

Задание №3. Верстка пользовательского интерфейса

Описание задания:

1. Создайте новую ветку в локальном репозитории под названием «features/2_add_forms_layout». Перейдите в новую ветку.
2. В проекте View переименуйте Form1 в MainForm. При переименовании форм и других элементов в коде стоит использовать стандартную функцию переименования, а не исправлять название вручную через редактор кода. Для этого кликните правой кнопкой по названию класса на панели Solution Explorer или в редакторе кода, и в контекстном меню выберите пункт «Переименовать» (Rename). Укажите новое название и нажмите Enter. Убедитесь, что переименовался и класс в исходном коде, и сам файл в дереве решения.
3. Откройте дизайнер форм для MainForm и выполните верстку согласно ТЗ. На ваше усмотрение вы можете реализовать верстку одним из трёх способов или их комбинацией: а) только с помощью Anchor; б) с использованием SplitContainer; в) с использованием TableLayout.
4. Убедитесь, что верстка главного окна адаптивная («резиновая»), и ведет себя корректно при изменении размеров окна. Это можно сделать в дизайнера форм или запустив программу на исполнение.
5. Убедитесь, что все элементы окна выровнены, имеют одинаковое расстояние между собой.
6. Убедитесь, что подписи элементов не содержат грамматических ошибок.
7. Переименуйте все элементы управления согласно требованиям к именованию. Не должно быть элементов с названием listBox1, label5 и т.д. Каждый элемент должен отражать свой тип и своё назначение. Например, NotesListBox, SurnameLabel и т.д.
8. Если работа по верстке главного окна завершена, сделайте промежуточную фиксацию в локальный репозиторий. Оставьте комментарий, например «ui(MainForm): добавлена верстка главного окна». Синхронизируйтесь с удаленным репозиторием.
9. Добавьте в проект вторую форму согласно ТЗ. Вторая форма предназначена для редактирования данных пользователя. В зависимости от варианта, это может быть заметка (Note), контакт человека (Contact) и т.п. Название формы должно формироваться из названия редактируемого объекта и добавления слова «Form» как указания, что этот класс является формой. Например, если форма предназначена для редактирования заметок, тогда название формы должно быть NoteForm. Если форма предназначена для редактирования

информации о городах, тогда название формы должно быть CityForm и т.п. Обратите внимание на единственное число (NoteForm, а не NotesForm), и отсутствие вспомогательных глаголов (типа AddNoteForm, EditForm и т.д.).

10. Откройте дизайнер форм для второй формы и выполните верстку согласно ТЗ. Убедитесь, что верстка адаптивная (если это требуется). Проверьте, что все элементы выровнены относительно друг друга, а также в отсутствии грамматических ошибок.

11. Переименуйте все элементы управления согласно требованиям к именованию.

12. Если работа по верстке второго окна завершена, сделайте промежуточную фиксацию в локальный репозиторий. Оставьте комментарий, например «ui(NoteForm): добавлена верстка окна». Синхронизируйтесь с удаленным репозиторием.

13. По аналогии с предыдущими окнами, добавьте окно AboutForm согласно ТЗ. Проверьте верстку, грамматические ошибки, именование. Если все сделано верно, сделайте фиксацию с версткой третьего окна.

14. Чтобы убедиться, что верстка всех окон работает правильно не только в дизайнера форм, но и в исполняемом приложении, добавьте в главное окно обработчики для кнопок, вызывающих второстепенные окна. Второе окно (например, NoteForm) должно открываться в диалоговом режиме, если пользователь нажмет кнопку «Add» в главном окне. Окно должно закрываться при нажатии на кнопки «OK» или «Cancel». Окно About должно открываться при вызове его из главного меню главного окна.

15. Запустите приложение, убедитесь, что по нажатию на соответствующие кнопки окна открываются и закрываются. Убедитесь, что верстка окон работает правильно. Если в ходе проверки были обнаружены недочеты, исправьте их. По завершению сделайте фиксацию в репозиторий.

16. В свойствах проекта View добавьте иконку приложения (её можно найти, например, на <https://icons8.com/>). Также добавьте иконку для главного окна. Сделайте фиксацию с добавленными иконками. Синхронизируйтесь.

17. В Visual Studio перейдите в ветку develop. Затем выполните слияние (merge) ветки features/2_add_forms_layout в ветку develop. Теперь все изменения из ветки с добавленным решением должны переместиться в ветку develop. Убедитесь в этом с помощью GitHub.

Типовые ошибки и возможные проблемы:

Наиболее частая проблема – студент решил переименовать форму не с помощью встроенных инструментов, а вручную. Так как форма состоит из нескольких файлов, то ручное переименование класса только в одном файле приведет к моментальной ошибке в среде разработки с возможной потерей верстки окна (если дизайнер форм при этом открыт на одной из вкладок).

Чтобы избежать проблемы, используйте встроенные инструменты переименования, как это описано в задании, или инструменты переименования Resharper.

Если проблема уже возникла, то вам необходимо:

- открыть папку с формой в Проводнике;
- скопировать три файла целевой формы в отдельную папку (вне папки репозитория);
- переименовать файлы согласно правильному названию формы;
- открыть файлы и переименовать класс формы внутри исходного кода файлов;
- в Visual Studio удалить из проекта «сломанную» форму;
- добавить в проект переименованную форму - «Добавить существующий элемент» и выбрать файл с бизнес-логикой формы, например, NoteForm.cs.

При добавлении файла с бизнес-логикой формы, файл верстки (NoteForm.designer.cs) и файл ресурсов формы (NoteForm.resx) подключатся в проект автоматически.

Вторая частая проблема – неправильное именование форм и элементов управления на формах. Аргументация правильного именования приведена в тексте задания. Пожалуйста, ознакомьтесь с ней внимательно и проверьте свою работу еще раз.

Защита задания:

1. Во время защиты студент должен показать правильную верстку всех окон приложения.
2. Названия всех трёх форм должны быть корректными, согласно заданию.
3. Названия форм в исходном коде и названия файлов форм должны совпадать.
4. У проекта View должна быть настроена иконка в свойствах проекта.
5. У главного окна должна быть добавлена иконка, а заголовок должен быть назван в соответствии с версией программы.
6. Приложение должно компилироваться и запускаться.
7. Проверяющий должен иметь возможность открыть любое окно в исполняемом приложении, чтобы убедиться в корректности верстки.
8. Все элементы на всех формах должны быть выровнены относительно друг друга, должны соблюдаться одинаковые расстояния между элементами.
9. Верстка главного окна (обязательно) и второго окна (в зависимости от ТЗ) должны быть адаптивными.
10. Если в макете интерфейса кнопки представляются в виде пиктограмм и отображены в плоском стиле, то и в верстке форм кнопки должны быть плоскими в виде пиктограмм.
11. Во время защиты студент должен показать, что все элементы управления на формах названы правильно.
12. На формах не должно содержаться лишних/избыточных элементов, форма должна быть сверстана лаконично. Все элементы, которые студент поставил случайно и забыл удалить, должны быть удалены.
13. Студент должен показать на GitHub, что при выполнении задания была создана отдельная ветка `features/2_add_forms_layout`, и она была слита в `develop`.