



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Факультет компьютерных наук  
Департамент программной инженерии

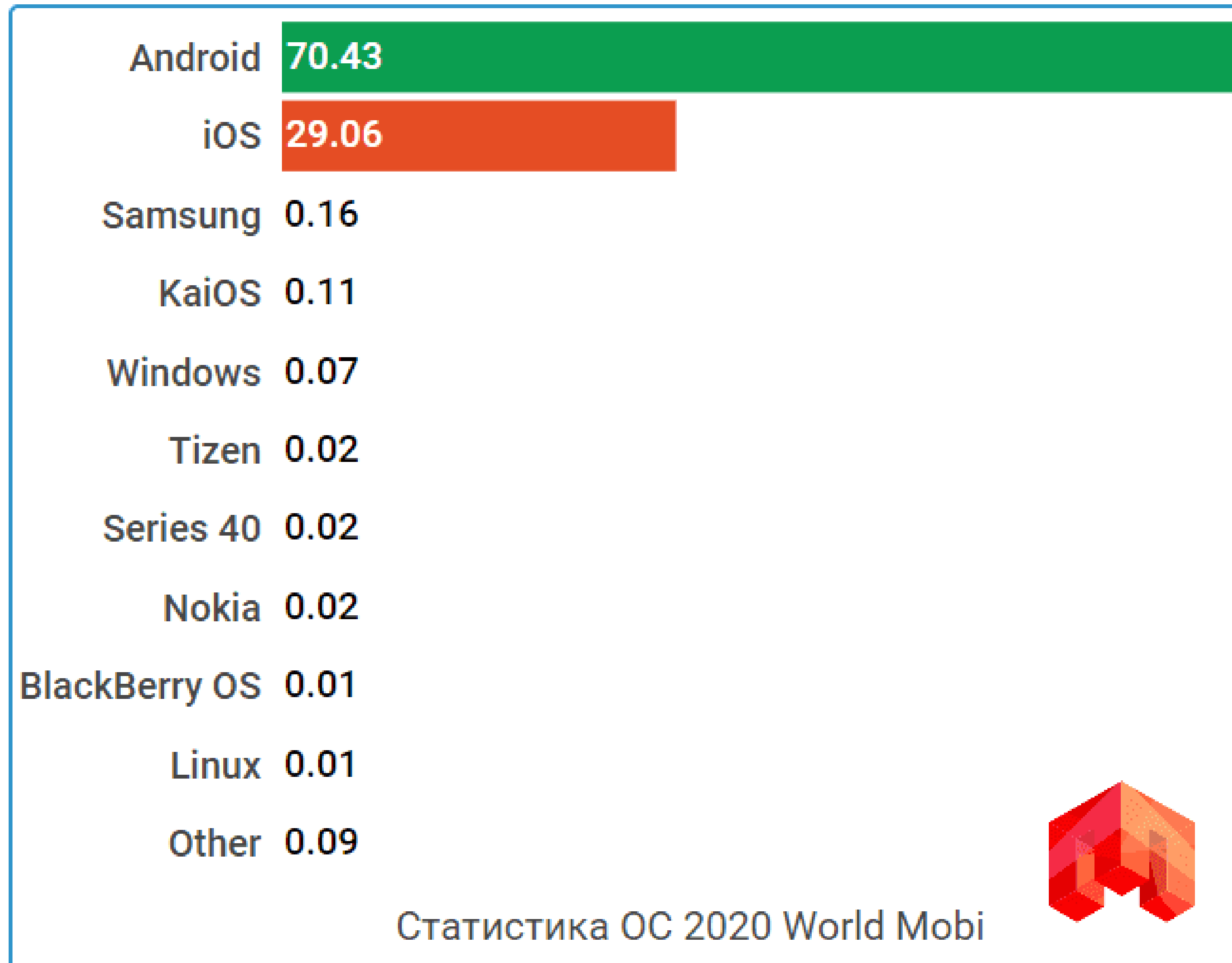
# ОБЕСПЕЧЕНИЕ КАЧЕСТВА И ТЕСТИРОВАНИЕ

Семинар 14: Тестирование мобильных приложений

Москва, 2020

# МОБИЛЬНЫЕ ОС

Android	Unknown
71.18%	0.14%
iOS	KaiOS
28.19%	0.13%
Samsung	Windows
0.24%	0.03%





# ANDROID STUDIO

Android Studio — интегрированная среда разработки для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Позволяет писать приложения как на Java\Kotlin по отдельности так и в виде смешанных проектов.



## ESPRESSO

Espresso - это фреймворк для тестирования. Раньше распространялась как отдельная библиотека, но начиная с версии 2.0, Espresso стала частью репозитория Android Support Repository, что позволило быстро и удобно добавить поддержку тестов в ваш проект. Работает на устройствах и эмуляторе.

