Серия Обучение для профессионалов

Основы

Том 1

Адаптированная объектная модель компонентов Экосистемы

Владимир Башев

Оглавление

Введение

Добро пожаловать! Мы рады представить вам официальный учебный курс Обучение для профессионалов по основам Адаптированной Объектной Модели Компонентов (АОМК) – технологическому стандарту компании ПИРФ для создания программного обеспечения. Эта книга является исчерпывающим руководством по разработке программных компонентов. Она рассматривает все вопросы, связанные с TO DO.

Методология

Эта книга учит практическому применению технологии АОМК. Она делится на проекты, которые последовательно знакомят вас с основами технологии АОМК. К концу книги вы сможете разработать свои собственные компоненты используя технологию АОМК.

Каждый урок в книге представляет самостоятельную ценность, поэтому можно перейти к любому уроку в любое время. Однако каждый урок строится на основе материала, изученного в предыдущем уроке и новичкам лучше всего прочесть книгу с начала до конца.

1

Файлы урока	Eco.Pro.Training\001.InsideACOM\Lessons\Lesson01
Дополнительные материалы	Eco.Pro.Training\001.InsideACOM\Books
Время	Урок рассчитан на 130 минут
Задачи	Общее представление АОМК технологи
	Практическое использование на примере Калькулятора

Часть 1

Простейший путь понять, в чем новизна и революционность АСОМ-компонентов и библиотеки Eco.Framework – это поработать с ними. Прочитайте эту главу, и вы получите первое представление о богатейших возможностях АСОМ-компонентов и библиотеки Eco.Framework.

Урок 1.

Перед тем как приступить к уроку, небольшое отступление. Так как библиотека Eco.Framework проектируется и разрабатывается исключительно для операционной системы Эко ОС, то отдельные интерфейсы компонентов, связанные с системной и аппаратной частью, могут быть не доступны для операционных систем таких как Windows, Mac OS X, RedHat Linux, Solaris, Android, iOS.

ВАЖНО: Библиотека Eco.Framework не является кроссплатформенной, как это может показаться на первый взгляд. Об ограничения библиотеки Eco.Framework под ту или иную операционную систему, можно узнать, обратившись к справочному руководству библиотеки Eco.Framework.

Первый урок и все последующие уроки рассчитаны на работу по ОС Windows с использованием среды разработки VisualStudio 2010. Тем не менее основы использования компонентов, описанные в уроках справедливы для использования в других ОС и соответствующих сред разработки.

Технология АОМК

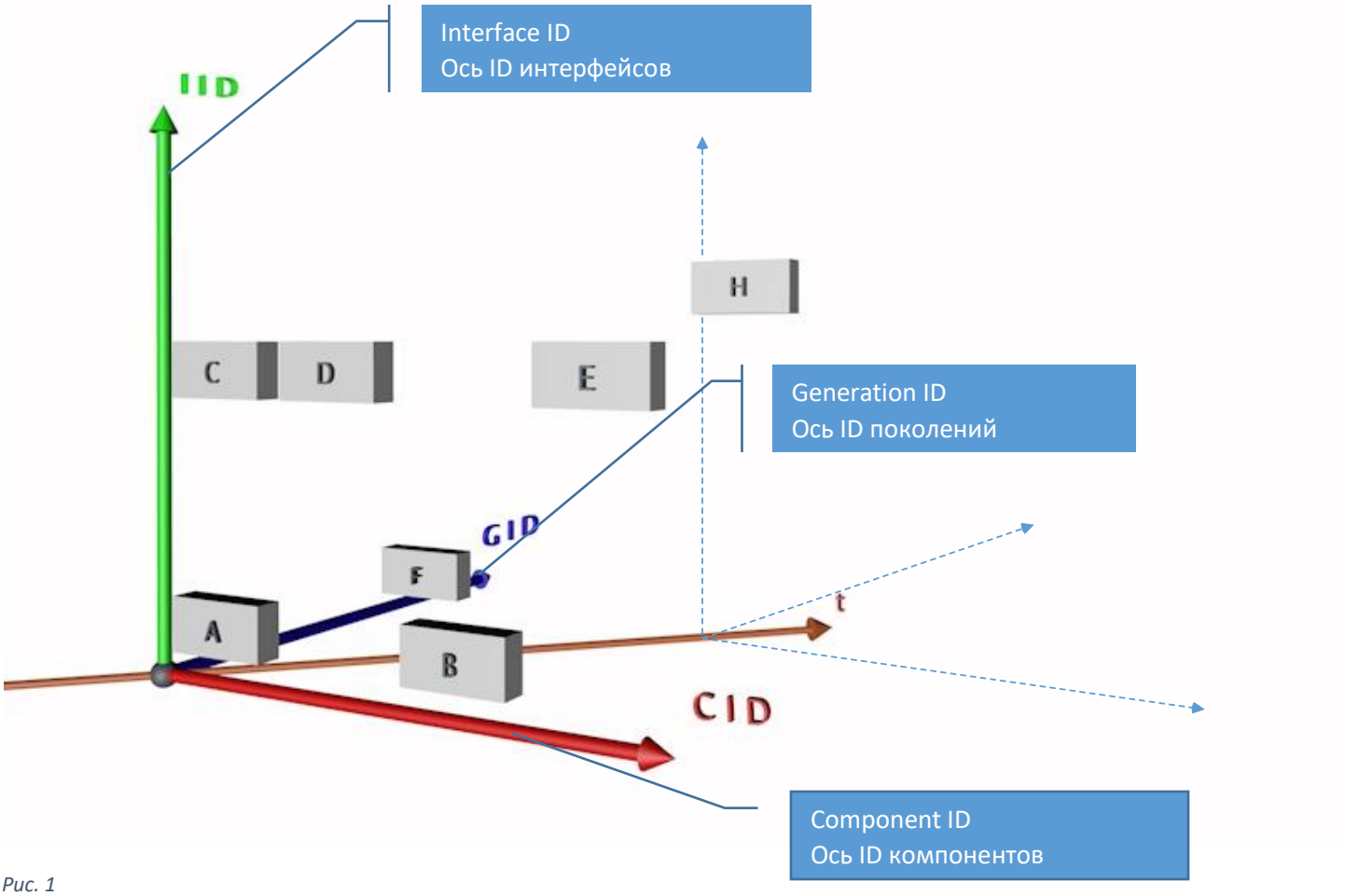


Рис. 1

Технологию Адаптированной Объектной Модели Компонентов (сокр. АОМК) можно представить в виде пространства и времени, показанного на Рис. 1, где время отвечает за изменения, связанные с компонентом и его новой сборкой, в результате исправления допущенных ошибок в разработке или недоработках заложенного функционала (несоответствию заявленным в техническом паспорте компонента характеристикам).

