**Задание 03.**

В интегрированной среде разработки *Microsoft Visual Studio* на языке *Visual C#* разработать графический пользовательский интерфейс, состоящий из двух экранных форм (основной и дочерней), все элементы управления которого полностью настраиваются кодом без использования графического конструктора.

Иначе:задача состоит в полной автоматизации компоновки графического пользовательского интерфейса.

По итогам выполнения работы сдаются строго два файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл с архивом всего решения (*Solution*), написанного на языке *Visual C#* в среде *Microsoft Visual Studio* в режиме *Windows Forms Application* (*\*.zip* или *\*.rar*).

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

Раздел «Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению» наполнить демонстрацией скриншотами выполнения каждого из пунктов раздела «Требования к оформлению» из данной постановки задачи.

**Требования к именам файлов:**

**Общий вид формата имени файла:** «*Дата. Задание. Фамилия.mcdx*»

**Формат записи даты:** «*ГГГГММДД*», где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** «Задание *NNk*», где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые\*:**

«*20240328. Задание 03и. Иванов.docx*»

«*20240328. Задание 03и. Иванов.rar*»

«*20240328. Задание 03и. Иванов.vsdx*»

**Примеры правильных имён файлов, которые сдаются на проверку повторно в тот же день\*:**

«*20240328. Задание 03и. Иванов (1).docx*»

«*20240328. Задание 03и. Иванов (1).rar*»

«*20240328. Задание 03и. Иванов (1).vsdx*»

\* Правила распространяются только при обмене файлами через электронную почту. При работе на платформе *GitHub* устранение замечаний проводится в одних и тех же файлах без необходимости их переименования

**Внимание!** Не забудьте выполнить автоматическую нумерацию страниц в отчёте.

Отчёт по выполненной работе должен содержать:

0. Титульный лист.

1. Формулировку цели работы.

2. Описание задачи согласно выданному варианту.

3. Составление диаграммы классов, входящих в состав решения.

4. Составление сети Петри запрограммированного технологического процесса.

5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ).

6. Подбор тестовых примеров.

7. Листинг (код) составленного программного обеспечения.

8. Графический пользовательский интерфейс программного обеспечения и его описание.

9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

10. Расчёт тестовых примеров с использованием составленного программного обеспечения.

11. Формулировку вывода о проделанной работе.

Рекомендации к отчёту, доказывающие самостоятельность выполнения работы и упрощающие процедуру проверки отчёта преподавателем:

1. Выполнение дополнительных скриншотов для случаев, когда текстовое описание проделанных действий становится громоздким или трудным к восприятию.

2. Нумерация рисунков (если есть) с подписями, содержащими названия рисунков, например, «Рисунок 1 – Схема алгоритма программы».

**Цель работы (одна из возможных формулировок)**: «Закрепить навыки разработки визуального пользовательского интерфейса, освоить работу с текстовыми файлами в среде *Microsoft Visual Studio*, научиться взаимно увязывать одни элементы управления с другими, получить представление о перерисовке и перемасштабировании».

**Основное задание на работу**: «В интегрированной среде разработки *Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой пользовательский интерфейс, содержащий главное меню, позволяющее:

1. Начать работу с приложением.
2. Прервать работу приложения.
3. Предоставить пользователю справочную информацию о работе с приложением.

Сама программа должна реализовывать вывод в списки значений аргумента и соответствующих им значений функций. Список функций должен обязательно содержать следующие пункты:

1. Логарифм по основанию 2,
2. Тангенс,
3. Возведение в квадрат,
4. Косинус,
5. Натуральный логарифм,
6. Арккосинус,
7. Извлечение корня,
8. Арктангенс,
9. Синус,
10. Десятичный логарифм,
11. Арксинус.

Таблица 1 – Варианты расположения расчётных функций в комбинированном или обычном списке

|  |  |
| --- | --- |
| **№ вар.** | **Порядок индексов функций из списка** |
| **1** | 5, 3, 11, 4, 1, 9, 10, 6, 7, 8, 2. |
| **2** | 4, 3, 8, 7, 11, 2, 10, 9, 6, 1, 5. |
| **3** | 10, 9, 11, 8, 6, 5, 7, 4, 3, 2, 1. |
| **4** | 11, 2, 9, 4, 6, 5, 7, 8, 3, 10, 1. |
| **5** | 7, 3, 8, 4, 6, 2, 11, 9, 10, 5, 1. |
| **6** | 10, 9, 5, 1, 3, 11, 2, 6, 7, 8, 4. |
| **7** | 1, 7, 3, 9, 5, 11, 10, 4, 2, 6, 8. |
| **8** | 4, 3, 7, 5, 6, 2, 8, 9, 11, 10, 1. |
| **9** | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. |
| **10** | 7, 5, 1, 4, 9, 8, 10, 3, 11, 6, 2. |
| **11** | 8, 6, 11, 2, 9, 10, 3, 4, 5, 1, 7. |
| **12** | 2, 1, 10, 11, 5, 9, 8, 6, 7, 3, 4. |
| **13** | 11, 8, 7, 5, 9, 10, 2, 4, 1, 6, 3. |
| **14** | 8, 2, 10, 7, 3, 11, 5, 9, 6, 1, 4. |
| **15** | 9, 1, 6, 7, 4, 11, 5, 10, 8, 3, 2. |
| **16** | 5, 4, 8, 2, 3, 1, 10, 9, 6, 11, 7. |
| **17** | 11, 7, 10, 6, 9, 3, 4, 8, 5, 2, 1. |
| **18** | 1, 7, 2, 6, 5, 9, 10, 3, 11, 4, 8. |
| **19** | 6, 7, 11, 8, 9, 3, 5, 4, 10, 2, 1. |
| **20** | 3, 2, 1, 11, 6, 9, 7, 8, 4, 10, 5. |

**Требования к оформлению**:

1. Заголовок экранной формы должен содержать надпись вида: «Задание №3 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг].