Перед тестированием следует отключить системные анимации. В настройках Settings | Developer options есть три опции для анимации, которые желательно отключить.

## КЛАССЫ

- Espresso – основной класс. Содержит в себе статические методы, такие как нажатия на системные кнопки (Back, Home), вызвать/спрятать клавиатуру, открыть меню, обратиться к компоненту
- ViewMatchers – позволяет найти компонент на экране в текущей иерархии
- ViewActions – позволяет взаимодействовать с компонентом (click, longClick, doubleClick, swipe, scroll и т.д.)
- ViewAssertions – позволяет утвердить состояние компонента



## ПОДКЛЮЧЕНИЕ

```
dependencies {  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.0'  
}
```



# ПРИМЕР

```
@RunWith(AndroidJUnit4.class)
@LargeTest
public class ChangeTextBehaviorTest {
    private String stringToBetyped;
    @Rule
    public ActivityScenariRule<MainActivity> activityRule
        = new ActivityScenarioRule<>(MainActivity.class);
    @Before
    public void initValidString() {
        // Specify a valid string.
        stringToBetyped = "Espresso";
    }
    @Test
    public void changeText_sameActivity() {
        // Type text and then press the button.
        onView(withId(R.id.editTextUserInput))
            .perform(typeText(stringToBetyped), closeSoftKeyboard());
        onView(withId(R.id.changeTextBt)).perform(click());
        // Check that the text was changed.
        onView(withId(R.id.textToBeChanged))
            .check(matches(withText(stringToBetyped)));
    }
}
```





# ПРИМЕР

```
// Щелчок кнопки у первой активности
public void onClick(View view) {
    EditText nameEditText = (EditText) findViewById(R.id.editTextName);
    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
    intent.putExtra("name", nameEditText.getText().toString());

    startActivity(intent);
}

// Запускается вторая активность
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);

    String user = "";
    user = getIntent().getExtras().getString("name");

    TextView infoTextView = (TextView)findViewById(R.id.textViewInfo);
    infoTextView.setText("Hello " + user);
}
```





# TEST

```
public class MainActivityTest {  
  
    @Rule  
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(  
        MainActivity.class);  
  
    @Before  
    public void setUp() throws Exception {  
  
    }  
  
    @Test  
    public void onClick() throws Exception {  
        onView(withId(R.id.editTextName)).perform(typeText("Barsik"));  
        onView(withId(R.id.button)).perform(click());  
        onView(withId(R.id.textViewInfo)).check(matches(withText("Hello Barsik")));  
    }  
}
```



## ПОИСК ПО НЕ УНИКАЛЬНЫМ ID

```
onView(allOf(withId(R.id.button_signin), withText("Sign-in")));
```

Или

```
onView(allOf(withId(R.id.button_signin), not(withText("Sign-out"))));
```



# ИНТЕГРАЦИЯ С WEBDRIVER

```
dependencies {  
  androidTestImplementation 'androidx.test.espresso:espresso-web:3.1.0'  
}
```



# ИНТЕГРАЦИЯ С WEBDRIVER

```
@LargeTest
@RunWith(AndroidJUnit4.class)
public class WebViewActivityTest {

    private static final String MACCHIATO = "Macchiato";
    private static final String DOPPIO = "Doppio";

    @Test
    public void typeTextInInput_clickButton_SubmitsForm() {
        // Lazily launch the Activity with a custom start Intent per test
        try (ActivityScenario.launch(withWebFormIntent())) {

            // Enable JavaScript.
            onView(withId(R.id.web_view)).perform(forceJavaScriptEnabled());

            // Selects the WebView in your layout.
            // If you have multiple WebViews you can also use a
            // matcher to select a given WebView, onView(withId(R.id.web_view)).
            onView(withId(R.id.web_view))
                // Find the input element by ID
                .withElement(findElement(Locator.ID, "text_input"))
```

```
        // Clear previous input
        .perform(clearElement())
        // Enter text into the input element
        .perform(DriverAtoms.webKeys(MACCHIATO))
        // Find the submit button
        .withElement(findElement(Locator.ID, "submitBtn"))
        // Simulate a click via JavaScript
        .perform(webClick())
        // Find the response element by ID
        .withElement(findElement(Locator.ID, "response"))
        // Verify that the response page contains the entered text
        .check(webMatches(getText(), containsString(MACCHIATO)));
    }
}
```



## ЛИТЕРАТУРА

1. <https://developer.android.com/training/testing/espresso/>
2. <https://developer.android.com/training/testing/ui-testing/espresso-testing>
3. <https://medium.com/mindorks/android-testing-part-1-espresso-basics-7219b86c862b>
4. <http://www.vogella.com/tutorials/AndroidTestingEspresso/article.html>

**СПАСИБО! ВОПРОСЫ?**



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