Пространство, с левосторонней системой координат, имеет следующие оси:

X-ось (Component ID, сокр. CID) – ось идентификаторов компонентов, отвечает за создание различных по характеристикам компонентам, от разных разработчиков/производителей для соответствующего интерфейса и поколения.

Y-ось (Interface ID, сокр. IID) – ось интерфейсов, отвечает за развитие компонентов, добавление нового функционала и возможностей.

Z-ось (Generation ID, сокр. GID) – ось поколений, отвечает за научно-технический прогресс, на этапах которого были разработаны те или иные компоненты.

На Рис. 1 компоненты отображены виде параллелепипедов с буквами, где:

A – Компонент с интерфейсом X относящийся к первому поколению ОС разработанный компанией N

B – Компонент с интерфейсом X поддерживающим механизм агрегирования, относящийся к первому поколению ОС разработанный компанией M

C – Компонент с интерфейсом X и новым интерфейсом Y относящийся к первому поколению ОС разработанный компанией N

D – Компонент с новым интерфейсом Y и включающий в себя компонент A с интерфейсом X, относящийся к первому поколению ОС разработанный компанией N

E – Компонент с новым интерфейсом Y и агрегирующий компонент B с интерфейсом X, относящийся к первому поколению ОС разработанный компанией M

F – Компонент с интерфейсом X относящийся к второму поколению ОС разработанный компанией N

H – Компонент с интерфейсом X и новым интерфейсом Y относящийся к второму поколению ОС разработанный компанией M

Необходимость в создании разных компонентов с одним и тем же интерфейсом и для одного и того же поколение, диктуется следующим:

- когда архитектурно на аппаратном уровне используется оборудование разных производителей. К примеру, видеокарта от NVidia или встроенная от Intel.

- когда компоненты от разных производителей имеют разные технические решения. К примеру графическая библиотека API OpenGL версии 1.1 используемая аппаратное ускорение или программное решение.

- когда один компонент поддерживает механизм агрегирования, а другой компонент нет. Данный пример будет рассмотрен в этой книге.

Необходимость в создании компонентов с новым интерфейсом, диктуется следующим:

- когда необходимо добавить новые возможности к существующему функционалу. К примеру, компонент калькулятор умел только складывать и вычитать, добавился новый интерфейс, позволяющий выполнять операции деления и умножения.

- когда необходимо унифицировать взаимодействие с различными компонентами, имеющими разные интерфейсы со схожим функционалом. К примеру, слой аппаратных абстракций (HAL- анг.) реализация компонента, обеспечивающего взаимодействие между программным обеспечением и компонентами аппаратного обеспечения.

Необходимость в создании компонентов под разные поколения, диктуется следующим:

- когда выпускается новое поколение ЦП, МП, МК, SoC и т.п. обратно совместимое с предыдущим поколением, новые решения разрабатываются с учетом новых возможностей, и чтобы обеспечить обратную совместимость компонентов на программном уровне используется системный интерфейс с идентификатором соответствующего поколения. К примеру, новые 32 разрядные процессоры Intel обратно совместимы с 16 разрядными процессорами, включают в себя виртуальный x86 режим.

Демонстрация АОМК технологии

В данном уроке мы исследуем АОМК технологию на примере компонентов калькулятора.

Пример 1

Первый пример демонстрирующий АОМК технологию, мы начнем с постановки задачи, но при этом будем считать, что необходимые компоненты уже разработаны и мы их будем использовать в своих решениях.

Задача: Необходимо разработать клиента, который мог бы работать с компонентами разных производителей, поддерживающих интерфейс IEcoCalculatorX - выполняющий операции сложения и вычитания.

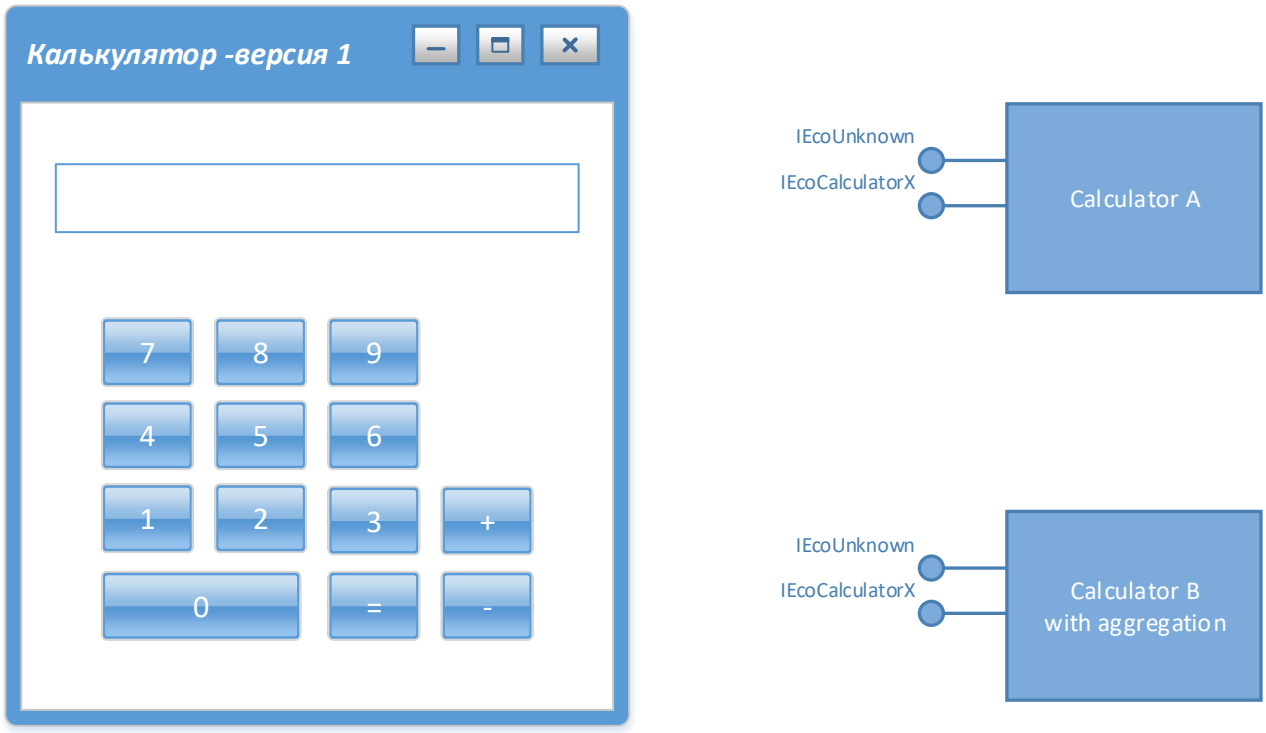


Рис. 2

При решении данной задачи, мы использовали два компонента «Eco.CalculatorA» и «Eco.CalculatorB», которые поддерживаю интерфейс IEcoCalculatorX.

