Android Development - Assignment 1

Exercise 1

Question 1

The lines are:

```
if (index >= _suggestions.length) {
  _suggestions.addAll(generateWordPairs().take(10));
}
```

Each time we are out of pre-generated word pairs, these lines will supply another 10 new pairs. In addition, we haven't specified *itemCount* parameter for the builder. Therefore, it will ask the *itemBuilder* for new items with infinitely.
If we remove these two (three) lines, the builder still will be called to supply 11-th, 12-th elements and so on, but our pre-generated list has only ten pairs. This situation will cause an error.

Question 2

We can use *ListView.separated* instead of *ListView.build* to get the desired effect. In my opinion, it's better to use *ListView.separated* because it improves code readability (the method explicitly says we want to have list of items separated with dividers) and it's better to use something already done (apparently by professionals and with many error-checking) instead of reinventing the wheel.

Despite this, *List.build* may still be useful if we work with infinite lists, or we want to have some strange rules for dividers (divide in groups of three, divider between each five entries etc.).

Question 3

onTap() logic requires updating the UI of our app by changing a state of a stateful widget. To notify the flutter system about the changes and to make it fetch them correctly, we have to call the *setState()* method and provide it with a function that executes these changes.

Exercise 2

Question 1

MaterialApp widget represents an application that uses Material Design. It wraps and adds material-design specific functionality and styles.
Examples of properties:

- *title* - a title of the application that will be showed above the task manager's app snapshots.
- *theme* - defines visual properties, like colors, fonts and shapes, for widgets in this app.
- *home* - the default route of the app. It will be displayed first when the application is started.

Question 2

key is a property that defines an identifier for Widgets or Elements. It is used by flutter to distinguish between Elements of the same Widget and with the same parent.
Flutter maintains an elements' tree in parallel to the widgets tree. Elements are the ones who are holding the State. Each time we are changing the widgets tree's structure, like deleting elements from a list, reordering them, flutter tries to change the elements' tree accordingly. The only connection that it has between a widget and an element is a type of the widget. Therefore, if we reorder two list entries (same widget) flutter won't change the elements tree, because each element still has a valid corresponding widget in the widgets tree. Keys helps the flutter to distinguish between widgets of the same class of the same parent in situations like above.

Memes

What the hell is this?

Developers
Native Android
I don't want to play with you anymore
memeking.co.il

כשאתה אומר למרצה אחראי בשפות
שהשתמשו במושגי הקורס בתרגול של
פיתוח אנדרואיד: