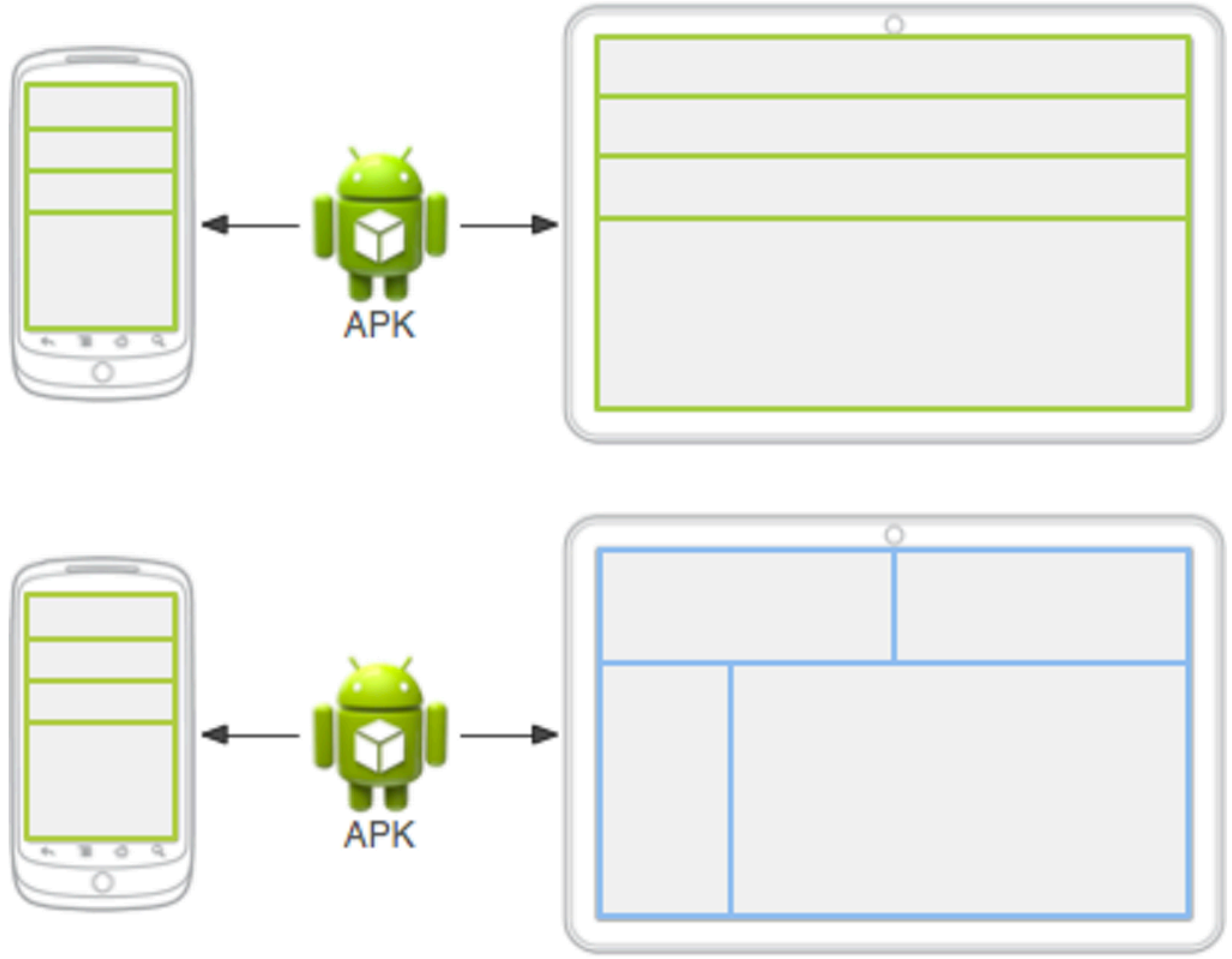


MOBILE APPLICATION DEVELOPMENT

ANDROID (2017)

