

Создание графических интерфейсов для Java-программ в NetBeans 6.0

NetBeans — это кроссплатформенная (существуют реализации как для Windows, так и для Linux) интегрированная свободно распространяемая среда разработки от фирмы Sun.

Последнюю версию NetBeans IDE 6.0 (на момент написания статьи) вы можете свободно скачать с сайта <http://www.netbeans.org>.

Список поддерживаемых NetBeans технологий выглядит внушительно: Ajax, C/C++, Databases, Debugger, Desktop, Editor, GUI Builder, Java EE, Java ME, Java SE, Javascript, Mobile, Profiler, Refactor, REST, Rich Client Platform, Ruby, SOA, SOAP, UML, Web.

В этой статье мы оценим только одно из достоинств этой среды разработки — удобство проектирования графических Java-приложений (GUI Builder). Мы будем последовательно создавать простое диалоговое окно.

Для начала создадим новый проект — File ► New project.

Мастер проведет нас через этапы несложного процесса создания нового Java-приложения (рис. 1, 2).

Мы назвали наш проект gui-tutorial и по предложению NetBeans назначили его основным проектом — теперь в окне Projects он будет выделен жирным шрифтом.

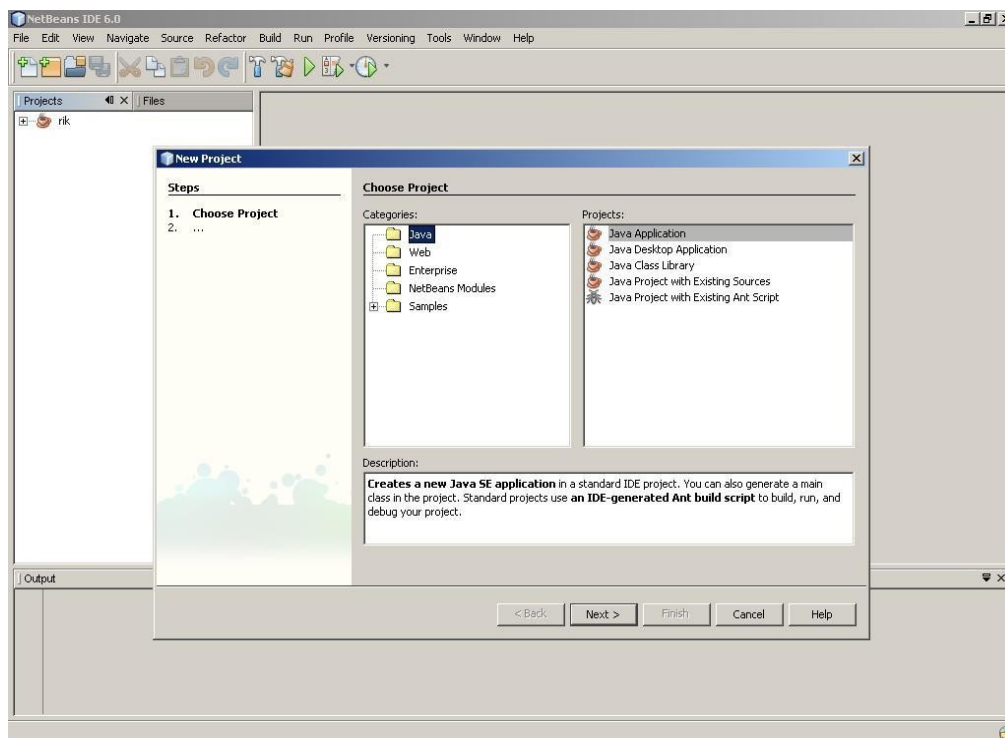


Рисунок 1. Создание Java-проекта

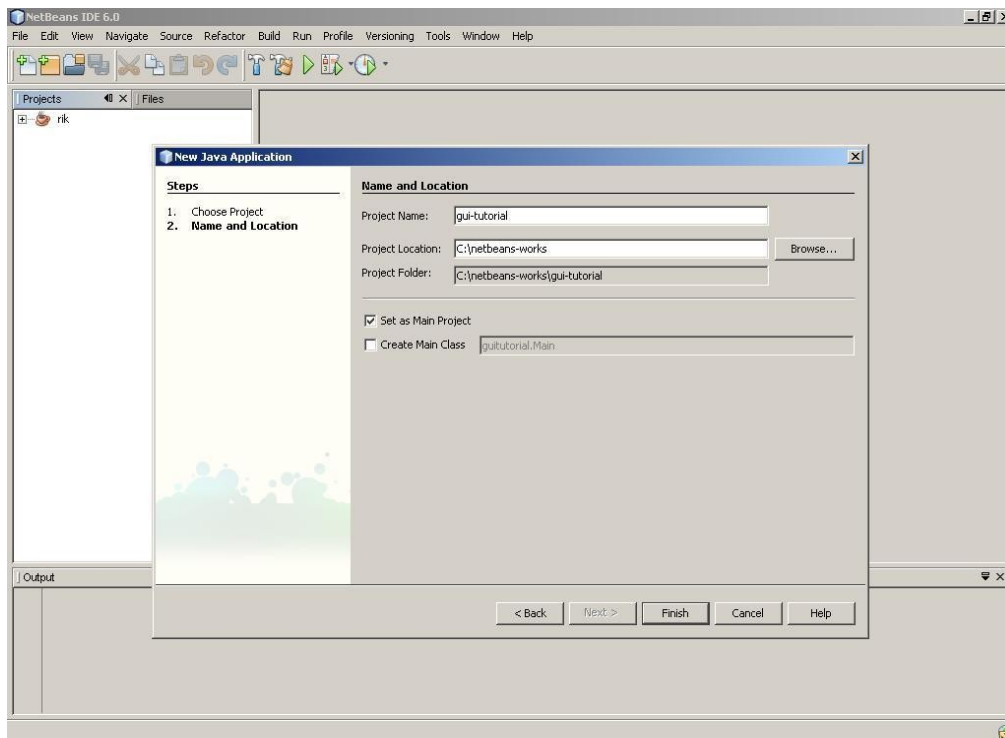


Рисунок 2. Продолжение создания проекта

Мы создадим новую панель для ввода фамилии имени и отчества и назовем ее FioPanel. Для этого нажмите правой кнопкой на имени проекта в окне Projects и выберите New ► JPanel Form (рис.3).

