

Задания к курсу

“Распределённые системы”

3 курс, 2024-2025 уч. г. (Часть 1)

Все задания выполняются с использованием платформы WPF (Windows Presentation Foundation), либо AvaloniaUI.

1. Разработайте библиотеку, предоставляющую:

1.1. Стили для элементов управления WPF:

- Button: кнопка с текстом, с возможностью настройки извне цвета шрифта, размера шрифта, цвета фона относительно свойств объекта Button;
- Button: кнопка с картинкой, с возможностью настройки извне изображения и его масштабов (xScale, yScale);
- ScrollViewer: панель с прокрутками (вертикальной / горизонтальной), с возможностью настройки извне цвета ползунка прокрутки;

1.2. Элементы управления WPF:

- Spinner (загрузка - круги различных радиусов,двигающиеся вокруг центра), с возможностью настройки извне: цвета кругов, количества кругов, размера кругов, направления движения кругов (по часовой стрелке / против часовой стрелки), скорости вращения кругов;
- DialogHost (рабочая область - белый прямоугольник со скруглёнными углами поверх фона - чёрного прямоугольника), с возможностью настройки извне: радиуса закругления углов рабочей области (каждого по отдельности, по умолчанию 0.0), прозрачности фона (по умолчанию 0.4), команды нажатия на фон диалога. В рабочей области диалога необходимо разместить ContentPresenter;
- SpinnerDialog (диалог загрузки) - DialogHost, в содержимом которого находятся друг под другом Spinner и TextBlock (с текстом по умолчанию “Please, wait...”), с возможностью настройки извне: текста, размера шрифта текста, а также необходимых свойств инкрустированного в диалог элемента управления Spinner;
- MessageDialog (диалог с сообщением и кнопками) - DialogHost, в содержимом которого находится завёрнутый в стилизованный ScrollViewer (с ползунками, отображающимися при необходимости) TextBlock (с автоматическим переносом текста) и кнопки (единовременно: либо ОК, либо Yes / No, либо ОК / Cancel), с возможностью настройки извне: текста, размера шрифта текста, типа диалога (влияет на отображаемые на диалоге кнопки, реализовать через enum), команд для кнопок, а также необходимых свойств инкрустированного в диалог элемента управления типа ScrollViewer;
- NumericKeyboard (экранная клавиатура) - панель с кнопками для цифр 0-9 и кнопкой C (стереть), с возможностью настройки извне: стиля для кнопок с цифрами, стиля для кнопки стирания, команды для обработки нажатия на кнопку цифры, команды для обработки нажатия на кнопку стирания;
- LetterKeyboard (экранная клавиатура) - панель с кнопками для букв а-я и a-z, кнопкой C (стереть), кнопкой переключения языка (единовременно отображаются кнопки только для одного языка; по умолчанию - английский язык), кнопкой CapsLock (для изменения регистра вводимых через клавиатуру символов, по умолчанию - в нижнем регистре; изменения свойства должны затрагивать также и user interface), кнопкой стирания, кнопкой Enter (подтверждения ввода). Элемент управления должен позволять настраивать извне: стиль кнопок с буквами, стили зажатой кнопки CapsLock и отжатой кнопки CapsLock (выбор стиля в зависимости от данных реализовать при помощи триггера), стиля кнопки стирания, стиля кнопки Enter, и соответствующие перечисленным группам кнопок команды.

1.3. Инфраструктурные компоненты, собираемые в nuget-пакет:

- Базовые классы моделей отображения для:
 - произвольных компонентов UI (ViewModelBase)

- страниц (PageViewModelBase)
 - диалогов (DialogViewModelBase)
 - Базовые классы конвертеров значения (ValueConverterBase) и конвертеров множества значений (MultiValueConverterBase), поддерживающих механизм встраивания в разметку (MarkupExtension)
 - Реализации конвертеров: арифметического, поразрядного, эквивалентности, отношения порядка, эквивалентности null, из значения типа bool
 - Реализацию класса команды
 - Интерфейс механизма пагинации для перехода между страницами в элементе управления Frame и его реализацию
 - Интерфейс механизма работы с диалогами и его реализацию
2. Реализуйте компилятор и оконное приложение редактора для написания программ для реализованного компилятора. Скомпилированные инструкции имеют вид 32-х разрядных целых чисел, в которых содержится код операции (младшие 5 бит) и индексы регистров, с которыми эта операция должна быть выполнена (регистры 32-битовые, каждый индекс регистра == 9 бит, от младших битов к старшим в инструкции расположены биты, соответствующие первому, второму, третьему индексам:

```

|bbbbbbbbb|bbbbbbbbb|bbbbbbbbb|bbbbbb|
| 3 operand | 2 operand | 1 operand | operation |

```

). Возможные значения кодов операций (в 10-й системе счисления):