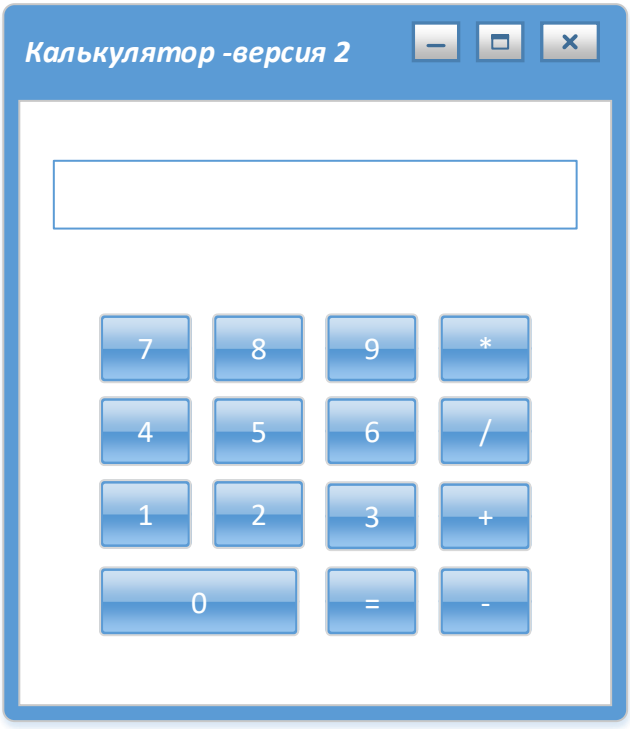
Примеры проектов находятся по пути «Eco.Pro.Training\001.InsideACOM\Lessons\Lesson01». Откройте папку «A», в которой находятся приложение-клиент DemoCalculator1.exe и компонент-сервер 4828F6552E4540E78121EBD220DC360E.dll. Запустите клиента и произведите операции сложения и вычитания. Теперь перейдите в папку «B», в которой находится тот же клиент DemoCalculator1.exe, но совсем другой компонент AE202E543CE54550899603BD70C62565.dll. Запустите клиента DemoCalculator1.exe из папки «B» и произведите операции сложения и вычитания. Как видно, один и тот же клиент может работать с разными компонентами, имеющими разные идентификаторы компонента CID, но реализующие один и тот же интерфейс IEcoCalculatorX. Данные примеры, отражают развитие по X-оси (CID – Component ID) на Рис. 1, где показаны компоненты «A» и «B» с разным идентификатором компонента. Механизм выбора компонента с разным CID, более подробно будет рассмотрен в следующих уроках.

Развитие по X-оси (CID – Component ID) Рис. 1, можно сравнить с лампочками, применяемыми в быту, где лампочки разной формы, разной мощности являются компонентами, цоколь – является интерфейсом, а люстра клиентом. Разные производители производят разные лампочки по форме и мощности со стандартизированным размером цоколя, к примеру, E27, которые используются в люстрах с патроном E27.

Пример 2

Второй пример демонстрирует АОМК технологию, отражая развитие по Y-оси (IID – Interface ID) на Рис. 1, где показаны компоненты «A» и «C» с одним и тем же идентификатором компонента. Начнем с постановки задачи.

Задача: Необходимо разработать нового клиента, который мог бы работать с компонентами разных производителей, как со старыми поддерживающими интерфейс IEcoCalculatorX - выполняющий операции сложения и вычитания, так и с новыми поддерживающими интерфейс IEcoCalculatorY - выполняющий операции деления и умножения.



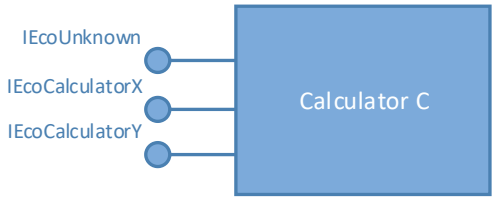


Рис. 3

При решении данной задачи, мы использовали три компонента «Eco.CalculatorA», «Eco.CalculatorB» которые поддерживаю интерфейс IEcoCalculatorX и «Eco.CalculatorC», который поддерживает интерфейсы IEcoCalculatorX и IEcoCalculatorY.

Примеры проектов находятся по пути «Eco.Pro.Training\001.InsideACOM\Lessons\Lesson01». Откройте папку «C», в которой находятся приложение-клиент DemoCalculator2.exe и компонент-сервер 4828F6552E4540E78121EBD220DC360E.dll. Запустите клиента и произведите операции сложения, вычитания, деления и умножения. Что мы наблюдаем!?. Мы наблюдаем нового клиента версии 2, который работает с новым компонентом «Eco.CalculatorC», но у которого идентификатор компонента совпадает с

идентификатором компонента «Eco.CalculatorA», т.е. условно компонент «Eco.CalculatorC» является второй версией компонента «Eco.CalculatorA». И в компоненте «Eco.CalculatorC» в отличии от компонента «Eco.CalculatorA» реализован интерфейс IEcoCalculatorY - выполняющий операции деления и умножения.

Как вы можете наблюдать, в этой же папке «C», находиться приложение-клиент DemoCalculator1.exe. Как оно поведет себя с новым компонентом!?, ведь старое приложение ничего не знает о новом… Давайте запустим и посмотрим. Как видно приложение успешно выполняет операции сложения и вычитания. В чем магия и как это работает!? Работает это следующим образом, новый компонент «Eco.CalculatorC» использует идентификатор компонента предыдущей версии «Eco.CalculatorA», поэтому старое приложение воспринимает его как компонент первой версии. Теперь исследуем обратную ситуацию, как поведет себя новый клиент, со старым компонентом «Eco.CalculatorA», для этого перейдите в папку «A», в которой вместе с клиентом DemoCalculator1.exe находится клиент DemoCalculator2.exe и запустите его. Результат, новый клиент успешно работает и со старым компонентом «Eco.CalculatorA» выполняя операции сложения и вычитания, но не может выполнять операции деления и умножения, так как старый компонент не поддерживает интерфейс IEcoCalculatorY. На Рис. 4 ниже отображены варианты использования. Не забудьте проверить работу клиента DemoCalculator2.exe в папке «B», с компонентом «Eco.CalculatorB».

