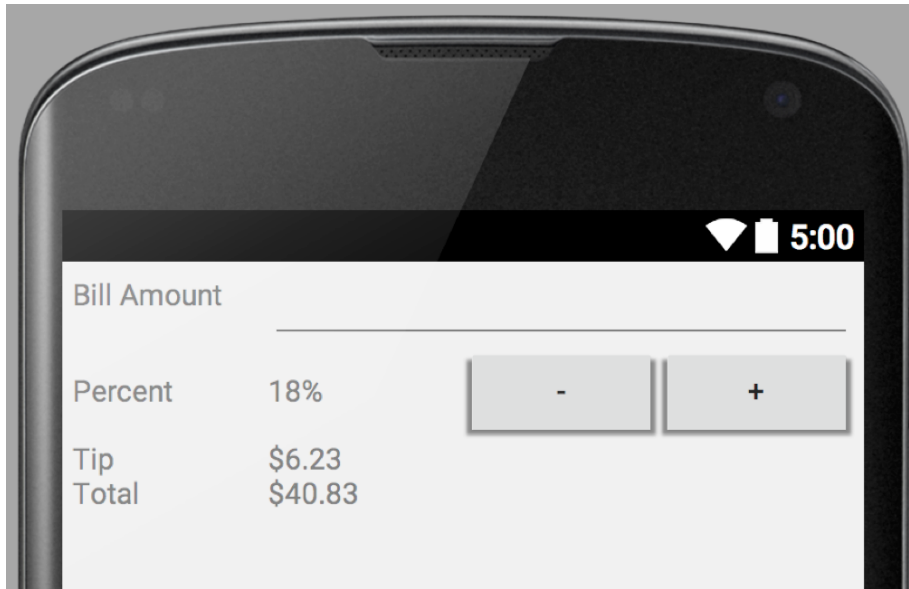
- Nếu người dùng gõ chữ “OFF” thì button sẽ tự động bị disable (không click được).
- Nếu người dùng nhập “ON” thì button sẽ tự động được enabled (cho phép click).
- Lưu ý, trạng thái của button sẽ được thay đổi ngay khi người dùng gõ chữ “ON” hoặc “OFF”.

Gợi ý làm bài:

- Để viết sự kiện click cho button, sử dụng `setOnClickListener` cho đối tượng Button đó, truyền vào một đối tượng thuộc kiểu `View.OnClickListener`.
- Để bắt sự kiện thay đổi text cho EditText, sử dụng phương thức `addTextChangedListener()` cho đối tượng EditText đó.
- Lấy và gán giá trị của TextView, EditText thông qua phương thức `getText()`, `setText()`.
- Để hiện thông báo Toast, sử dụng phương thức `Toast.Show()`.
- Để disable/enable một View (Button, EditText, TextView,...) sử dụng phương thức `setEnabled()`;