- 0 – вывести состояние всех регистров в системе счисления, которая записана в 1 операнде;
- 1 – поразрядная инверсия над содержимым 1 операнда, результат сохраняется в 3 операнд;
- 2 – дизъюнкция над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 3 – конъюнкция над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 4 – сложение по модулю 2 над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 5 – импликация над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 6 – коимпликация над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 7 – эквиваленция над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 8 – стрелка Пирса над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 9 – штрих Шеффера над 1 и 2 операндом, результат сохраняется в 3 операнд;
- 10 – сложение 1 и 2 операнда, результат сохраняется в 3 операнд;
- 11 – вычитание из 1 операнда 2 операнда, результат сохраняется в 3 операнд;
- 12 – умножение 1 и 2 операнда, результат сохраняется в 3 операнд;
- 13 – целочисленное деление 1 операнда на 2 операнд, результат сохраняется в 3 операнд;
- 14 – остаток от деления 1 операнда на 2 операнд, результат сохраняется в 3 операнд;
- 15 – обмен содержимого 1 и 2 операндов (операция swap);
- 16 – занести в 1 операнд в байт с номером, который находится во 2 операнде, байт, значение которого лежит на месте 3 операнда;
- 17 – вывести содержимое операнда 1 в системе счисления, основание которой записано во втором операнда;
- 18 - ввести в операнд 1 в системе счисления, основание которой записано во 2 операнде, значение с клавиатуры;
- 19 – найти максимальное значение 2^p , на которое делится 1 операнд, результат сохраняется в 3 операнд;
- 20 – сдвиг влево содержимого 1 операнда на количество бит, которое находится во 2-ом операнде, результат сохраняется в 3 операнд;
- 21 - сдвиг вправо содержимого 1 операнда на количество бит, которое находится во 2-ом операнде, результат сохраняется в 3 операнд;
- 22 – циклический сдвиг влево содержимого 1 операнда на количество бит, которое находится во 2-ом операнде, результат сохраняется в 3 операнд;
- 23 – циклический сдвиг вправо содержимого 1 операнда на количество бит, которое находится во 2-ом операнде, результат сохраняется в 3 операнд;

Поток команд поступает из двоичного потока (MemoryStream). Для запуска компиляции и исполнения пользователь может нажать на кнопку с картинкой Play (зелёный треугольник) (для компиляции и исполнения инструкций, указанных в редакторе), либо нажать на кнопку с картинкой папки для создания диалога открытия файла и выбора через него файла с инструкциями (для компиляции и исполнения инструкций, указанных в файле). Во время загрузки файла, процесса компиляции исходного кода и исполнения скомпилированного кода, поверх основного окна должен появиться диалог загрузки с меняющимся текстом: во время поиска файла - “Searching for file”, во время компиляции кода - “Compiling”, во время исполнения скомпилированного кода - “Executing”. При ошибке компиляции (проверку корректности инструкций реализуйте при помощи валидатора из задания 1) необходимо отобразить диалог с сообщением о количестве ошибок и информацией о каждой ошибке и кнопкой ОК (диалог может быть закрыт как по кнопке ОК, так и по нажатию на затемнённый фон). В случае успешной компиляции, в модальном окне, имитирующем консоль, должно быть произведено исполнение извлечённых из двоичного потока инструкций. При возникновении ошибки времени исполнения (деление на 0, невалидное основание системы счисления, некорректный пользовательский ввод (обработка ввода должна производиться посредством валидатора из задания 1)), исполнение инструкций должно быть прервано и должна быть сгенерирована исключительная ситуация собственного типа, содержащая порядковый номер (начиная с 1) инструкции и дополнительную информацию о произошедшей ошибке; исключительная ситуация должна быть перехвачена и в диалоге MessageBox должна быть выведена информация о произошедшей ошибке (диалог может быть закрыт как по кнопке ОК, так и по нажатию на затемнённый фон). После исполнения инструкций окно консоли должно продолжать отображаться до момента закрытия его пользователем; попытки закрытия окна консоли во время исполнения кода должны быть перехвачены и проигнорированы. Для ввода информации в консоль необходимо использовать стилизованные экранные клавиатуры, реализованные в задании 3 (ввод в консоль с физической или стандартной экранной клавиатуры не допускается).

Для написания инструкций реализуйте оконное приложение редактора. Редактор содержит две области, в которых содержатся инструкции в текстовом и в графическом формате. Графическая область представляет собой последовательность визуализированных инструкций и операндов: операции выбираются посредством элемента управления ComboBox, значения индексов операндов вводятся при помощи TextBox, в графической области имеются возможности добавления новых инструкций в произвольное место и удаления инструкций. Текстовый формат содержит строковое представление команд в формате

<команда>,<1 oprnd indx>,<2 oprnd indx>,<3 oprnd indx>

, по одной команде на каждой отдельной строке (пробельные символы между лексемами игнорируются). При внесении любых изменений в текстовый формат инструкций, эти изменения должны быть применены также к графическому формату и наоборот.