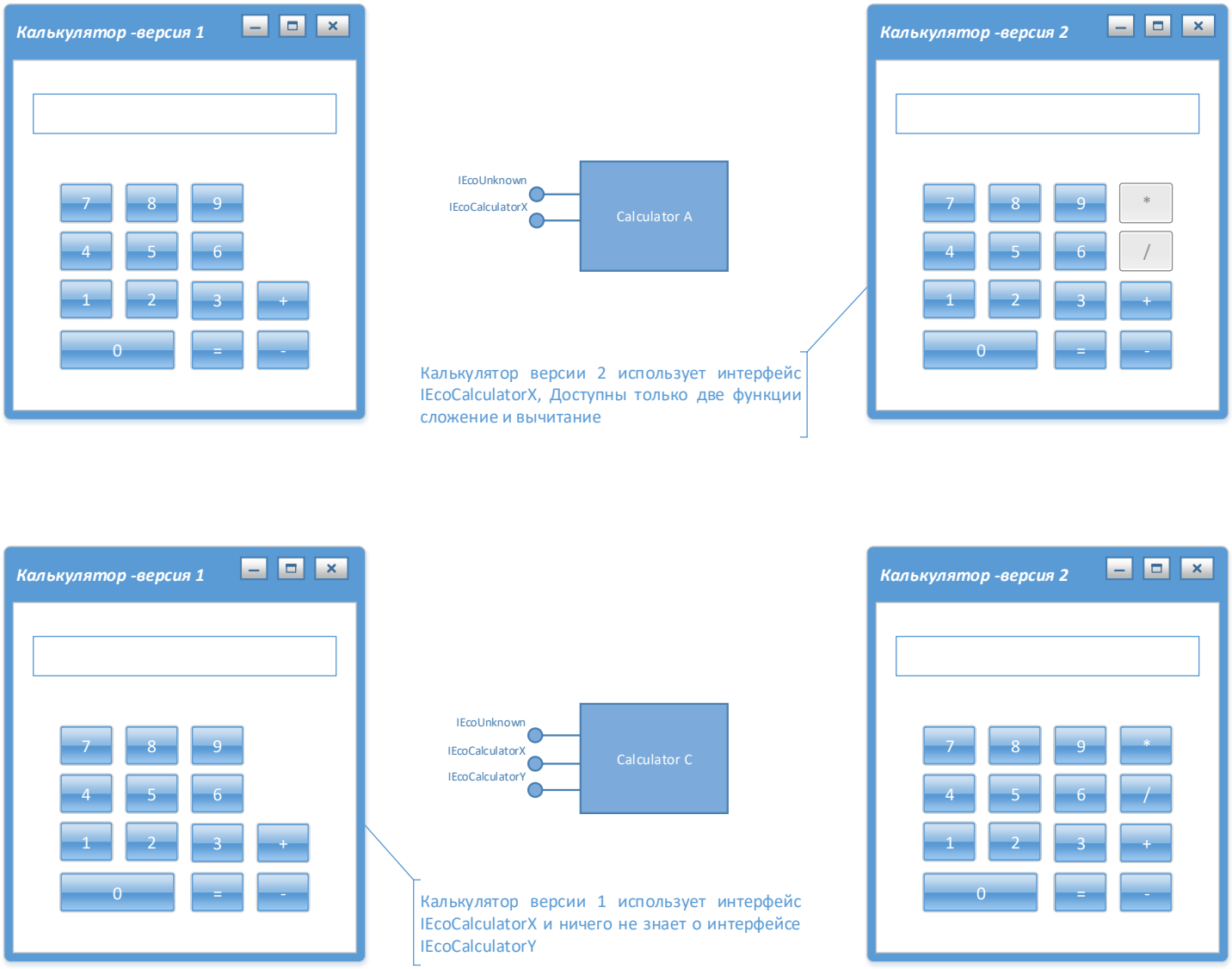


Рис. 4

Строгое движение по Y-оси (IID – Interface ID) имеет существенный недостаток, при реализации нового интерфейса IEcoCalculatorY, нам пришлось заново реализовывать интерфейс IEcoCalculatorX. Конечно в интерфейсе IEcoCalculatorX всего два метода и повторно их реализовать в новом компоненте нам не составило труда. Но что делать, если компонент содержит множество интерфейсов с достаточно большим набором методов и более сложной реализацией!? Следующий пример продемонстрирует, как решить данную проблему.

Пример 3

Этот пример демонстрирует АОМК технологию, отражая развитие как по Y-оси (IID – Interface ID), так и по X-оси (CID – Component ID) на Рис. 1, где показаны компоненты «A» и «D». Данный пример демонстрирует использование программного приема называемым «включение». Нет необходимости заново переписывать реализацию компонента «A», внешний компонент «D» повторно использует реализацию компонента «A».