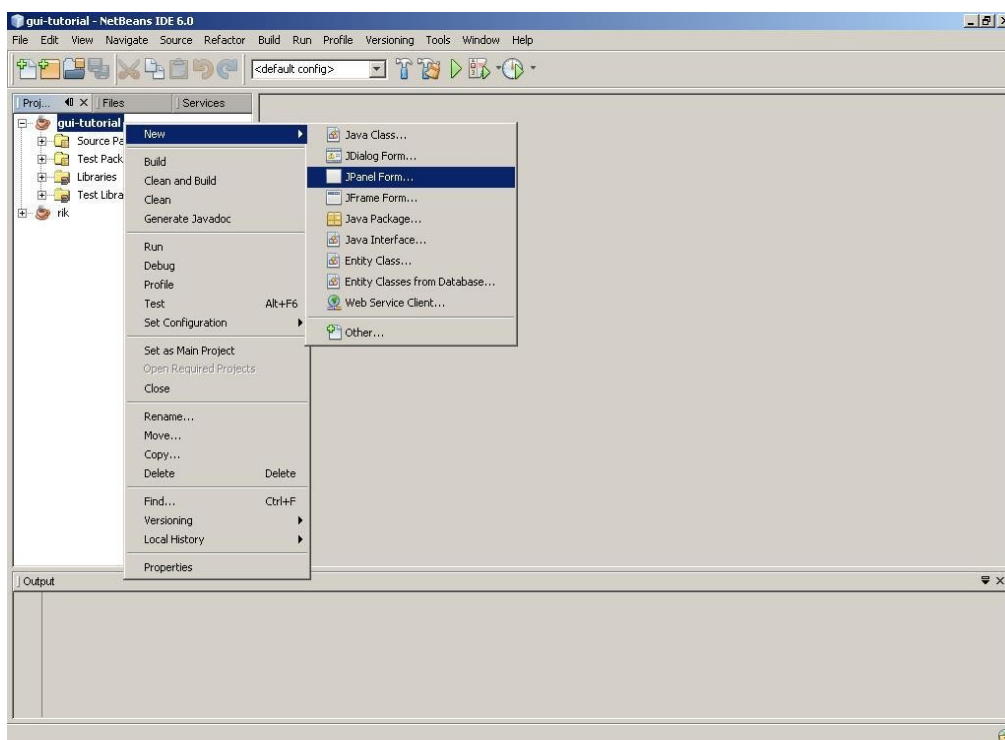


Рисунок 3. Создание панели

Введите имя панели FioPanel и нажмите Finish. В результате вы получите то, что изображено на рис. 4.

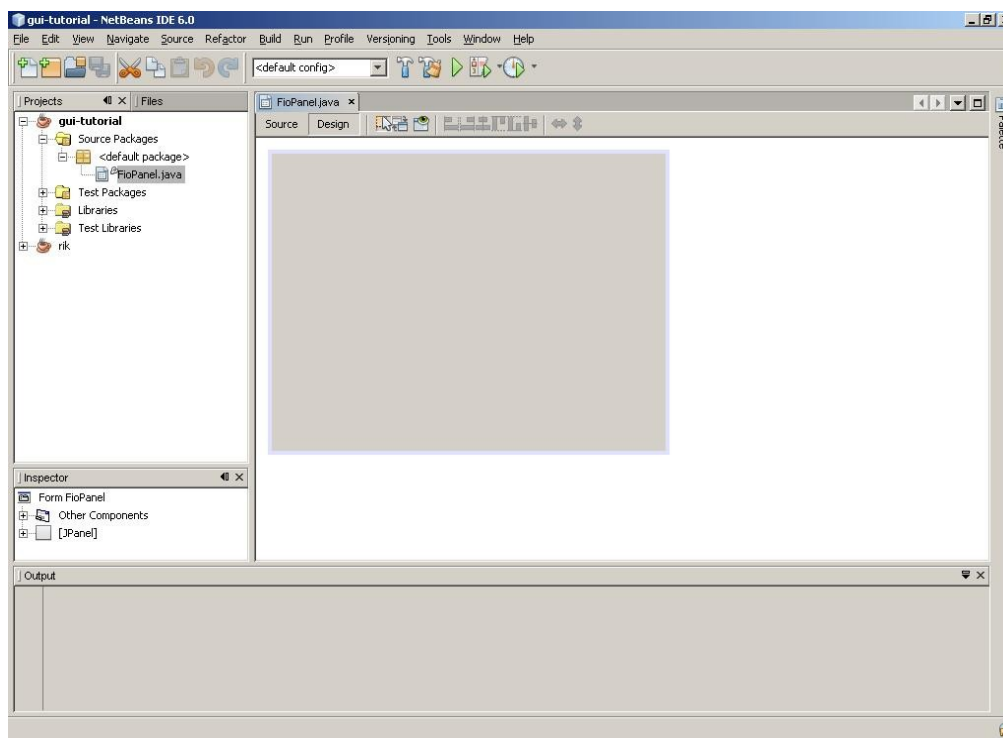


Рисунок 4. Результат создания панели

Как это принято в большинстве сред для проектирования графических приложений, вы можете работать с визуальной формой и расположенными на ней компонентами, или же непосредственно писать код. Переключение между этими режимами выполняется нажатием на кнопки **Design** и **Source** соответственно. В случае с NetBeans все ваши действия с визуальной формой сопровождаются автоматической генерацией кода, который в редакторе кода защищен от модификаций.

