

Организация лабораторных занятий

Требуется разработать 11 приложений, разбитых на 8 заданий:

- 1) CONSOLE, консольное приложение (6 баллов);
- 2) DIALOGS, вариант для Windows Forms (4 балла) и вариант для WPF (5 баллов);
- 3) SYNC, вариант для Windows Forms (4 балла) и вариант для WPF (5 баллов);
- 4) DRAGDROP, вариант для Windows Forms (4 балла) и вариант для WPF (5 баллов);
- 5) TIMER, вариант для WPF (5 баллов);
- 6) REGISTRY, вариант для Windows Forms (4 балла);
- 7) GRAPHICS, вариант для Windows Forms (4 балла);
- 8) MDIFORMS, вариант для Windows Forms (4 балла).

Сроки сдачи заданий:

CONSOLE, DIALOGS (всего 15 баллов) — до 16.04;

SYNC, DRAGDROP (всего 18 баллов) — до 30.04;

TIMER, REGISTRY (всего 9 баллов) — до 14.05;

GRAPHICS, MDIFORMS (всего 8 баллов) — нет ограничений.

Общие требования к выполнению заданий

Если окно содержит компоненты, для которых имеет смысл изменять размер, то оно не должно разворачиваться на весь экран и иметь переменный размер. Исключение составляют проекты WPF, содержащие однострочные поля ввода, для которых следует предусмотреть возможность увеличения их ширины при вводе большого текста (с одновременным увеличением ширины окна).

При размещении компонентов в окне следует добиваться оптимального использования пространства окна. В случае окон переменного размера для Windows Forms следует использовать свойства Anchor или Dock; в окнах WPF следует использовать подходящие групповые компоненты и пользоваться свойствами Margin и Padding (при этом свойство Margin **не должно** использоваться в проектах WPF в качестве аналога свойств Left и Top для проектов Windows Forms).

Все действия, предусмотренные в проекте, должны иметь связанные с ними клавиши-ускорители, информация о которых должна быть доступна в самой программе: в виде подчеркнутых символов в надписях компонентов, указания горячих клавиш в скобках (например, «Выполнить (F9)»), применения клавиш по умолчанию. Обход компонентов с помощью клавиши Tab должен осуществляться в естественном порядке: справа налево и сверху вниз.

Программы не должны содержать однотипные обработчики; вместо этого надо создавать обработчики, связываемые с несколькими компонентами. Однотипные действия над несколькими компонентами следует выполнять в цикле, используя для Windows Forms свойство окна Controls, а для WPF — свойства-коллекции групповых компонентов.

В проектах для WPF рекомендуется активно использовать привязку свойств (binding), а также механизм команд (Commands); общие обработчики для нескольких компонентов желательно настраивать с помощью метода SetHandler.

Имена компонентов, предлагаемые по умолчанию в проектах Windows Forms, изменять не следует, за исключением имен пунктов меню и кнопок быстрого доступа. В проектах WPF имена компонентов также следует формировать по схеме имен для Windows Forms (имя типа, начинающееся с маленькой буквы, за которым следует порядковый номер компонента в окне, например button1 — за исключением имен для пунктов меню и кнопок быстрого доступа). Те компоненты проектов WPF, к которым не предполагается обращение в коде (и которые не применяются для привязки свойств), именовать не обязательно.

Требования к оформлению выполненных заданий

При подготовке выполненного задания к отправке в систему EDU-MMCS его надо оформить в виде zip-архива, имя которого совпадает с названием задания (название формируется по названию темы и номеру задания в пределах темы, например CONSOLE2). Если задание требовалось доработать, то к имени добавляется латинская буква, обозначающая этап доработки (CONSOLE1a.zip, CONSOLE1b.zip и т. д.).

Zip-архив должен содержать следующие элементы:

- файл info.txt с данными студента (фамилия, имя, номер группы, номер варианта); если задание было доработано, то в этом файле надо указать, что именно было исправлено по сравнению с предыдущим вариантом)
- каталог WinForms с реализацией задания для Windows Forms (или с реализацией консольного приложения) — для заданий 1–4, 6–8;
- каталог WPF с реализацией задания для WPF — для заданий 2–5.

Каждый из каталогов WinForms и WPF должны содержать следующие каталоги:

- каталог EXE, содержащий откомпилированную программу, готовую к запуску. Если для нормальной работы программы (или для проверки некоторых ее режимов) необходимо иметь дополнительные файлы (или каталоги, в том числе вложенные), то эти файлы и каталоги тоже должны размещаться в данном каталоге. Для консольных программ в этот каталог следует добавить bat-файлы, позволяющие быстро запустить программу с дополнительными параметрами командной строки. Необходимо, чтобы правильность работы консольной программы могла быть проверена вне зависимости от того, на какой диск и в какой каталог будет распакован каталог EXE. Если программа работает неверно, то задание не засчитывается и возвращается на доработку (без анализа программного кода);
- каталог SOURCE, содержащий все файлы проекта в том виде, в котором они были созданы Visual Studio; в данном каталоге уже не требуется размещать откомпилированный файл программы. Программный код анализируется при условии правильной работы представленной программы.