**Лабораторная работа №3: Основы проектирования приложений Windows Forms**

**Контрольные вопросы**

*1. Какие файлы описывают класс формы?*

Файлы Program.cs, Form1.cs (либо другое название формы), Form1.Designer.cs и Form1.resx.

*2. Какие действия необходимо выполнить для создания обработчика события?*

Для добавления обработчика события мыши необходимо выбрать соответствующее событие (Click, DoubleClick, MouseEnter, MouseLeave, MouseDown, MouseUp, MouseMove,. MouseHover, MouseWheel) и дважды щелкнуть на поле, расположенном в соседнем от него столбце в окне свойств (таким образом обработчику присвоится имя по умолчанию). В данном поле можно ввести свое название для нового обработчика, или выбрать из выпадающего списка уже существующие обработчики.

*3. Где описывается код обработчика события? В каком файле регистрируется обработчик события (метод привязывается к событию)?*

После регистрации события в окне «Properties», автоматически добавляются строки кода в файлы Form1.Designer.cs (регистрируется обработчик события) и Form1.cs (добавляется метод, ассоциированный с данным событием).

*4. Как получить доступ к координатам курсора мыши?*

Необходимо обратиться к методам X и Y класса MouseEventArgs.

*5. Какой класс содержит методы, реализующие математические функции?*

Класс Math.

**Индивидуальное задание**

Было выполнено индивидуальное задание по выводу координат курсора мыши в заголовок формы. Также в текстовое поле формы выводится сумма координат курсора мыши (в textBox1). Код представлен ниже.

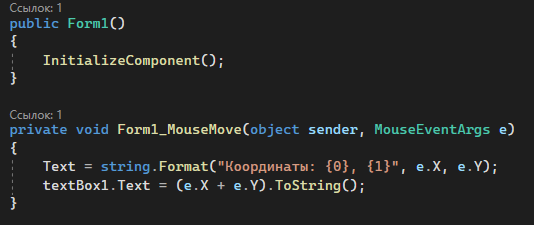


Рисунок 1. Код первого индивидуального задания

Пример работы программы представлен ниже (см. рис. 2).

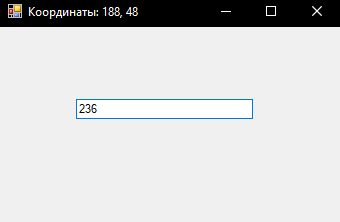


Рисунок 2. Пример работы программы первого задания

Также было выполнено индивидуальное задание по выводу результата выражения (см. рис. 3). Переменным F и W были присвоены значения координат курсора мыши X и Y, соответственно. Значения переменных E и T пользователь может ввести самостоятельно (есть значения по умолчанию). Результат выводится в заголовок окна. Если T = 0, то будет выводиться ERROR, так как деление на 0 невозможно.





Рисунок 3. Код второго индивидуального задания

Пример работы программы представлен ниже (см. рис. 4).

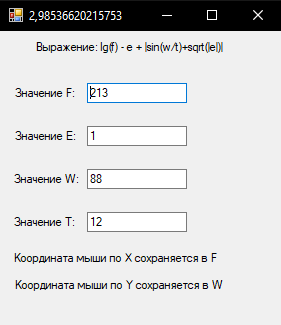


Рисунок 4. Пример работы программы второго индивидуального задания

Для удобства тестирования программ в Program.cs был добавлен вызов формы второго задания после закрытия формы первого задания (см. рис. 5). Результаты выполнения сохранены в соответствующем репозитории на Github.

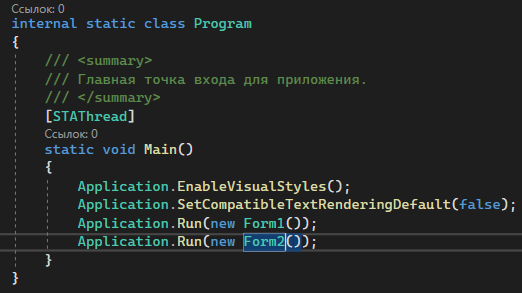


Рисунок 5. Изменённый код Program.cs