

Лабораторная работа №1

Знакомство с XAML

Цель работы: знакомство с приложением .Net MAUI и с основами языка XAML.

Задача работы: научиться создавать простые интерфейсы приложений с помощью фреймворка .Net MAUI.

Время выполнения работы: 6 часов (3 занятия)

Результат выполнения работы: программа, обеспечивающая функционал согласно заданию.

1. Задание для всех вариантов

а. Установите на компьютер VisualStudio для создания проектов .Net MAUI см. <https://learn.microsoft.com/en-us/dotnet/maui/get-started/installation?tabs=vswin>

Для отладки мобильных версий:

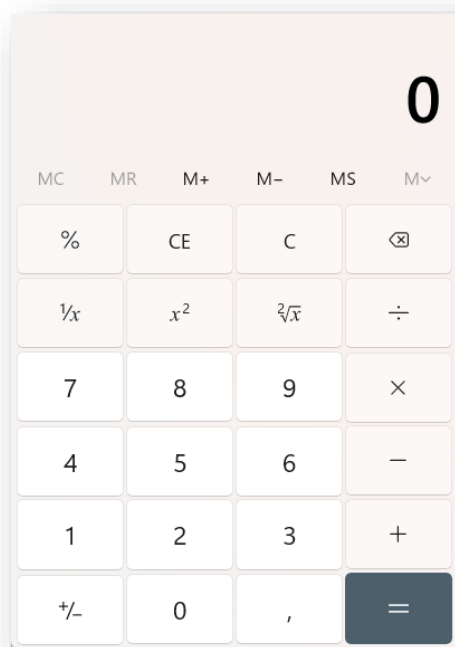
- использование эмулятора Android – см. <https://learn.microsoft.com/en-us/dotnet/maui/android/emulator/hardware-acceleration> и <https://learn.microsoft.com/en-us/dotnet/maui/android/emulator/device-manager>
- использование физического устройства Android – см. <https://learn.microsoft.com/en-us/dotnet/maui/android/device/setup>
- отладка на iOS – см <https://learn.microsoft.com/en-us/dotnet/maui/ios/cli>

Для Android и iOS можно использовать Rider версии не ниже 2022.2

б. Создайте приложение .Net MAUI

с. Ознакомьтесь со структурой проекта

д. Добавьте в проект страницу (ContentPage), реализующую функционал калькулятора Windows:



е. Для события «click» всех цифровых кнопок использовать ОДИН обработчик (один метод в «code-behind»).

ф. Добавить функцию согласно индивидуальному заданию.

г. Страница должна выбираться во Flyout меню навигации Shell

2. Индивидуальные задания

1. Вычисление x^y
2. Вычисление $|x|$
3. Вычисление 10^x
4. Вычисление e^x
5. Вычисление остатка от деления (**mod**)
6. Вычисление **log(x)**
7. Вычисление 2^x
8. Вычисление **sin(x)** (x в градусах)
9. Вычисление **ближайшего целого**
10. Вычисление **площади круга** для заданного радиуса

3. Вопросы для самопроверки

- 1) Какие есть панели компоновки представлений на странице?
- 2) Как определить в разметке XAML строки и столбцы Grid
- 3) Как можно задать высоту и ширину элемента?
- 4) Как указать что элемент находится в 1-м столбце 0-й строки сетки Grid?
- 5) Как обрабатываются события элементов страницы?
- 6) Как прочитать/установить текст элемента Label в code-behind?
- 7) Как подключить пространство имен XAML?
- 8) Где подключаются ресурсы, общие для всего приложения
- 9) Для чего используется Shell?
- 10) Какие объекты могут находиться внутри объекта Shell?
- 11) Как запустить метод по клику кнопки?
- 12) Как перейти на страницу с помощью класса Shell?
- 13) Как зарегистрировать маршрут к странице?
- 14) Чем отличается маршрут «/page» от «///page»?
- 15) Для чего используется класс ContentPage?

Лабораторная работа №2

Управление UI из вторичного потока

Цель работы: углубленное знакомство с приложением .Net MAUI и с основами языка XAML.