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы.

3. Чётные варианты отключают стандартный блок управления экранной формой и создают авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы.

4. Нечётные варианты заменяют стандартный курсор экранной формы со «стрелки» на «руку».

5. Справочная информация должна быть вызвана в дочерней экранной форме и считана в статический по размеру ярлык из текстового файла (чётные варианты) или в статическое по размеру текстовое поле с выставленным запретом на редактирование текстовой информации (нечётные варианты).

6. В режиме конструктора экранная форма не должна содержать ни одного элемента управления.

7. Решение (*solution*) должно содержать только одну единственную экранную форму. Экранная форма со справкой настраивается программным кодом.

8. Компоненты экранной формы со справкой должны быть чувствительными к изменению размера экранной формы по ширине и высоте (обязательно подключение события *Resize*).

9. Названия функций должны быть элементами комбинированного списка (чётные варианты) или обыкновенного списка (нечётные варианты).

10. Выбор конкретной функции из списка инициирует заполнение некоторого файла значениями аргумента и значениями функции. Разделитель данных подбирается согласно таблице вариантов.

Таблица 2 – Варианты разделителей, записываемых в файл

|  |  |
| --- | --- |
| 1. | $ |
| 2. | % |
| 3. | &\* |
| 4. | <> |
| 5. | { |
| 6. | $#@ |
| 7. | ^^ |
| 8. | ( |
| 9. | . |
| 10. | >> |
| 11. | , |
| 12. | ; |
| 13. | : |
| 14. | ++ |
| 15. | = |
| 16. | !@ |
| 17. | ||| |
| 18. | пробел |
| 19. | \_ |
| 20. | … |

11. Нечётные варианты заменяют цвет подложки экранной формы в соответствии с представленной ниже таблицей.

Таблица 3 – Варианты цветового оформления подложки

|  |  |
| --- | --- |
| 1. | *ActiveCaption* |
| 3. | *AppWorkspace* |
| 5. | *ControlText* |
| 7. | *Desktop* |
| 9. | *GradientActiveCaption* |
| 11. | *Highlight* |
| 13. | *HotTrack* |
| 15. | *Info* |
| 17. | *MenuHighlight* |
| 19. | *Window* |

Подобрать читаемый цвет текста к предложенной по варианту схеме.

12. Предусмотреть кнопку, нажатие на которую инициирует заполнение двух списков (один список под значения аргумента, другой список – под значения функции, округлённых по правилу *(N mod 5) + 1* для знаков после плавающей запятой, где *N* – номер варианта) данными о функциональной зависимости из файла. Обучающиеся с чётными номерами вариантов обновляют существующие основные списки, с нечётными номерами вариантов выводят значения, считанные из файла, в дополнительные списки.

13. В качестве исходного состояния функциональных компонентов главной экранной формы принимается либо недоступное состояние (*.Enabled = false*) – чётные варианты, либо невидимое состояние (*.Visible = false*) – нечётные варианты.

14. Кнопка «Начало работы с приложением» должна реализовывать активацию доступа пользователя к функциональным компонентам или отображение функциональных компонентов на экрвнной форме для пользователя.

15. Все элементы программы должны носить значащие имена переменных, в которых отражено существо этих элементов, например, экранная форма – *frmMain*, ярлык – *lblHelp*, комбинированный список – *cmbFigures* и т.д.

16. Выбор значения абсциссы в списке должен инициировать выбор соответствующего ему значения ординаты и наоборот – выбор значения ординаты должен инициировать выбор соответствующего ему значения абсциссы (работа с параметрами *SelectedIndex* или *SelectedItem* списков).

**Варианты индивидуального задания:**

Таблица 4 – Варианты дополнительных функциональных зависимостей, рассчитываемых программой

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Поз.** | **Функция** | **Поз.** |
| 1. 1 / log2x | 3 | 13. (x + 1) / ((x2 - 3·x)·(x - 0,3)) | 7 |
| 2. (x2 - x + 7) / (x + 1) | 1 | 14. |log2 |x| - 1| | 10 |
| 3. ((1 – x) / (1 + x))4/5 | 8 | 15. (3√x2) / (x2 - 1) | 3 |
| 4. 2 – (1 / (2x – 1)) | 4 | 16. x / (x2 - 5) | 2 |
| 5. 1 / cos(x) | 2 | 17. log2((x + 1) / (x - 1)) | 4 |
| 6. 1 / sin(x) | 5 | 18. log2 x / (log2 x - 3) | 8 |
| 7. 2·x – (1 / (2·x – 3)) | 11 | 19. |x| / (1 + x) | 9 |
| 8. log-21/3 3· (x2+2·x) | 9 | 20. 1 / tg(x) | 2 |
| 9. 1 / (2·x – 3) | 6 | 21. x / ((x2 - 5·x + 6)·(x + 0,3)) | 11 |
| 10. log1/2 (x2+2·x) | 8 | 22. x / sin(x) | 7 |
| 11. x / cos(x) | 10 | 23. x / (√(x2 - 5) | 3 |
| 12. ln(x) / x | 4 | 24. 1 / (√(x2 - 1) | 1 |