Первое, что мы должны сделать после создания панели, — определиться со способом компоновки элементов управления. Нетрудно догадаться, что на нашей панели будет три поля ввода. Предположим, что мы хотим расположить их в столбик. Тогда самым простым решением будет сеточный менеджер компоновки **GridLayout** с одним столбцом и тремя строками. Чтобы выбрать менеджер компоновки для нашей панели, нажимаем на правую кнопку мыши на панели и находим пункт меню **SetLayout**. Можете посмотреть, какие еще способы компоновки вам предлагает NetBeans. По умолчанию используется **Free Design** — он выделен жирным шрифтом. Нам надо выбрать **GridLayout**. Зададим теперь параметры сетки — для этого нужно в окне **Inspector** найти элемент **GridLayout** и, нажав на нем правой кнопкой, выбрать пункт **Properties**. В появившемся окне нас интересуют в первую очередь свойства **Columns** и **Rows**, которые мы должны установить соответственно в 1 и 3. Другие два свойства — **Horizontal Gap** и **Vertical Gap** определяют расстояние между элементами управления по горизонтали и вертикали. Установим для них значения 10.

Теперь поместим на форму панели сами поля для ввода фамилии, имени и отчества. Справа от формы панели находится кнопка **Palette**. Если вы подведете туда мышку, то увидите панель, предоставляющую всё многообразие доступных вам графических компонентов. На самом деле вы без труда можете добавить на эту панель внешние компоненты, в том числе и разработанные вами. Пока нам нужно найти поле ввода **Text Field**, оно расположено в разделе **Swing Controls**.

Выберем его и щелкнем мышкой на проектируемой нами панели — на ней появится новый элемент (рис. 5).