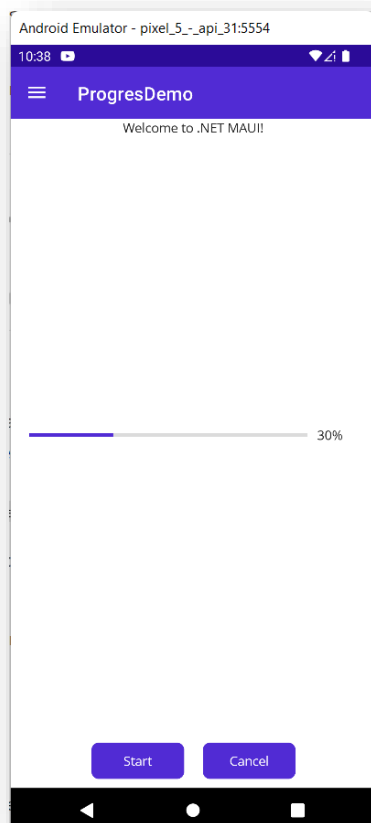
Задача работы: научиться управлять интерфейсом приложений из вторичного потока.

Время выполнения работы: 2 часа (1 занятие)

Результат выполнения работы: программа, обеспечивающая функционал согласно заданию.

1. Задание

В проект из Лабораторной работы №1 добавьте страницу, содержащую надпись, ProgressBar и две кнопки:



По клику на кнопку Start **асинхронно** (**await, async Task**) запускается вычисление интеграла функции $y=\sin(x)$ на участке от 0 до 1 (использовать метод прямоугольников). Для итерации использовать шаг 0,00000001. Для увеличения времени выполнения вычисления на каждой итерации можно ввести задержку в виде цикла из 100000 формальных вычислений (например, умножения двух чисел). Подобрать значения так, чтобы вычисление занимало около секунд.

ProgressBar должен отображать прогресс вычисления. Также прогресс должен выводиться в процентном отношении (см. рисунок).

Кнопка Cancel предназначена для отмены вычисления (использовать CancellationToken).

Надпись в верхней части экрана (на рисунке это «Welcome to .NET MAUI!») должна меняться на следующие сообщения:

- «Вычисление», когда идет вычисление интеграла (по клику кнопки Start);
- результат вычисления интеграла по завершении вычисления;
- «Задание отменено», если была нажата кнопка Cancel **во время**

вычисления

2. Вопросы для самопроверки

- 1) Какие элементы управления для установки (выбора) значений вы знаете?
- 2) Какие элементы управления, кроме ProgressBar, можно использовать для индикации того, что выполняются какие-то вычисления?
- 3) Какое событие вы используете для обработки данных при выборе значения в элементе Picker?
- 4) Как можно управлять элементами XAML из вторичного потока?
- 5) Что такое ресурсы? Какие наиболее часто используемые ресурсы приложений MAUI вы знаете?
- 6) Как подключить ресурс на страницу,
- 7) Чем отличается статический ресурс от динамического?
- 8) Как описать словарь ресурсов?
- 9) Что такое BindingContext?
- 10) Как привязать свойство одного объекта View к свойству другого объекта View на той же странице?
- 11) Как привязать свойство объекта View к свойству в code-behind?
- 12) Как проверить, выполняется ли код в основном потоке?

Лабораторная работа №3

Работа с базой данных

Цель работы: знакомство с ORM SQLite.Net.

Задача работы: научиться сохранять данные в локальной БД, выводить на экран списки данных.

