

## Практическое занятие № 2

**Тема практического занятия:** Среда разработки.

**Цель практического занятия:** овладение навыками в среде разработки.

**В результате выполнения данной работы обучающийся должен уметь:**

1. Запускать среду разработки и создавать проект.
2. Настраивать среду разработки для комфортной работы.
3. Сохранять проект.
4. Создавать графический интерфейс с помощью конструктора.

**знать:**

1. Основные возможности изучаемой среды разработки.
2. Структуру проекта, создаваемого в среде разработки.
3. Основные элементы управления, позволяющие создавать графический интерфейс.

**Перечень оборудования, необходимого для выполнения задания:**

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Монитор 34", изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).

Программное обеспечение:

- Microsoft Visual Studio.

**Общие теоретические сведения:**

Интегрированная среда разработки (Integrated Development Environment, IDE) – это программное обеспечение, которое объединяет различные инструменты и функциональные возможности для разработки программного обеспечения в одном комплексе. Она предоставляет средства разработки, отладки, компиляции, анализа кода, управления проектами и другие полезные функции, что позволяет программистам эффективно создавать, отлаживать и поддерживать программы.

Интегрированная среда разработки обычно включает в себя:

1. Текстовый редактор, который обеспечивает удобное написание и редактирование исходного кода.
2. Компилятор или интерпретатор для преобразования исходного кода в исполняемый файл или байт-код.
3. Отладчик, который помогает выявлять и исправлять ошибки и неправильное выполнение программы.

4. Средства автодополнения, проверки синтаксиса и другие функции, которые помогают повысить производительность и качество кода.

5. Управление версиями, что позволяет отслеживать изменения в коде, реализовывать совместную работу и восстанавливать предыдущие версии кода.

6. Встроенную поддержку различных языков программирования и фреймворков.

Примеры сред разработки:

- Eclipse IDE
- Microsoft Visual Studio,
- NetBeans,
- Android Studio,
- IntelliJ IDEA.

Современные среды разработки позволяют конструировать графический интерфейс с помощью наборов элементов управления. Такой интерфейс будет являться событийно-ориентированным.

В событийно-управляемом интерфейсе, приложение обычно содержит набор элементов управления (кнопки, поля ввода, полосы прокрутки и т. д.), на которые пользователь может действовать. При возникновении события, такого как нажатие кнопки или изменение значения поля ввода, программа запускает соответствующий обработчик события, который определен заранее. Обработчик события выполняет нужные действия, например, обновление данных, вывод информации на экран или вызов других функций.

### **Задание:**

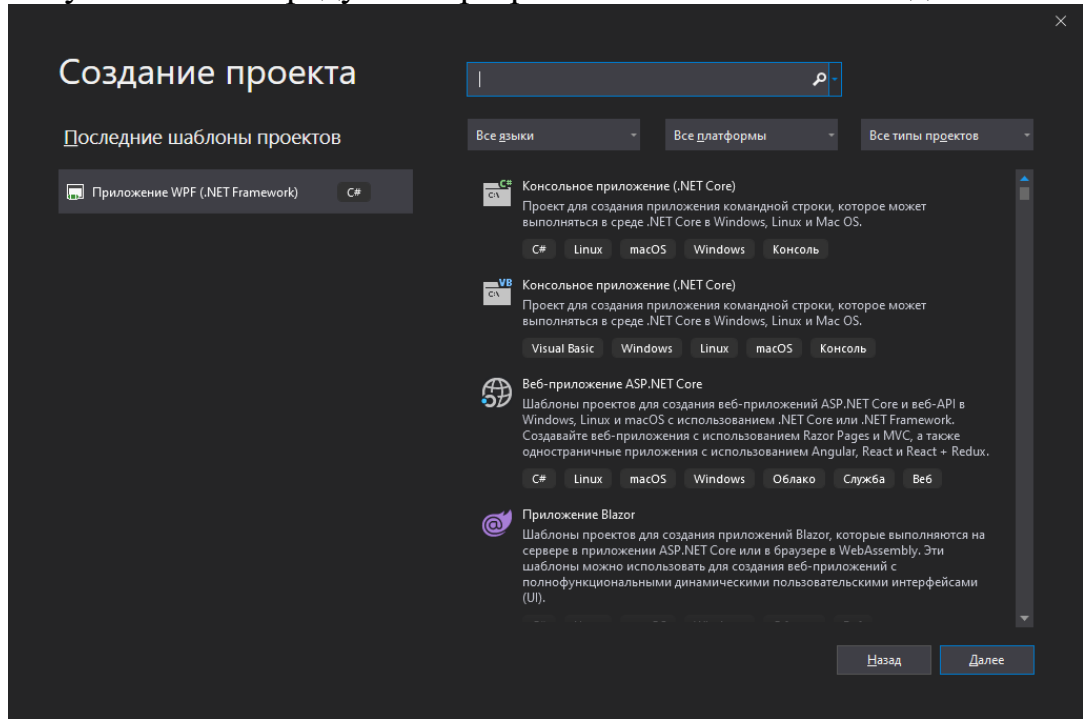
1. Создайте многооконное приложение в среде разработки.
2. Приложение должно состоять из одного главного окна и трех вспомогательных.
3. Главное окно должно содержать три кнопки, каждая из которых соответствует одной из задач.
4. Вспомогательные окна кроме элементов, предусмотренных заданием, должны содержать кнопку возврата к главному окну.
5. Задача первого окна. В окне должны быть расположены надпись «Добрый день» и красная кнопка, на которой написано «Не нажимать!». При нажатии на кнопку надпись должна меняться на «Написано же: «Не нажимать!», кнопка должна исчезать.
6. Задача второго окна. В окне должен быть расположен «светофор» из трех ламп, гореть должна только красная (лампы могут быть круглыми или прямоугольными). Рядом с каждой лампой должна быть расположена кнопка, с помощью которой можно эту лампу включить. При включении лампы все остальные должны погаснуть.
7. Задача третьего окна. В окне должна быть расположена кнопка с надписью «Нажми меня!». При нажатии на кнопку она должна перемещаться в другое место окна.