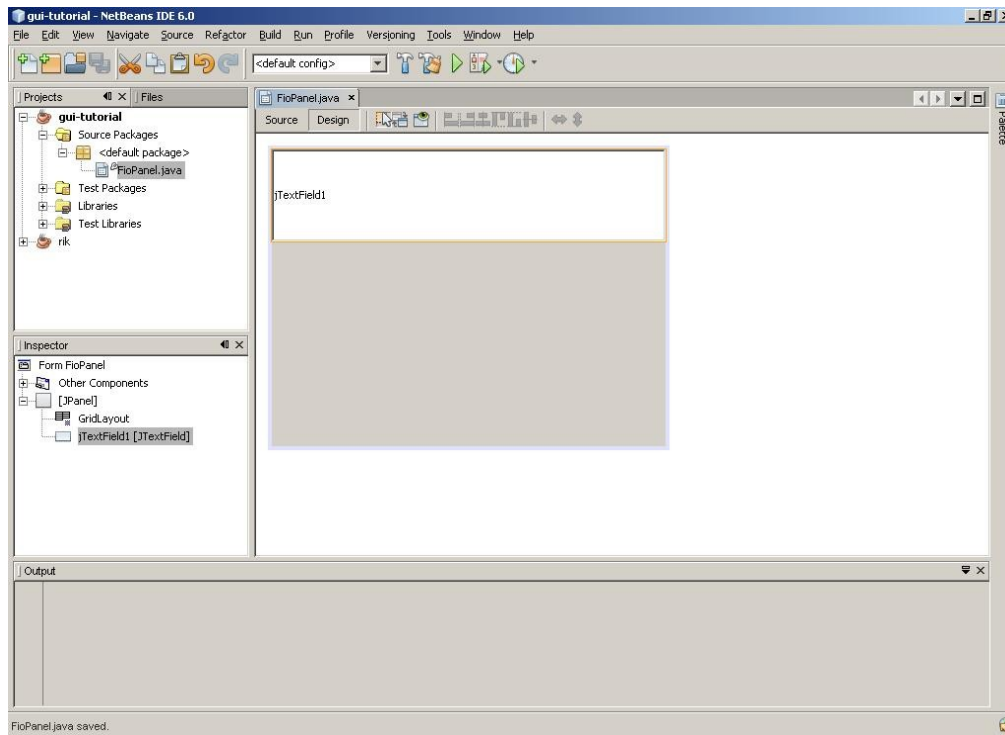


Рисунок 5. Создание поля ввода

Это нехитрое действие придется повторить еще два раза, если вы, конечно, не предпочтете сэкономить немного времени, воспользовавшись испытанной технологией Copy — Paste. Для этого выделив мышью только что созданное поле ввода, нажмите Ctrl+C и два раза Ctrl+V. Вы увидите, что появилась еще пара полей ввода. Кстати, NetBeans поддерживает не только одиночное копирование/вырезание элементов, но и групповые операции, в том числе групповое изменение свойств. Чтобы убедиться в этом, выделите мышкой все три поля, удерживая при этом нажатой кнопку Shift. Теперь нажмите правую кнопку мыши и выберите пункт Properties. Найдите свойство text и сотрите содержащуюся в нем строку jTextField1. Закройте окно свойств и вы увидите, что у всех трех полей ввода исчез содержащийся в них текст.

Для того чтобы панель выглядела более естественно, зацепившись мышкой за ее правый нижний угол, уменьшим ее размер (рис. 6).

Назовем также поля ввода более осмысленными именами с помощью того же окна свойств (свойство Variable Name во вкладке Code): surnameField, firstnameField, patronymicField.