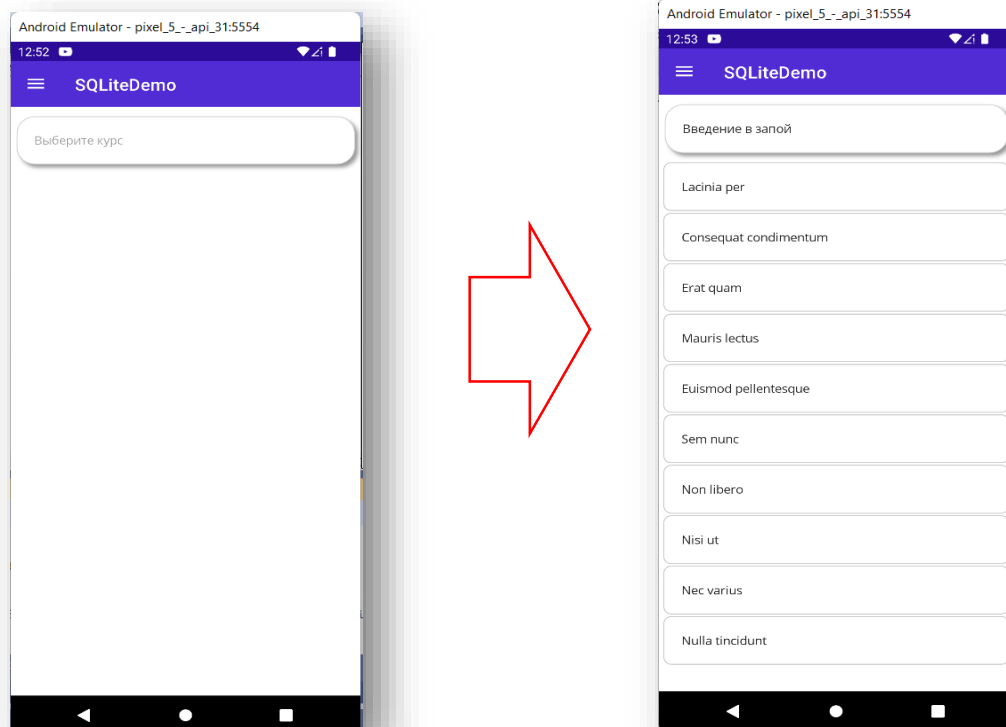
Время выполнения работы: 4 часа (2 занятия)

Результат выполнения работы: программа, обеспечивающая функционал согласно заданию.

1. Задание

Использовать проект из лабораторных работ 1-2.

Разработать страницу, на которой при выборе группы выводится список названий объектов в этой группе, например:



Для вывода списка используйте элемент **CollectionView**

3. Рекомендации к заданию:

Загрузку списка групп выполнить по событию страницы **Loaded**
 Загрузку списка объектов группы выполнять по событию **SelectedIndexChanged**

- 1) Загрузите в проект NuGet пакет **sqlite-net-pcl**.
- 2) Добавьте в проект папку Entities. В созданной папке опишите классы, отображаемые на таблицы БД согласно индивидуальному заданию. Отношения между таблицами БД один-ко-многим (одна группа объектов содержит много объектов).

Например: курсы – слушатели. Один курс проходят много слушателей.
 Используйте атрибуты для указания ключевых полей, например:
 Класс «Course»

```
[Table("Courses")]
public class Course
{
    [PrimaryKey, AutoIncrement, Indexed]
    public int Id { get; set; }
    public string Name { get; set; }
    public DateTime StartDate { get; set; }
    public int Duration { get; set; }
}
```

Класс «Trainee»

```
[Table("Trainees")]
public class Trainee
{
    [PrimaryKey, AutoIncrement, Indexed]
    [Column("Id")]
    public int TraineeId { get; set; }
    public string Name { get; set; }
    public string Photo { get; set; }
    [Indexed]
    public int CourseId { get; set; }
}
```



Обратите внимание на свойство CourseId класса Trainee. Это свойство указывает, на ключ в таблице «Courses», а, следовательно, говорит, к какому курсу относится данный слушатель (Foreign Key в реляционных базах данных).

3) Добавьте в проект папку Services.

В папке Services создайте интерфейс **IDbService**, в котором опишите методы:

- Получение списка групп объектов
- Получение списка объектов, принадлежащих группе.
- Создание базы данных и заполнение ее начальными значениями

Например:

```
public interface IDbService
{
    IEnumerable<Course> GetAllCourses();
    IEnumerable<Trainee> GetCourseMembers(int id);
    void Init()
}
```

В приведенном примере параметр `int id` – это ключевое поле в таблице Courses.

Примечание: в вашем проекте названия методов и классы сущностей должны соответствовать индивидуальному заданию.

4) В папке Services опишите класс **SQLiteService**, который наследуется от интерфейса **IDbService**, и реализует методы интерфейса с помощью библиотеки **SQLite.Net** (см. п.1 задания). В методе `Init` создайте базу данных и заполните ее исходными данными – 2-4 группы объектов и по 5-10 объектов для каждой группы.

5) В классе MauiProgram зарегистрируйте созданный сервис:

```
services.AddTransient<IDbService, SQLiteService>();
```

6) Добавьте в проект страницу, которая будет отображать выпадающий список групп, зарегистрированных в БД (элемент `Picker`).