## Указания по технике безопасности:

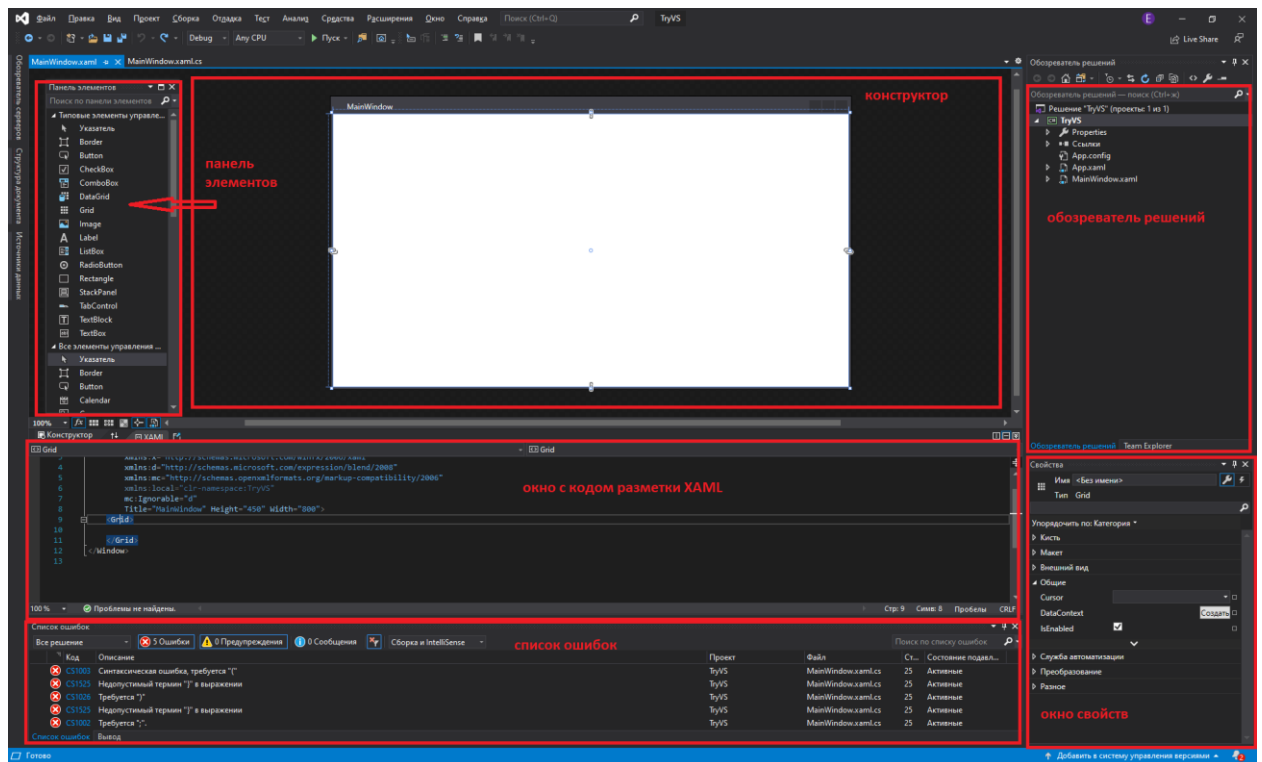
Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