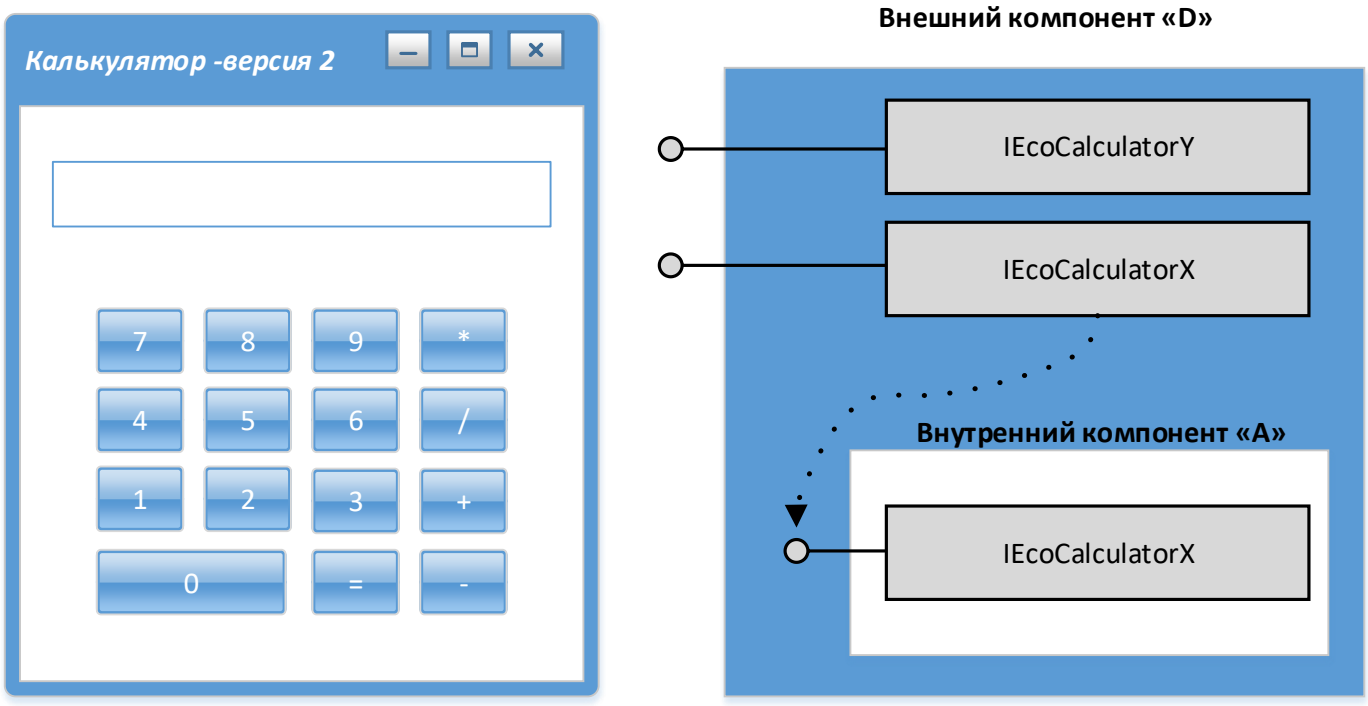


Рис. 5

Примеры проектов находятся по пути «Eco.Pro.Training\001.InsideACOM\Lessons\Lesson01». Откройте папку «D», в которой находятся приложение-клиент DemoCalculator2.exe и два компонента 4828F6552E4540E78121EBD220DC360E.dll, 3A8E44677E82475CB4A3719ED8397E61.dll. Запустите клиента и произведите операции сложения, вычитания, деления и умножения.

Пример 4

Этот пример демонстрирует АОМК технологию, аналогично предыдущему примеру, отражая развитие как по Y-оси (IID – Interface ID), так и по X-оси (CID – Component ID) на Рис. 1, где показаны компоненты «B» и «E». Только в этом примере демонстрируется использование другого программного приема,

называемого «агрегированием». Где также нет необходимости заново переписывать реализацию компонента «B», внешний компонент «E» агрегирует интерфейс и передает указатель на него непосредственно клиенту.

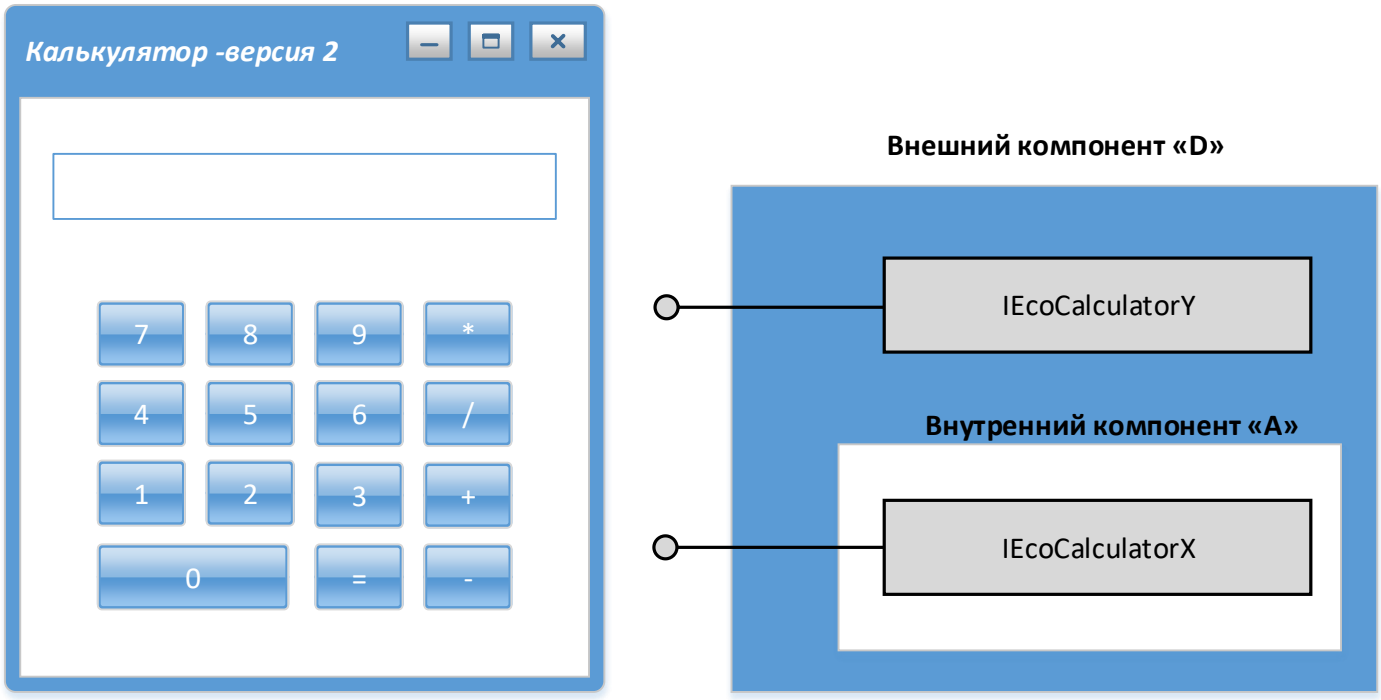


Рис. 6

Примеры проектов находятся по пути «Eco.Pro.Training\001.InsideACOM\Lessons\Lesson01». Откройте папку «E», в которой находятся приложение-клиент DemoCalculator2.exe и два компонента AE202E543CE54550899603BD70C62565.dll, 872FEF1DE3314B87AD44D1E7C232C2F0.dll. Запустите клиента и произведите операции сложения, вычитания, деления и умножения.

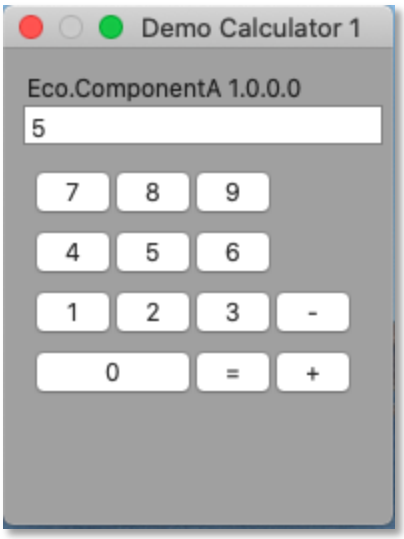
Пример 5

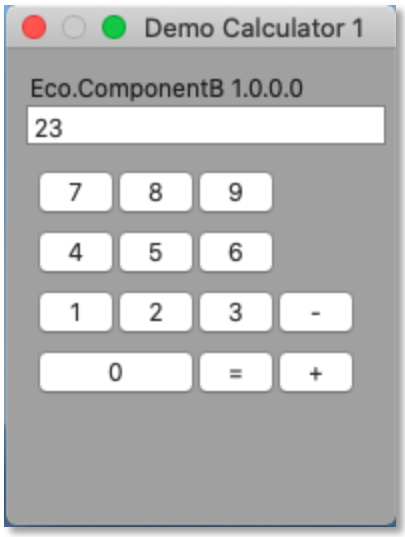
Последний пример демонстрирует AOMK технологию, отражая развитие по Z-оси (GID – Generation ID) на Рис. 1, где показаны компоненты «A» и «F».