7) Внедрите **IDbService** через конструктор страницы

8) Зарегистрируйте страницу в качестве сервиса в классе MauiProgram

4. Индивидуальные задания

1. Больничная палата – пациенты
2. Тематический зал музея – экспонаты
3. Виды процедур SPA салона – процедуры
4. Виды транспорта транспортной компании – автомобили
5. Бригады –выполняемые работы
6. Коктейли – ингредиенты
7. Авторы – книги
8. Бренды косметики – изделия
9. Спортивные команды – участники
10. Блогер – Сториз
11. Исполнители – песни
12. Наборы суши – суши в наборе
13. Супергерои – способности
14. Туристические маршруты – достопримечательности
15. Категория гостиничного номера – предоставляемый сервис
16. Сотрудники – должностные обязанности
17. Номинации конкурса – участники
18. Проект – задачи, выполняемые в рамках проекта
19. Позиции меню пиццерии – заказы клиентов
20. Марка автомобиля – объявления о продаже

5. Вопросы для самопроверки

- 1) Что такое внедрение зависимостей?
- 2) Какие есть способы внедрения?
- 3) Что такое контейнер IoC?
- 4) Как зарегистрировать сервис в проекте .Net MAUI?
- 5) Чем отличается сервис Transient от Singleton?
- 6) Как описать отношение один-ко-многим в классах C# ?
- 7) Как создать базу данных с помощью пакета sqlite-net-pcl?
- 8) Как записать данные в таблицу БД с помощью пакета sqlite-net-pcl?
- 9) Как получить отфильтрованные данные из таблицы БД с помощью пакета sqlite-net-pcl?
- 10) Что такое ORM?
- 11) Что такое первичный ключ (primary key)?
- 12) Какие виды отношений между таблицами реляционной базы данных вы знаете?
- 13) Что означает атрибут [PrimaryKey, AutoIncrement] в описании сущности?
- 14) Что такое поставщик базы данных (Data Base Provider)?

Лабораторная работа №4

Работа с REST API сервисом

Цель работы: Изучение возможности использования интернет-соединения в мобильных приложениях.

Задача работы: научиться получать, обрабатывать и отображать данные с веб-сервисов.

Время выполнения работы: 4 часа (2 занятия)

Результат выполнения работы: программа, обеспечивающая функционал согласно заданию.

1. Задание

Использовать проект из лабораторных работ 1-3.

Добавить в проект страницу – «Конвертер валют».

Конвертер должен:

1. Показывать официальный курс белорусского рубля, установленный Национальным банком Республики Беларусь **на выбранную дату** для следующих валют:

- Российский рубль
- Евро
- Доллар США
- Швейцарский франк
- Китайский юань
- Фунт стерлингов

2. Выбирать валюту для конвертации

3. Позволять выполнять пересчет суммы в белорусских рублях в выбранную иностранную валюту (см. п. 2.) и обратно

6. Рекомендации к заданию:

- 1) Используйте API, описанный здесь: <https://www.nbrb.by/apihelp/exrates>
- 2) Можно использовать класс **Rate**, приведенный здесь: <https://www.nbrb.by/apihelp/exrates/ratescs.zip>

3) В папке Services создайте интерфейс `IRateService`, предоставляющий метод получения курса валют, указанных в п.1 задания, на требуемую дату:

```
public interface IRateService
{
    IEnumerable<Rate> GetRates(DateTime date);
}
```

4) Добавьте класс `RateService`, реализующий интерфейс `IRateService`.

5) В конструктор класса `RateService` внедрите объект `HttpClient` для отправки запросов веб-сервису

6) В классе `MauiProgram` зарегистрируйте созданный сервис:

7) В классе `MauiProgram` зарегистрируйте сервис – `HttpClientFactory`:

```
services.AddHttpClient<IRateService>(opt =>
    opt.BaseAddress = "https://www.nbrb.by/api/exrates/rates")
```

8) В конструкторе страницы «Конвертер валют» внедрите сервис `IRateService`.

9) В классе `MauiProgram` зарегистрируйте сервис – страницу «Конвертер валют»:

7. Вопросы для самопроверки

- 1) Для чего используется класс `Connectivity`?
- 2) Какие базовые функции предоставляет сервис REST?
- 3) Есть ресурс API: <https://xxxx/books/2>, который представляет доступ к книге, у которой `id=2`. От чего будет зависеть результат запроса к этому ресурсу (просто чтение данных, модификация данных или удаление ресурса)?
- 4) Как получить объект класса `HttpClient`?