**Технология выполнения работы (этапы, последовательность действий):**

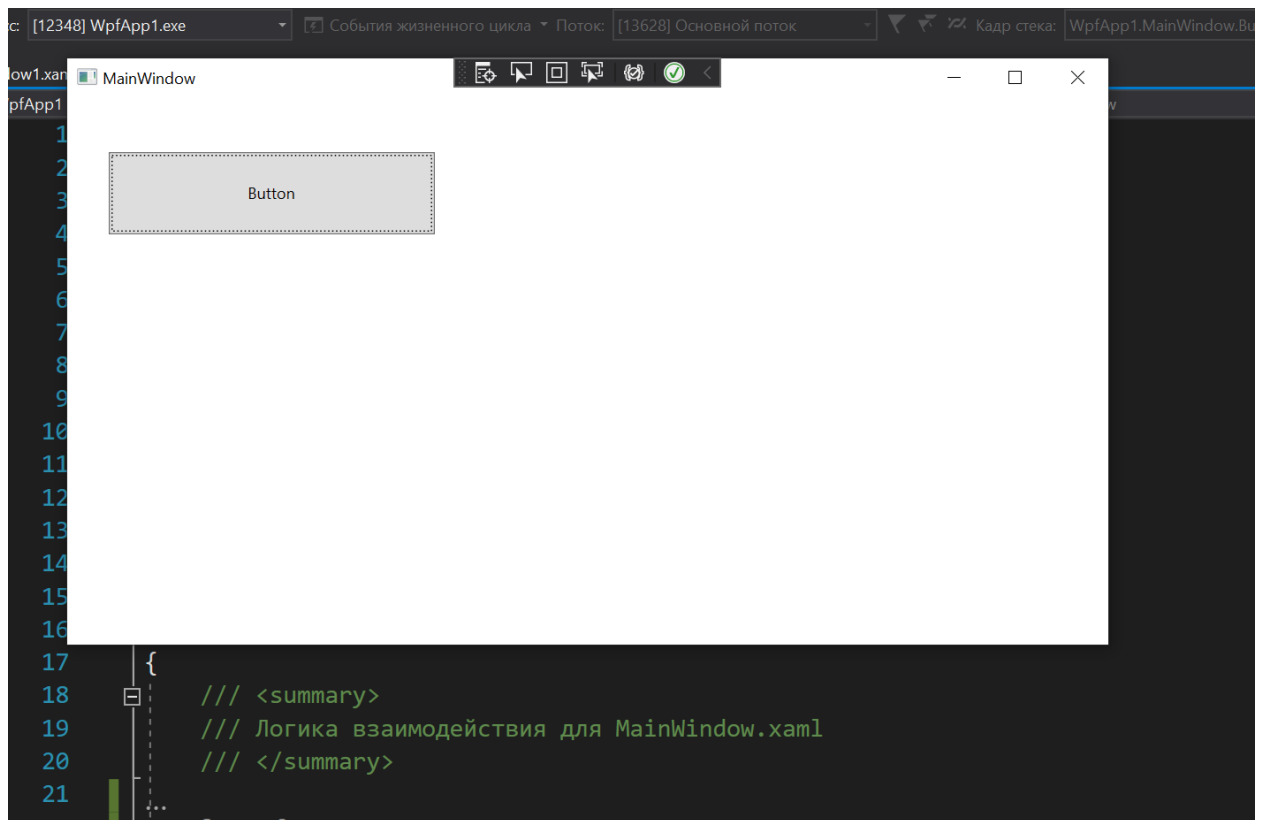
1. Запустите среду разработки и создайте проект