Теперь, чтобы сделать нашу панель полноценным JavaBean-компонентом, придется написать немного кода.

Речь идет о создании get- и set- методов для чтения и записи фамилии, имени и отчества: getSurname(), setSurname(String surname), getFirstname(), setFirstname(String firstname), getPatronymic(), setPatronymic(String patronymic), getFullname().

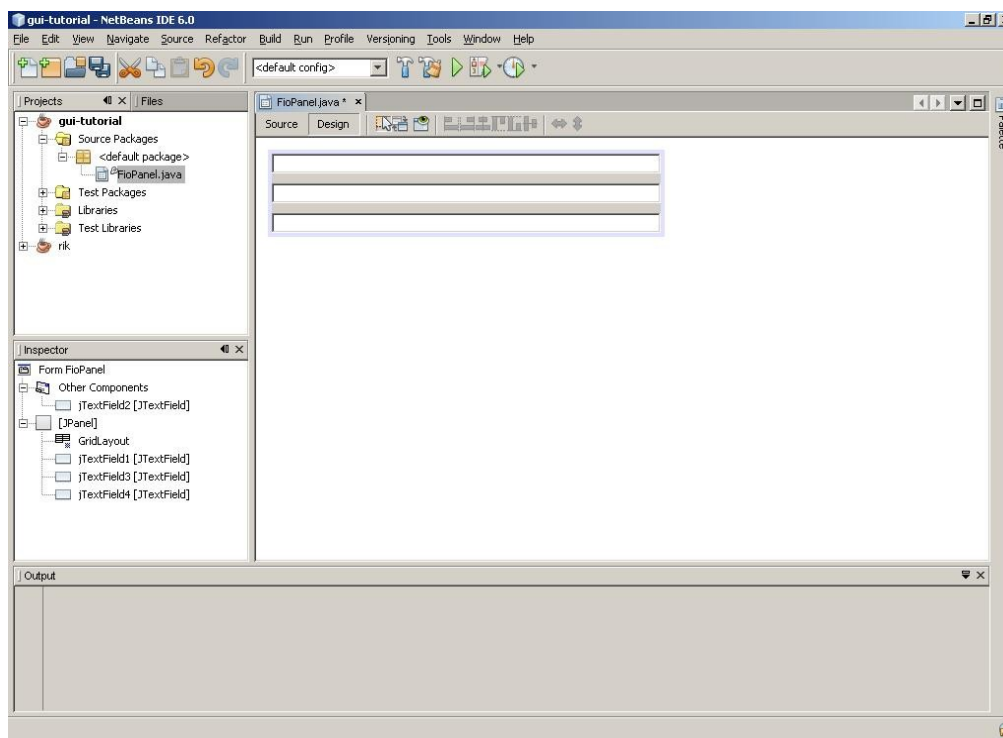


Рисунок 6. Измененная панель

Эти методы мы набираем, перейдя в редактор кода:

```
public String getSurname() {
    return surnameField.getText();
}

public void setSurname(String surname) {
    surnameField.setText(surname);
}

public String getFirstname() {
    return firstnameField.getText();
}

public void setFirstname(String firstname) {
    firstnameField.setText(firstname);
}

public String getPatronymic() {
    return patronymicField.getText();
}

public void setPatronymic(String patronymic) {
    patronymicField.setText(patronymic);
}

public String getFullname() {
    return surnameField.getText() + " " +
        firstnameField.getText() + " " +
        patronymicField.getText();
}
```