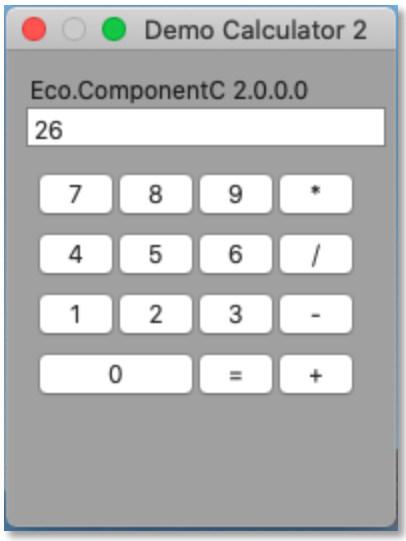
Снимки экранов

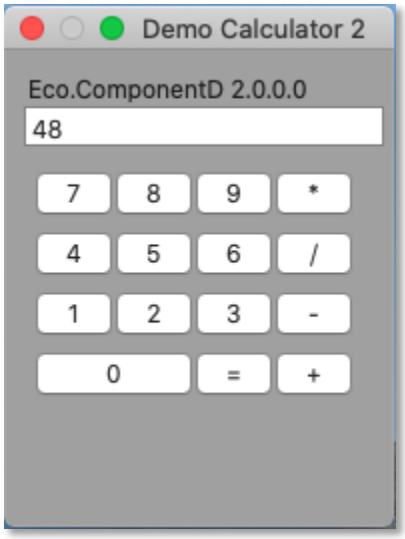
Серия снимков экранов для Windows.

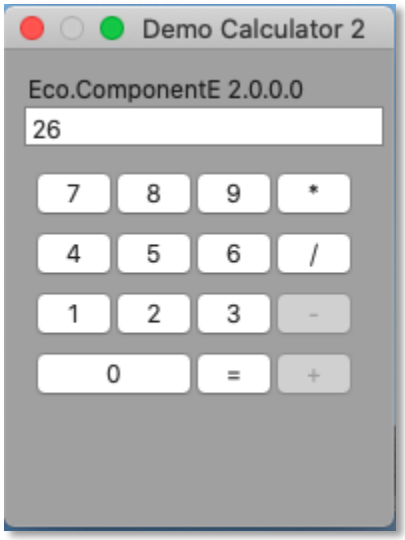
Серия снимков экранов для Mac OS X.



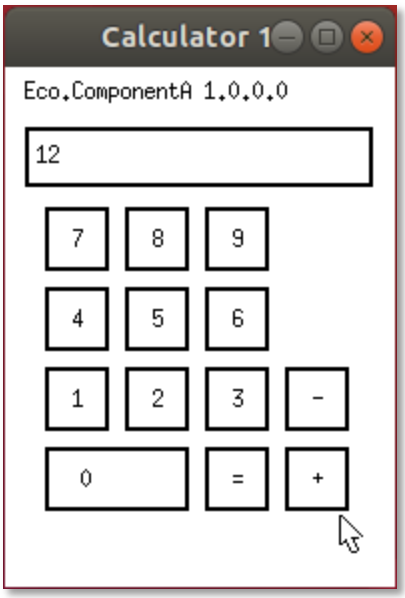


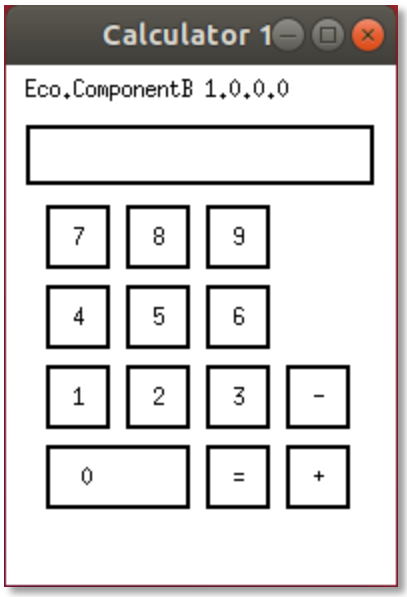


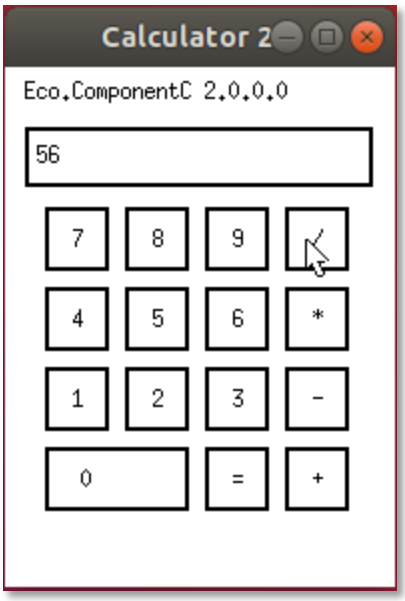


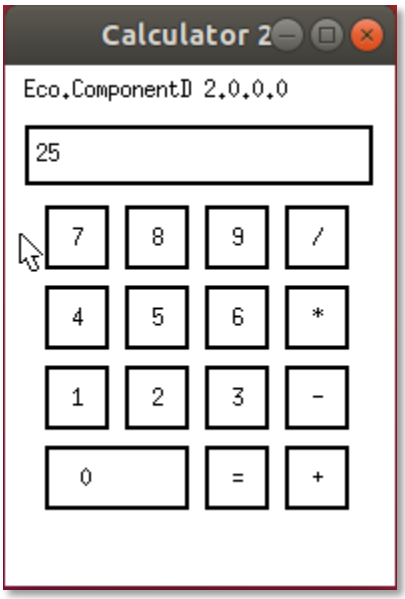


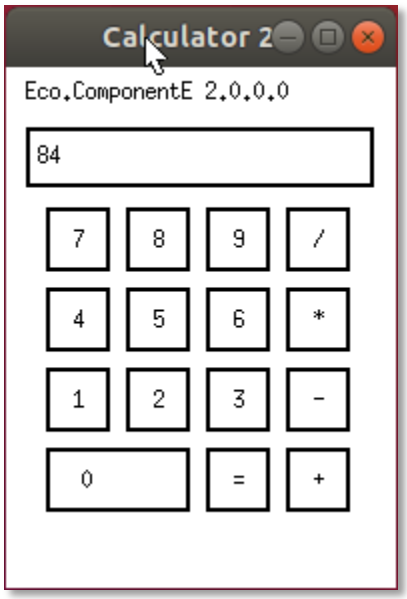
Серия снимков экранов для Linux.