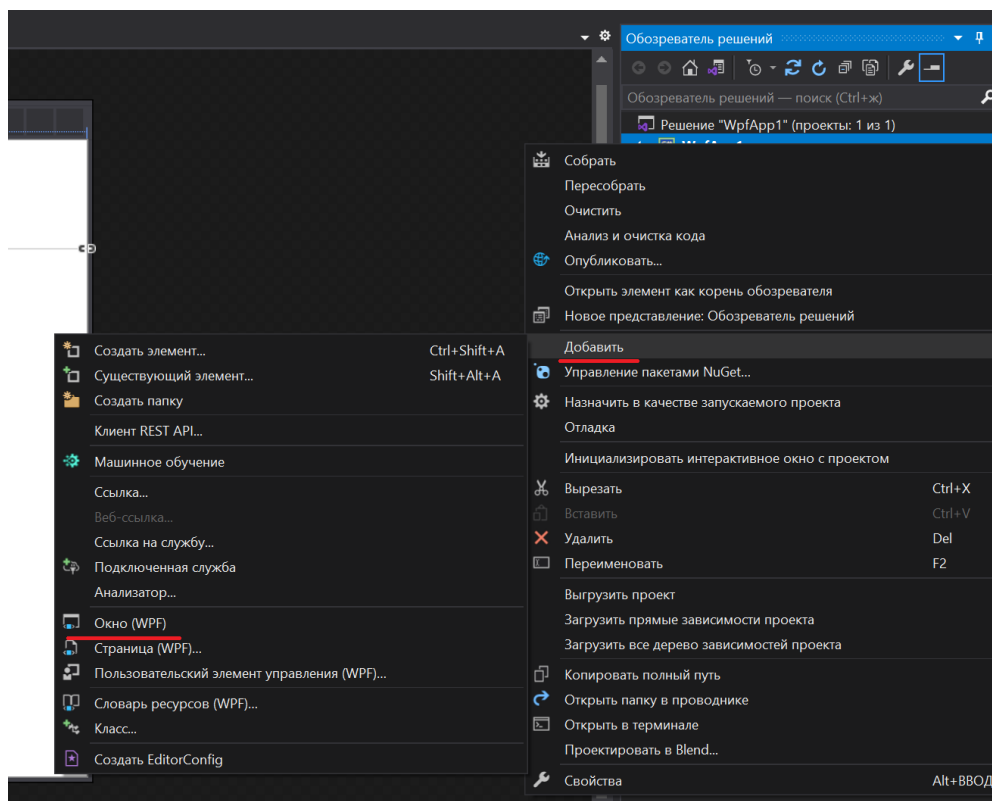
2. В главном окне вы увидите конструктор интерфейса:



3. Разместите три кнопки для перехода к другим окнам. Кнопка выглядит вот так:



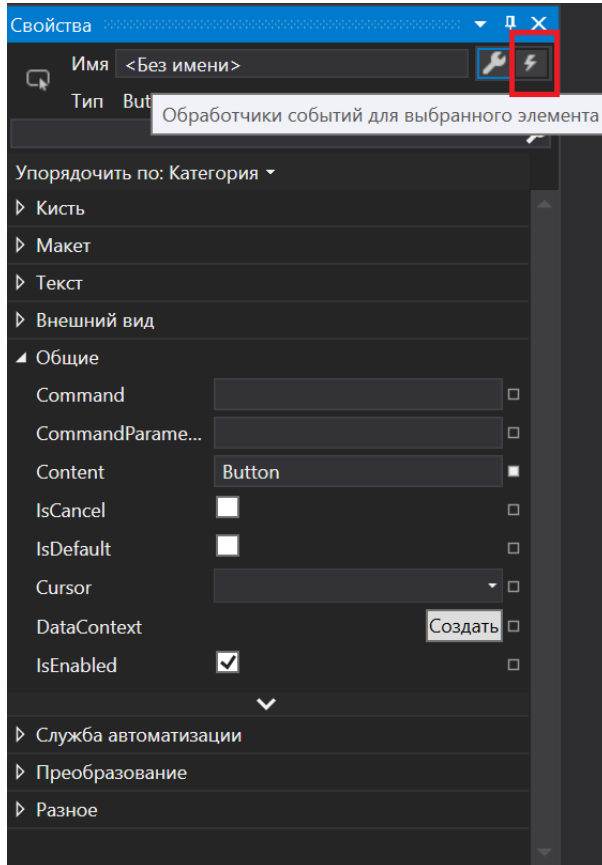
4. Добавьте в проект три вспомогательных окна.