LECTURE 15: RESOURCES

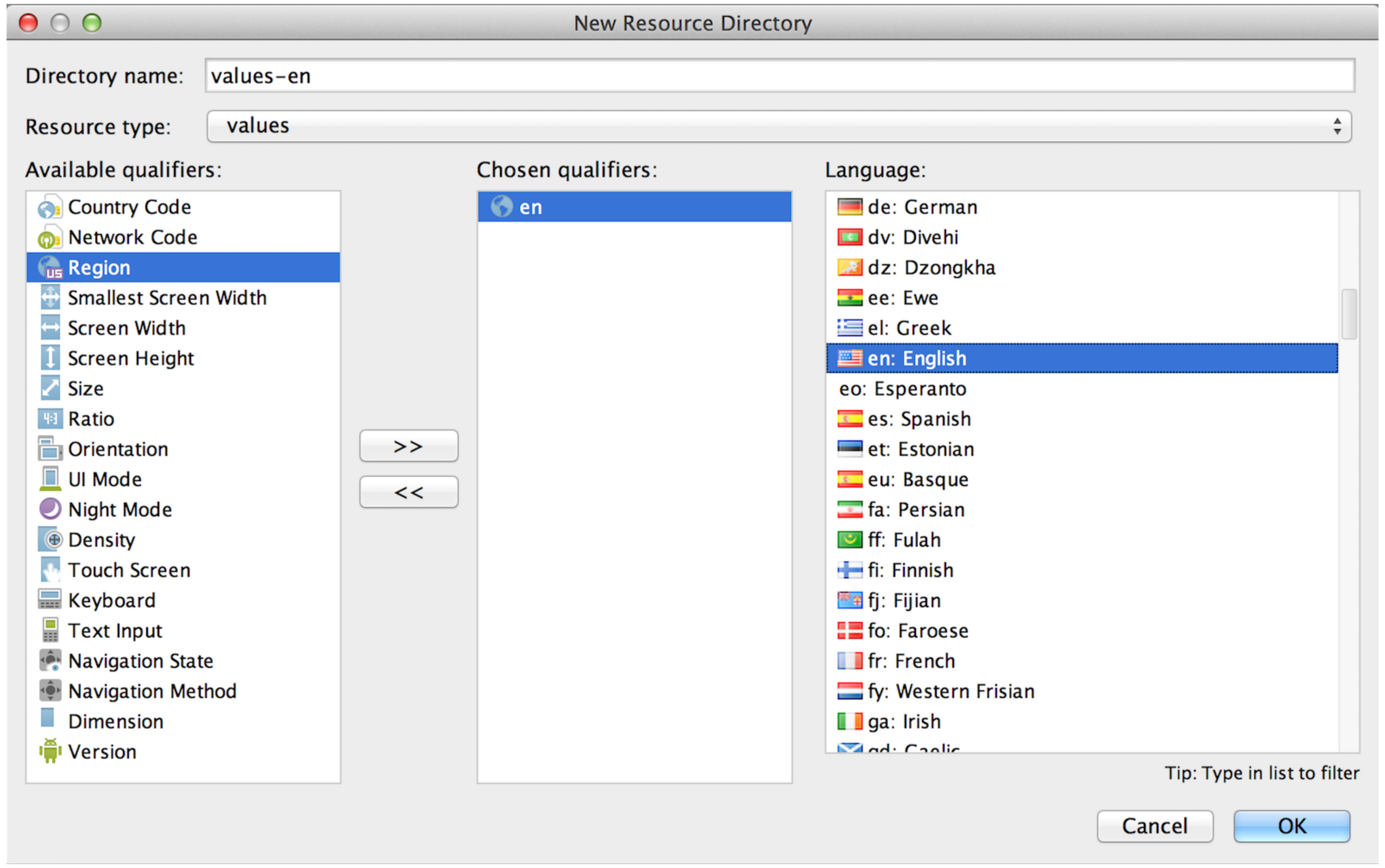
RESOURCES

- Android encourages developers to separate application logic from the storage of predefined resources such as text, images, layouts, colors, and so on.
- Resources are stored within Android projects in the `res` folder, which contains subfolders for a variety of resource types and configurations.
  - Enables a structure with which a variety of devices or situations may be supported.
  - Defines a predictable organization for potentially large collections of resources.
- Resources are typically accessed in code through the use of the R class.

EXAMPLE RESOURCES BY DIRECTORY

- The `drawable/` directory contains XML or bitmap representations of visual content.
- The `mipmap/` directory contains multi-resolution drawables which define app icons.
- The `layout/` directory contains XML representations of layouts.
- The `values/` directory contains simple values such as colors, integers, and strings.
- The `xml/` directory contains arbitrary XML files which can be parsed at runtime.
- The `font/` directory contains fonts which define text appearance in an application.
- There are a variety of other directories besides the ones listed here.

RESOURCE QUALIFIERS



New Resource Directory

Directory name: values-en

Resource type: values

Chosen qualifiers: en

Language: en: English

- Qualifiers allow programmers to specify different resources for different situations.
- Android has a large set of available qualifier types, and will automatically choose appropriate resources at runtime based on their qualifiers.
- Can specify different layouts for different screen sizes or orientations, different images for different locales, different color schemes for night mode, and a variety of other variations that override the app's default resources in certain instances.

QUALIFIER EXAMPLE: LOCALIZATION

- Text found in `res/values/strings.xml` serves as the default values of any defined Strings, and should represent the text you expect most of your users will be able to read.
- Text resources can be redefined in other directories which use qualifiers to indicate that the new definitions should override the defaults in some locales.
  - Example: text defined in `res/values-ja/strings.xml` will be used instead of text defined in `res/values/strings.xml` if the user's device is set to use Japanese as its locale.
- Android Studio provides a built-in editor for managing localizations of text.

CONFIGURATIONS

- Android makes use of information provided by a Context's Configuration to determine which resources to choose based on their qualifiers.
- Configurations can be obtained through the `.resources.configuration` accessor on a Context and describe the current state of the device the application is running on.
  - Contain information about the device's current locale, screen dimensions, and so on - all of the information which can be used as qualifiers on resources.
  - Can be used to make determinations about device functionality and state beyond simply selecting appropriate resources.

RESOURCES EXAMPLE: DISPLAY DENSITY

- Android specifies most drawing information in device-independent pixels. These are indicated by the notation `dp` after a numeric size value.
- One `dp` represents the size of a pixel on a medium-density screen at 160dpi, thus being 1/160 of an inch wide. The density of the device's display relative to this default scale can be obtained with the `.resources.displayMetrics.density` accessor of a Context.
- Can convert between `dp` and pixels with the following operations:

```
val pixels: Int = dp * context.resources.displayMetrics.density
val dp: Int = pixels / context.resources.displayMetrics.density
```