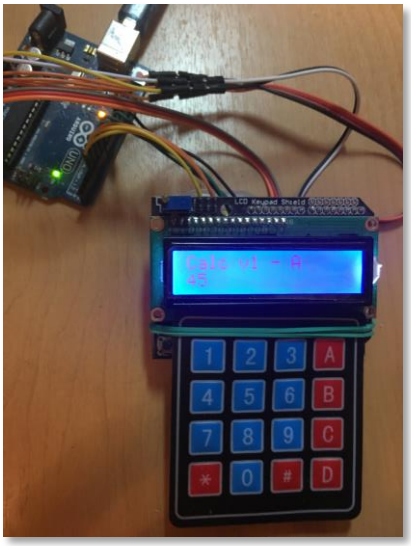


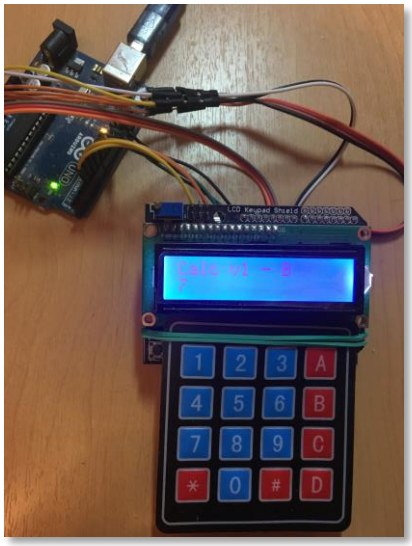


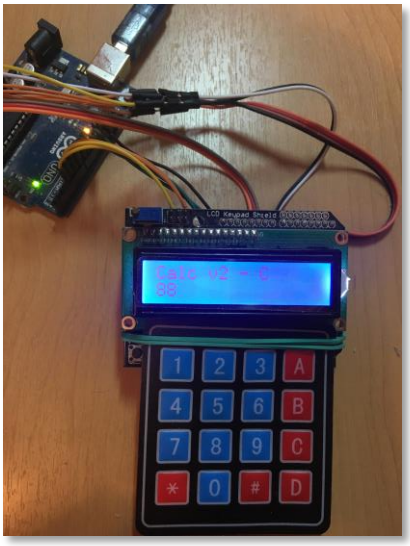


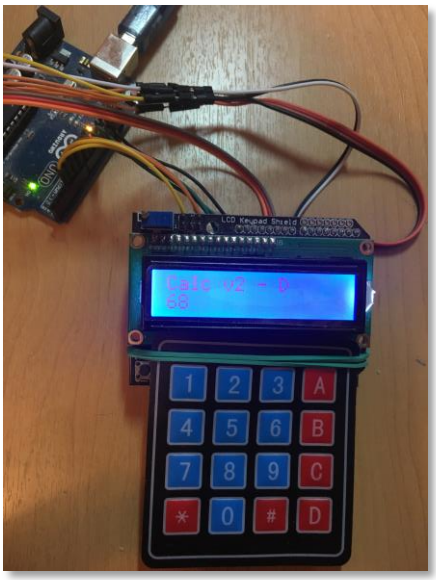


Серия снимков экранов для AVR8.