Bài 03

Tạo ứng dụng như hình bên dưới



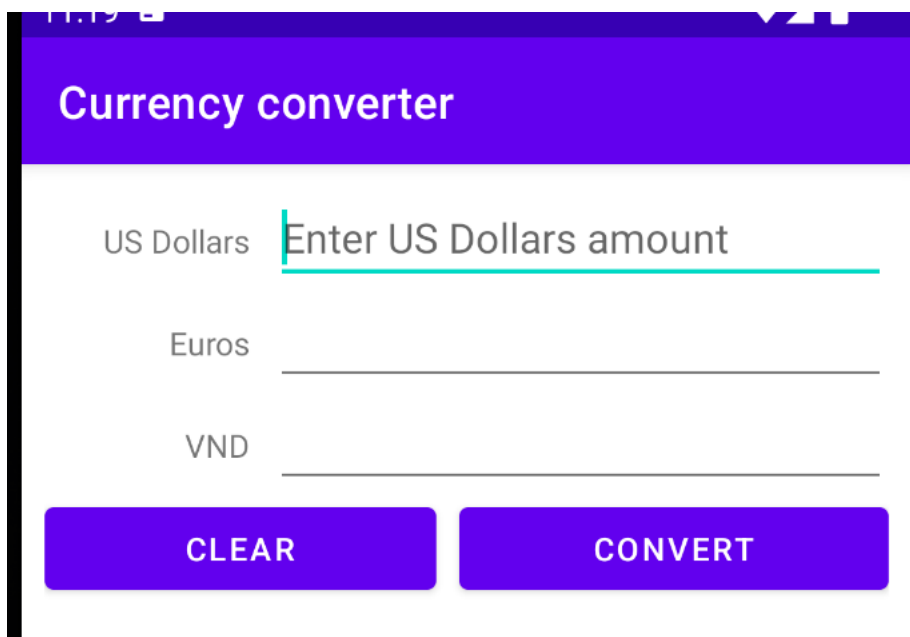
- Người dùng nhập số tiền cần thanh toán vào mục **Bill Amount** tương ứng
- **Percent** tiền tip mặc định là **10%** hóa đơn, giá trị của **Tip** và **Total** sẽ được cập nhật tự động khi người dùng thay đổi giá trị **Bill Amount**.
- Người dùng có thể nhấn nút “+”, “-” để tăng hoặc giảm % tiền tip và giá trị của **Tip** và **Total** sẽ được cập nhật tự động theo sự thay đổi này.
- **Percent** tối thiểu là 5%, khi giảm đến 5% thì button “-” sẽ bị disabled

Hướng dẫn làm bài:

- Để bắt sự kiện thay đổi text cho EditText, sử dụng phương thức `addTextChangedListener()` cho đối tượng EditText đó.
- Lấy và gán giá trị của TextView, EditText thông qua phương thức `getText()`, `setText()`.
- Để disable/enable một View (Button, EditText, TextView,...) sử dụng phương thức `setEnabled()`;

Bài 04

Tạo ứng dụng như hình bên dưới




The screenshot shows a mobile application titled "Currency converter" with a blue header. Below the header, there are three input fields for currency conversion. The first field is labeled "US Dollars" and contains the placeholder text "Enter US Dollars amount". The second field is labeled "Euros" and is empty. The third field is labeled "VND" and is empty. At the bottom of the form, there are two blue buttons: "CLEAR" and "CONVERT".

- Người dùng chỉ được phép nhập vào số tiền bằng US Dollar, các EditText Euros và VND chỉ readonly.
- Khi nhấn nút Convert, ứng dụng sẽ tự chuyển đổi sang đơn vị tiền tệ Euro và VND và hiển thị lên EditText tương ứng.
- Khi nhấn nút Clear, giá trị trong các EditText sẽ bị xóa.

Bài 05

Tạo ứng dụng như hình bên dưới

- Khi nhấn nút Click here to see Results, TextView sẽ hiển thị nội dung phản ánh các lựa chọn của người dùng.
- Làm hai trường hợp cho cả CheckBox và RadioButton. Sau khi làm xong trường hợp với Checkbox có thể copy lại code và chỉnh sửa lại (cho nhanh) để làm cho bài RadioButton.

 **Android Checkboxes**

List of Checkboxes for selection

☒ Android

☒ iOS

☐ Windows

☐ RIM

Click here to see Results


The following were selected...

Android : true

iOS : true

Windows : false

RIM : false

 **Android RadioButtons**

List of Radio Buttons for selection

☐ Android

☒ iOS

☐ Windows

☐ RIM

Click here to see Results

Hướng dẫn làm bài:

- Tạo file xml để thiết kế giao diện cho từng dòng. Để biết một CheckBox/RadioButton có được chọn hay không, sử dụng phương thức isChecked() của đối tượng đó.
- Về nguyên tắc, chỉ một RadioButton được chọn tại một thời điểm. Để làm được điều đó, các RadioButton phải được đặt trong cùng một nhóm. Sử dụng thẻ <RadioGroup>.