Итак, наша панель для ввода фамилии, имени и отчества готова, и мы можем использовать ее в наших приложениях. Для примера создадим диалог на ее основе. В окне Projects нажмите правой кнопкой на имени проекта и выберите New ► JDialog Form. Назовем новый диалог FioDialog. Поместить панель

FioPanel на диалог очень просто, но сначала нужно скомпилировать класс панели. Простейший способ сделать это — нажать F9, выделив папку Source Packages в окне Projects. После этого просто перетащите мышью панель из окна Projects на пространство диалога. Можно с помощью мышки установить нужный размер панели и самого диалога (рис. 7).

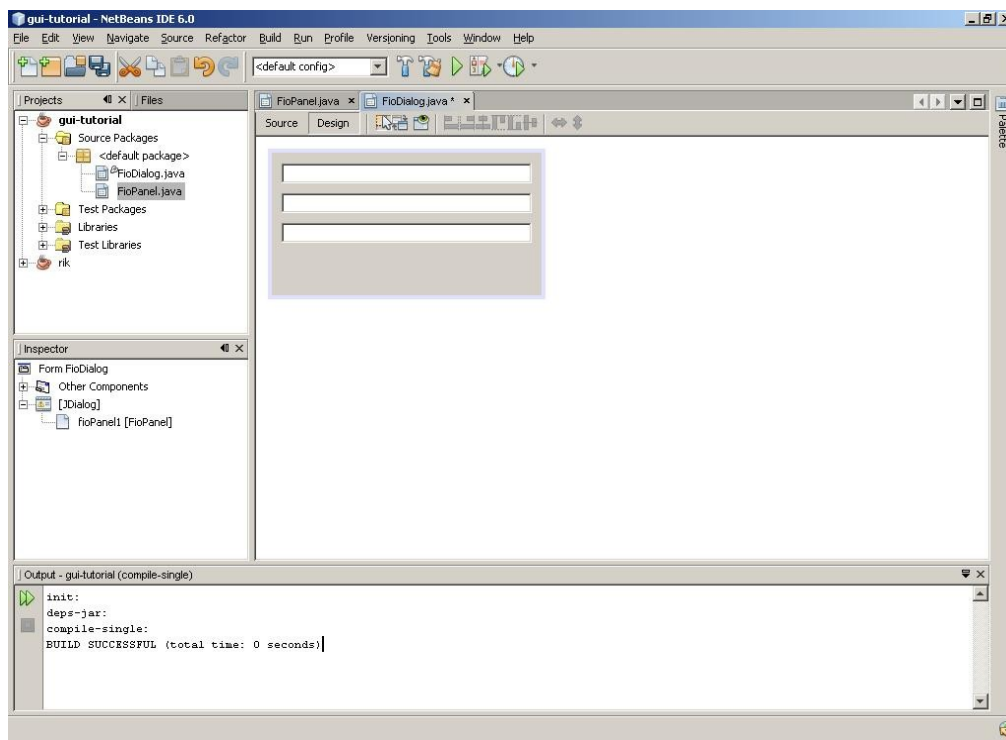


Рисунок 7. Размеры изменены

Нажмите теперь правой кнопкой на вставленной в диалог панели и выберите пункт Properties. Если вы внимательно изучите набор свойств панели, то найдете среди них свойства surname, firstname, patronymic и fullname. Тех, кто интересуется на основе чего IDE создала эти свойства, и тех, кто равнодушен к тонкостям и подробностям, отсылаем к спецификации JavaBeans — <http://java.sun.com/javase/technologies/desktop/javabeans/index.jsp>. Пока обратите внимание на то, что первые три из этих свойств мы можем изменять, и результаты наших изменений немедленно отразятся на состоянии панели. Установите, например, для свойства surname значение «Иванов» (рис. 8). Что касается свойства fullname, то его значение мы изменить не можем — это свойство только для чтения. Что и не удивительно, учитывая, что для этого свойства мы создали только get-метод getFullname(). (За подробностями отсылаю опять-таки к спецификации JavaBeans).