5. Напишите обработчики событий для перехода к другим окнам. Пример обработчика событий для перехода к окну:

```
ссылка: 1
private void Button_Click(object sender, RoutedEventArgs e)
{
    Window1 pw = new Window1();
    pw.Show();
}
```

6. Запустите приложение. Для этого охватите ладонью компьютерную мышь сверху и, не отрывая мышь от стола, ведите курсор мыши на верхнюю часть экрана к слову «Пуск».
7. Проверьте работоспособность приложения: при нажатии кнопки должен быть обеспечен переход к соответствующему второстепенному окну. Нажимать кнопки следует, охватив ладонью компьютерную мышь сверху и, не отрывая мышь от стола, ведя ее к кнопке. Как только курсор мыши окажется на кнопке, необходимо нажать левую кнопку мыши.
8. Оформите интерфейс первого окна в соответствии с заданием.
9. Добавьте к кнопке обработчик события, в котором будет удаляться кнопка и меняться надпись в текстовом поле.



10. Добавьте к первому окну кнопку возврата в главное окно и напишите соответствующий обработчик событий.
11. Запустите приложение и проверьте работоспособность исчезающей кнопки. Для этого попробуйте нажать на кнопку способом, описанном в пункте 7 (с помощью левой кнопки мыши). Если код написан верно, кнопка должна исчезнуть, то есть, вы не должны ее видеть в окне приложения.
12. Оформите интерфейс второго окна в соответствии с заданием.
13. Добавьте к кнопкам управления светофором обработчики события, которые будут раскрашивать нужную лампу в соответствующий цвет (красный, желтый или зеленый) и гасить все остальные (красить в серый).
14. Добавьте ко второму окну кнопку возврата в главное окно и напишите соответствующий обработчик событий.
15. Запустите приложение и проверьте работоспособность светофора: при нажатии на кнопку, соответствующую красному свету светофора, должен подсвечиваться элемент красного цвета, при этом все остальные элементы должны быть серыми; при нажатии на кнопку, соответствующую желтому свету светофора, должен подсвечиваться элемент желтого цвета, при этом все остальные элементы должны быть серыми, при нажатии на кнопку, соответствующую зеленому свету светофора, должен подсвечиваться элемент зеленого цвета, при этом все остальные элементы должны быть серыми. Способ нажатия на кнопку описан в пункте 7.
16. Оформите интерфейс третьего окна в соответствии с заданием.
17. Добавьте к кнопке обработчик события, в котором кнопка будет перемещаться на новое место.
18. Добавьте к первому окну кнопку возврата в главное окно и напишите соответствующий обработчик событий.
19. Проверьте работоспособность перемещения кнопки: при нажатии на кнопку она должна менять свое положение в окне. Способ нажатия на кнопку описан в пункте 7.
20. Проверьте работоспособность возврата в главное окно из вспомогательных. Для этого нажимайте на кнопки перехода во вспомогательные окна, находясь в главном окне, и на кнопку перехода в главное окно, находясь во вспомогательных окнах. Способ нажатия на кнопку описан в пункте 7.
21. Сохраните проект. Для этого в меню выберите пункт «Файл», далее – действие «Сохранить все».

### **Требование к отчету:**

1. Приложение собирается и компилируется без ошибок, запускается.
2. Из главного окна есть переход в три вспомогательных.
3. При возврате в главное окно из вспомогательного вспомогательное окно закрывается.
4. В первом окне при нажатии на кнопку кнопка исчезает, меняется надпись. Кнопка не должна проявляться при наведении на нее мышью.

5. Во втором окне при нажатии на кнопку управления лампой загорается соответствующая лампа, при этом остальные лампы гаснут. При попытке включить уже зажженную лампу ничего не меняется.
6. В третьем окне при попытке нажать на кнопку кнопка перемещается.

### **Контрольные вопросы:**

1. Для чего нужна среда разработки?
2. Как создать проект в среде разработки?
3. Что такое компилятор?
4. Из чего состоит проект? В какой папке находится исполняемый файл?
5. Приведите примеры элементов управления.
6. Как можно добавить окно в приложение?
7. Что такое обработчик события, как его создать?
8. Что такое событийно-управляемый интерфейс?

### **Основные и дополнительные источники, электронные ресурсы:**

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1913989>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1927269>.