# Working with Swing in Java

<https://netbeans.org/kb/docs/java/gui-functionality_ru.html>

Для продолжения процесса создания интерфейса необходимо создать контейнер Java, в который будут помещены другие требуемые элементы графического интерфейса. В этом действии контейнер будет создан с помощью элемента JFrame. Контейнер будет помещен в новый пакет, который будет отображаться в узле "Source Packages".

### Создание контейнера JFrame

1. В окне 'Проекты' щелкните правой кнопкой мыши узел NumberAddition и выберите Создать > Другие.
2. В диалоговом окне создания файла выберите категорию Swing GUI Forms и тип файла JFrame Form. Нажмите кнопку "Далее".
3. Введите NumberAdditionUI в качестве имени класса.
4. Выберите пакет my.numberaddition.
5. Нажмите кнопку 'Готово'.

Среда IDE создает форму NumberAdditionUI и класс NumberAdditionUI в приложении NumberAddition и открывает форму NumberAdditionUI в GUI Builder. Пакет my.NumberAddition заменяет собой пакет по умолчанию.

### Запуск программы

**Для выполнения программы в среде IDE выполните следующие действия:**

1. Выберите Run ("Запуск") > Run Main Project ("Запуск главного проекта") (как вариант, нажмите F6).
2. **Примечание.** При открытии окна с указанием того, что для Project NumberAddition не задан основной класс, следует выбрать my.NumberAddition.NumberAdditionUI в качестве основного класса в том же окне и нажать кнопку ОК.

**Для запуска программы вне среды IDE выполните следующие действия:**

1. Для сборки архива JAR приложения выберите "Run > Clean and Build Main Project" (Shift-F11).
2. При помощи проводника по файловой системе или диспетчера файлов перейдите в каталог NumberAddition/dist.
3. **Примечание.** Местоположение каталога проекта NumberAddition зависит от пути, указанного при создании проекта в шаге 3 в разделе [Упражнение 1. Создание проекта](https://netbeans.org/kb/docs/java/gui-functionality_ru.html#Exercise_1).
4. Дважды щелкните файл NumberAddition.jar.

Через несколько секунд приложение запустится.

**Примечание.** Если при двойном щелчке файла JAR не выполняется запуск приложения, дополнительные сведения о настройке связей файлов JAR в используемой операционной системе см [эту статью](https://netbeans.org/kb/articles/javase-deploy.html#troubleshooting).

Можно также запустить приложение из командной строки.

**Для запуска приложения из командной строки выполните следующие действия:**

1. Вызовите командную строку или окно терминала.
2. В командной строке измените текущий каталог на каталог NumberAddition/dist.
3. В командной строке введите следующий оператор:
4. java -jar NumberAddition.jar
5. **Примечание.** Убедитесь, что my.NumberAddition.NumberAdditionUI задан как основной класс до запуска приложения. Для провери этого, щелкните правой кнопкой узел мыши узел проекта NumberAddition на панели 'Проекты', выберите 'Свойства' во всплывающем меню и выберите категорию 'Выполнить' в диалоговом окне 'Свойства проекта'. В поле 'Основной класс' должно отображаться my.numberaddition.NumberAdditionUI.

### Обработка событий

Как выполняется обработка событий? При каждом выборе события из меню событий среда IDE автоматически создает так называемый прослушиватель событий и связывает его с компонентом разработчика. Для более подробного ознакомления с процессом обработки событий выполните следующие действия.

1. Вернитесь к файлу NumberAdditionUI.java в редакторе. Щелкните вкладку "Source" для просмотра исходного кода графического интерфейса.
2. Выполните прокрутку вниз и просмотрите реализованные методы jButton1ActionPerformed(), jButton2ActionPerformed() и jButton3ActionPerformed(). Эти методы называются обработчиками событий.
3. Теперь перейдите к методу initComponents(). Если этот метод отсутствует, найдите строку Generated Code и щелкните знак + рядом с этой строкой для отображения скрытого метода initComponents().
4. Обратите внимание на синий блок, окружающий метод initComponents(). Этот код был автоматически создан средой IDE и не может быть изменен пользователем.
5. Теперь посмотрите на сам метод initComponents(). Помимо прочего, он содержит код, инициализирующий элементы графического интерфейса и помещающий их в форму. Этот код создается и обновляется автоматически при размещении и изменении элементов в режиме проектирования.
6. В методе initComponents() найдите следующий фрагмент:
7. jButton3.setText("Exit");  
   jButton3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
    jButton3ActionPerformed(evt);  
    }  
    });
8. В этом месте к элементу графического интерфейса, в данном случае к jButton3, добавляется объект прослушивания событий "ActionListener". Интерфейс "ActionListener" имеет метод "actionPerformed" объекта "ActionEvent", который реализуется путем простого вызова обработчика событий jButton3ActionPerformed. Теперь эта кнопка реагирует на события действий. Каждый раз при нажатии кнопки создается событие "ActionEvent", которое передается в метод "actionPerformed" интерфейса прослушивания событий, исполняющий код, предусмотренный разработчиком для этого события в обработчике событий.

Как правило, для получения возможности реагирования каждый интерактивный элемент графического интерфейса должен быть зарегистрирован в каком-либо интерфейсе прослушивания событий и иметь связанный обработчик событий. Как наглядно показано, IDE NetBeans автоматически обрабатывает подключение прослушивателя событий, что обеспечивает для пользователей возможность сосредоточиться на реализации фактической бизнес-логики, которая должна инициироваться событием.