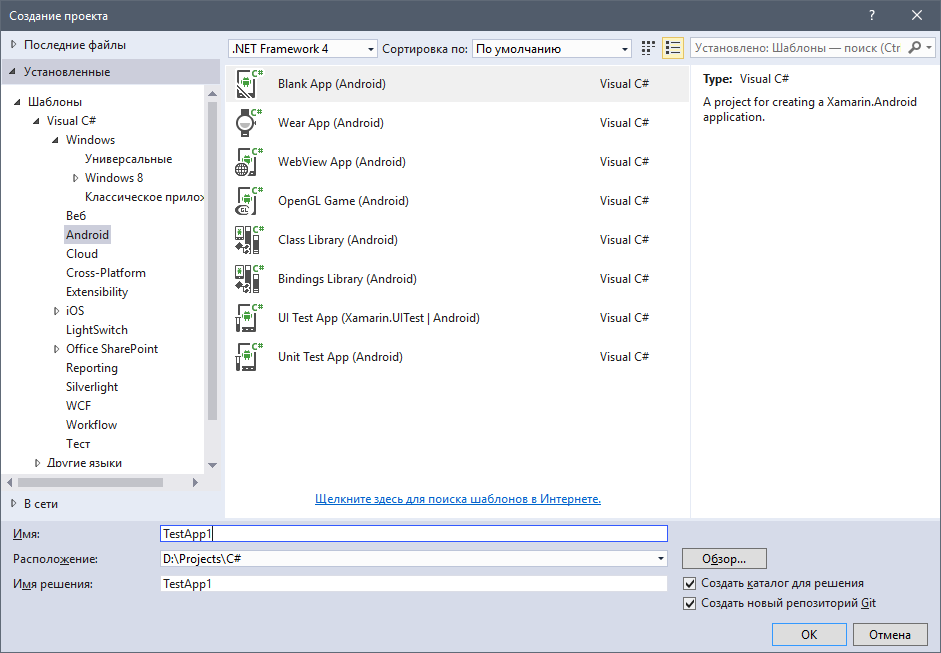
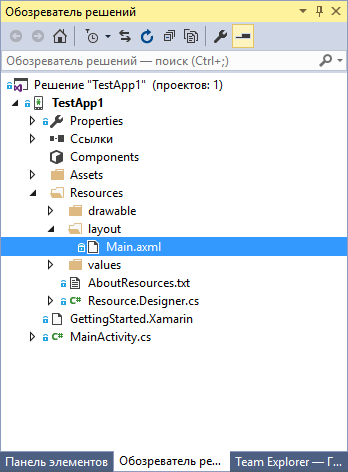
**Лабораторная работа №1. Простая программа. Отладка.**

**Создание проекта**

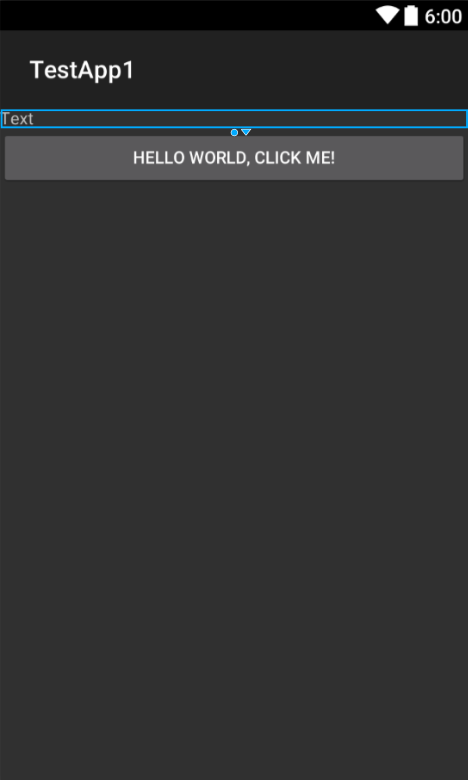
Создайте проект: Visual Studio → Android → Blank App (Android). Введите уникальное имя проекта и выберите его расположение:



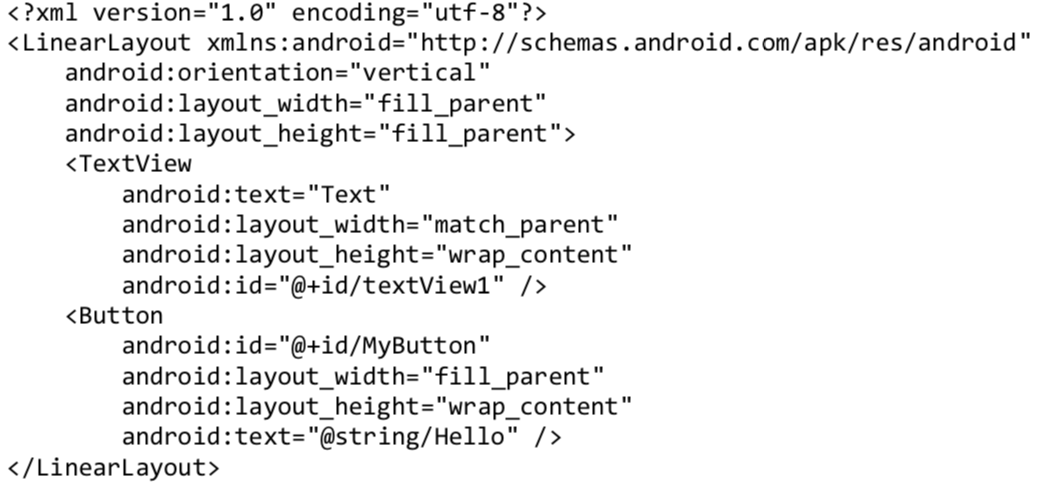
После создания проекта в Обозревателе решений в разделе Resources → layout найдите файл Main.axml и откройте его двойным щелчком:



Этот файл содержит главную активность приложения – то, что пользователь будет видеть на экране в конкретный момент времени. По умолчанию новая активность содержит только одну кнопку. Добавьте текстовую надпись (компонент TextView) – просто перетащите его из Панели элементов на активность. Мышкой можно перетаскивать элементы по активности, меняя их взаимное расположение:



В нижней части окна дизайнера расположены вкладки Design и Source. Первая открывает уже знакомое нам окно визуальной проектировки активности, вторая переключается на XML-формат представления активности. Созданная нами активность выглядит в XML-формате следующим образом:

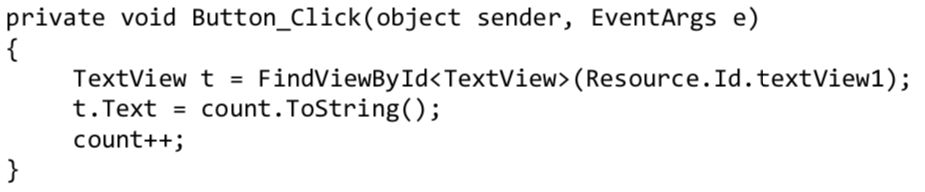


Разметка представляет собой иерархию элементов: каждый тег – это элемент, а его атрибуты – это свойства элемента. Разметку можно редактировать, все изменения будут отражены при переходе на вкладку Design или запуске приложения.

Важным свойством элемента является его идентификатор (android:id). Используя идентификатор, присвоенный элементу в разметке, можно затем обращаться к нему из кода программы. Xamarin автоматически генерирует идентификаторы для каждого нового элемента управления, который добавляется на активность (например, у добавленного текстового поля это @+id/textView1), но лучше менять их на более соответствующие контексту. Сделать это можно в окне Свойства, предварительно выделив мышкой элемент управления в активности, или непосредственно в разметке активности.

Откройте в Обозревателе решений файл MainActivity.cs – в этом файле находится класс MainActivity, содержащий код активности. В данный момент там находится только один переопределенный метод OnCreate, который выполняется при создании активности.

Добавьте в класс MainActivity следующий метод:



В этом методе сначала ищется элемент управления textView1. В отличие от .NET, в Xamarin (и в Android) в классе объекты не создаются автоматически для каждого элемента на активности, их нужно находить с помощью метода FindViewById, параметром которого является идентификатор элемента управления, заданный в свойствах элемента. Все идентификаторы элементов перечислены в классе Resource.Id.

Добавленный метод пока не вызывается при нажатии на кнопку. Чтобы это происходило, модифицируйте метод OnCreate: оставьте строку вызов метода SetContentView, а код для работы с кнопкой замените на следующий:

Button button = FindViewById<Button>(Resource.Id.MyButton);

button.Click += Button\_Click;

Попробуйте запустить приложение.

**Запуск и отладка приложения**

Наилучшим способом отладки является запуск приложения на физическом устройстве. Это позволяет правильно оценить его производительность. Однако не всегда возможно иметь для тестирования устройства с разными версиями ОС Android, с разными аппаратными характеристиками и т.д. Поэтому в состав средств разработки для мобильных устройств обычно входят эмуляторы.