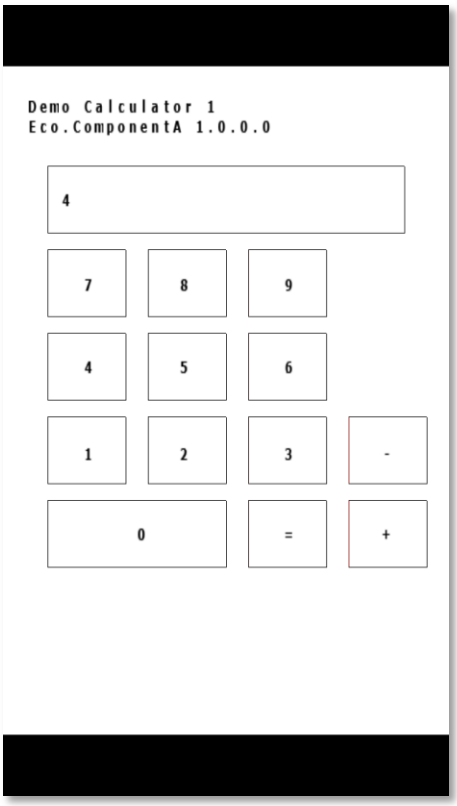


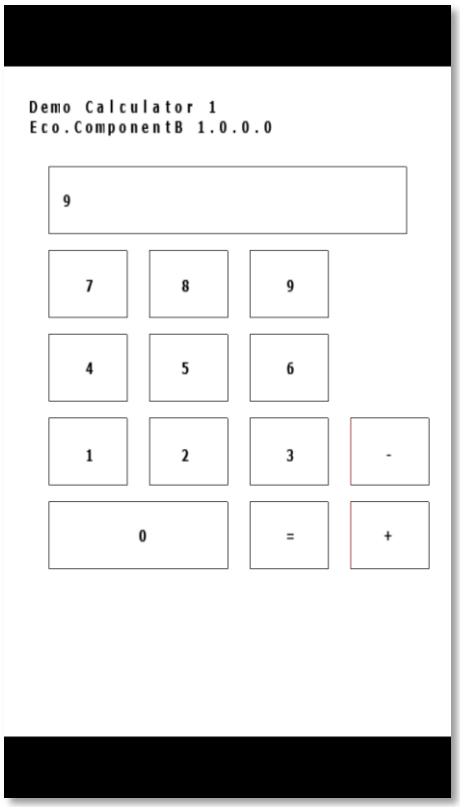


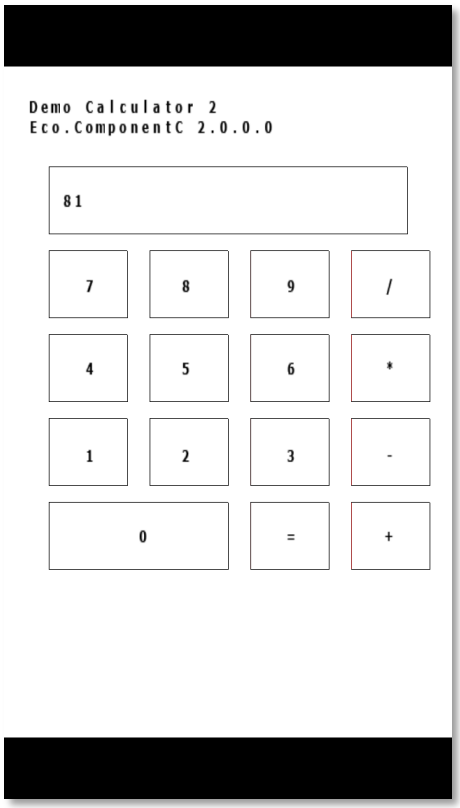


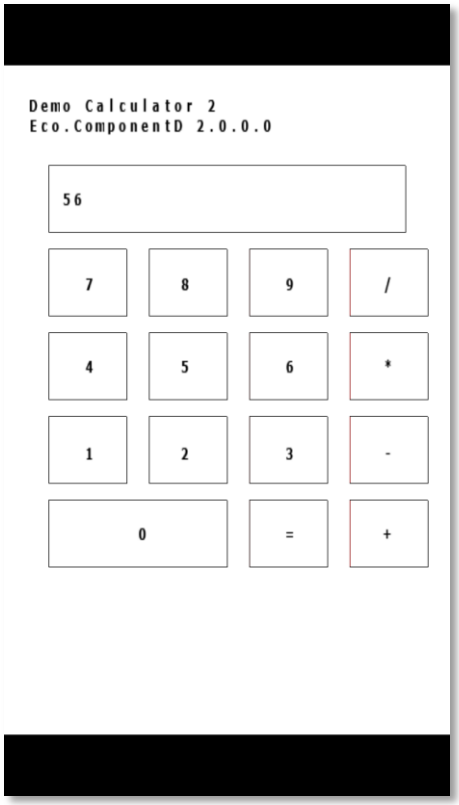


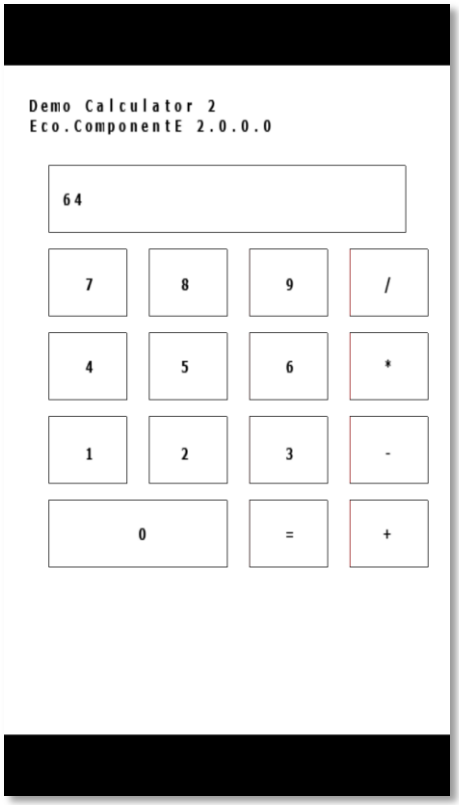
Серия снимков экранов для Android & iOS.











Заключение

Мы ознакомились с технологией АОМК на демонстрационных примерах, которые мы научимся создавать самостоятельно в последующих уроках шаг зашагом изучая технологию АОМК в тончайших деталях.

2

Файлы урока	Eco.Pro.Training\001.InsideACOM\Lessons\Lesson02
Дополнительные материалы	Eco.Pro.Training\001.InsideACOM\Books
Время	Урок рассчитан на 130 минут
Задачи	

Урок 2.

На первом уроке мы получили первое представление о возможностях АОМК технологии.

Для упражнений в данном уроке, создайте проект с названием «Lesson02». и подготовьте его по примеру первого урока. Пример проекта имеет путь «Eco.Pro.Training\001.InsideACOM\Lessons\».

Книги серии обучение для профессионалов

Серия Обучение для профессионалов

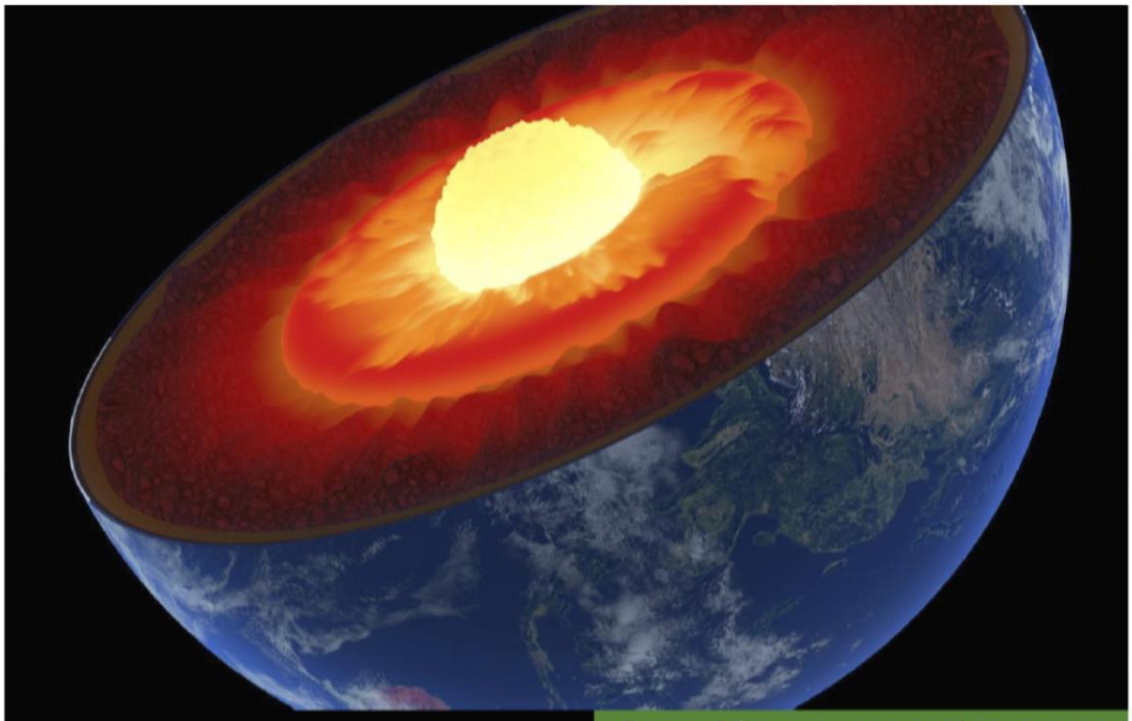
Набор инструментов

Том 3

Грамматика, анализаторы и компиляторы

Библиотеки Eco.Framework

Владимир Башев



Серия Обучение для профессионалов

Операционная система

Том 4

Ядро, загрузчик и компоненты

Эко ОС

Владимир Башев

Серия Обучение для профессионалов

Форматы исполняемых файлов

Том 5

Компоненты формирования и обработки форматов исполняемых файлов

Владимир Башев