Xamarin поддерживает несколько видов эмуляторов:

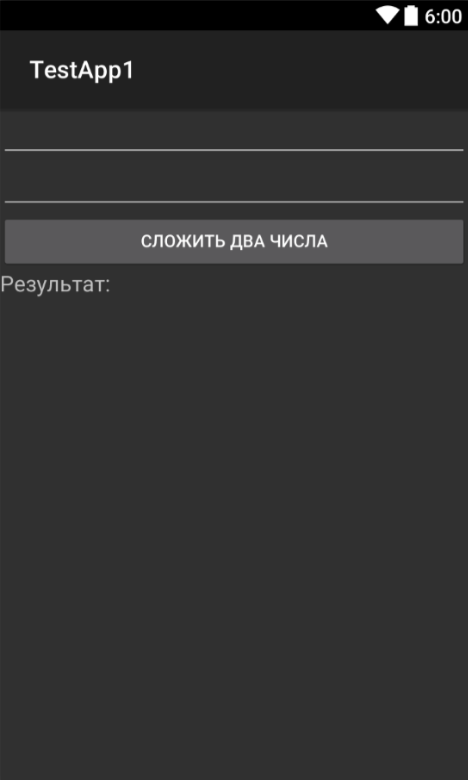
* Android Virtual Device (AVD). Это эмулятор Android-устройств, входящий в состав Android SDK. Позволяет создавать виртуальные устройства с требуемыми характеристиками. Эмуляция устройства осуществляется «с нуля», поэтому обычно достаточно медленная.
* Visual Studio Emulator. Этот эмулятор устройств позволяет отлаживать приложения как для Android, так и для других платформ. Входит в состав Visual Studio. Не позволяет создавать устройства вручную, но предлагает готовые настроенные образы устройств с различными характеристиками. Обычно более производителен, чем AVD.

Для запуска приложения на физическом устройстве нужно установить USB-драйвер, предназначенный для этой модели. Данный драйвер обеспечивает возможность отладки, а не просто передачи файлов. Скачать такой драйвер обычно можно на сайте производителя устройства.

Помимо установки драйвера, на устройстве нужно включить режим разработчика. На различных моделях телефона это может делаться по-разному: как правило, в настройках телефона нужно выбрать раздел «О системе», и семь раз коснуться пункта меню с номером сборки (или другого пункта, указанного в документации). В некоторых моделях для включения режима разработчика есть отдельная команда в меню. В разделе для разработчика нужно включить отладку по USB, а также другие опции, которые могут быть необходимы для запуска программы на устройстве.

**Простой калькулятор**

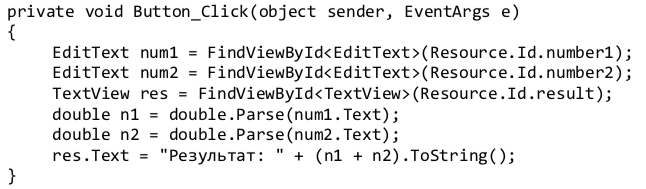
Модифицируйте приложение, добавив в него два поля типа EditText (в Панели элементов их несколько, нам нужны элементы типа Number, предназначенные для ввода чисел):



Измените идентификаторы полей ввода на @+id/number1 и @+id/number2 соответственно, а идентификатор поля с результатом – на @+id/result.

Изменить текст в поле TextView можно прямо в панели Свойства, для этого достаточно поправить свойство Text. Но делать так не рекомендуется: жестко заданные строки будут всегда отображаться на одном языке, это вызовет проблемы при локализации приложения на другие языки. Поэтому рассмотрим, как это правильно делается на примере кнопки. Поле Text у нее содержит значение @string/Hello – это означает, что в строковых ресурсах ищется параметр Hello, который и содержит искомую строку на нужном языке. Откройте файл Strings.xml в разделе Resources → Values, найдите там строку Hello и измените текст на «Сложить два числа».

Измените в коде класса код, выполняющийся при нажатии на кнопку, на следующий:



С учетом изложенной выше информации приведенный код должен быть понятен.

Запустите приложение. Введите числа в поля, нажмите на кнопку. Убедитесь, что значения суммируются и выводятся в активность.

**Задание**

Доработайте программу, добавив в неё другие математические операции: вычитание, деление, умножение, возможно, и другие математические действия по желанию.

**Оформление отчета**

Отчет должен состоять из поэтапного описания выполняемых действий. Отдельно представляется листинг и скрины, работающей программы. Отчет должен содержать титульный лист. Текст отчета должен быть оформлен по требованиям, предъявляемым к курсовым и ВКР. Каждый отчет защищается студентом индивидуально.