Добавим в диалог традиционную кнопку ОК. Чтобы изменить надпись на кнопке делаем один щелчок на выделенной кнопке и вводим: «ОК». Двойной щелчок на кнопке автоматически создаст обработчик нажатия на кнопку и перебросит нас в редактор кода для написания обработчика.

Вставим в обработчик следующий код:

```
JOptionPane.showMessageDialog(this, "Пользователь NetBeans " +  
this.fioPanel1.getFullname());
```


Чтобы диалог не выглядел так сиротливо, зададим для него заголовок окна, то есть установим свойство `title` в значение «ФИО». Обратите внимание на то, что все измененные вами свойства NetBeans заботливо выделяет жирным шрифтом в окне `Properties`. В отличие от всех остальных свойств, значения которых оставлены по умолчанию.

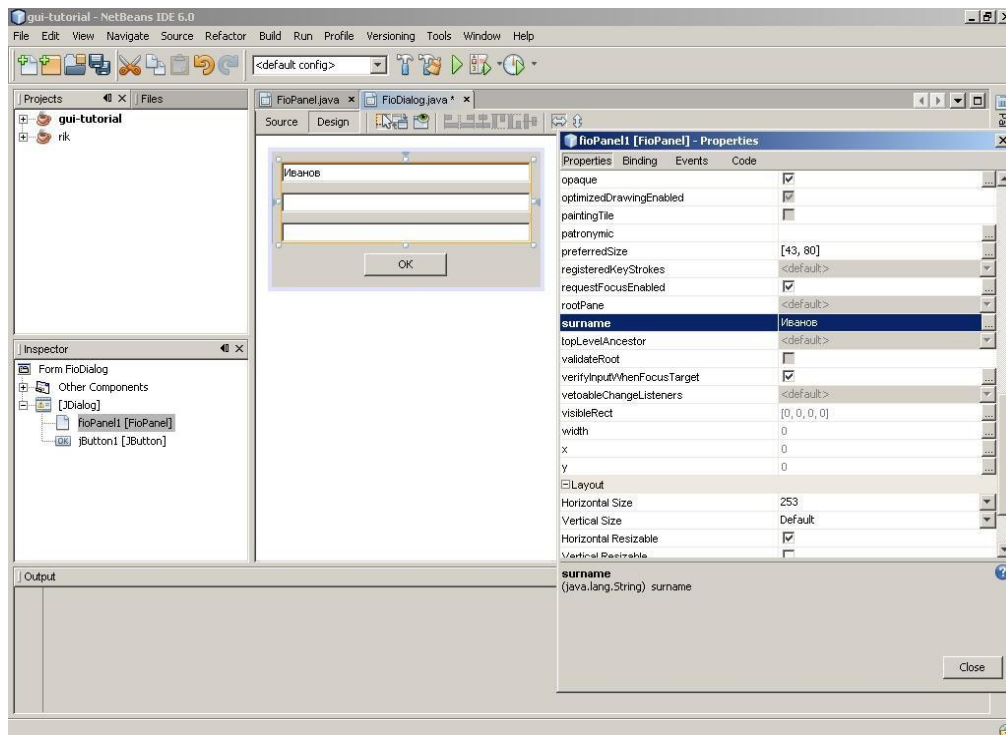


Рисунок 8. Установка значений свойств

Теперь наконец-то мы можем получить удовольствие от результата проделанной работы. Запустите диалог на выполнение, нажав `Shift+F6` (рис. 9).

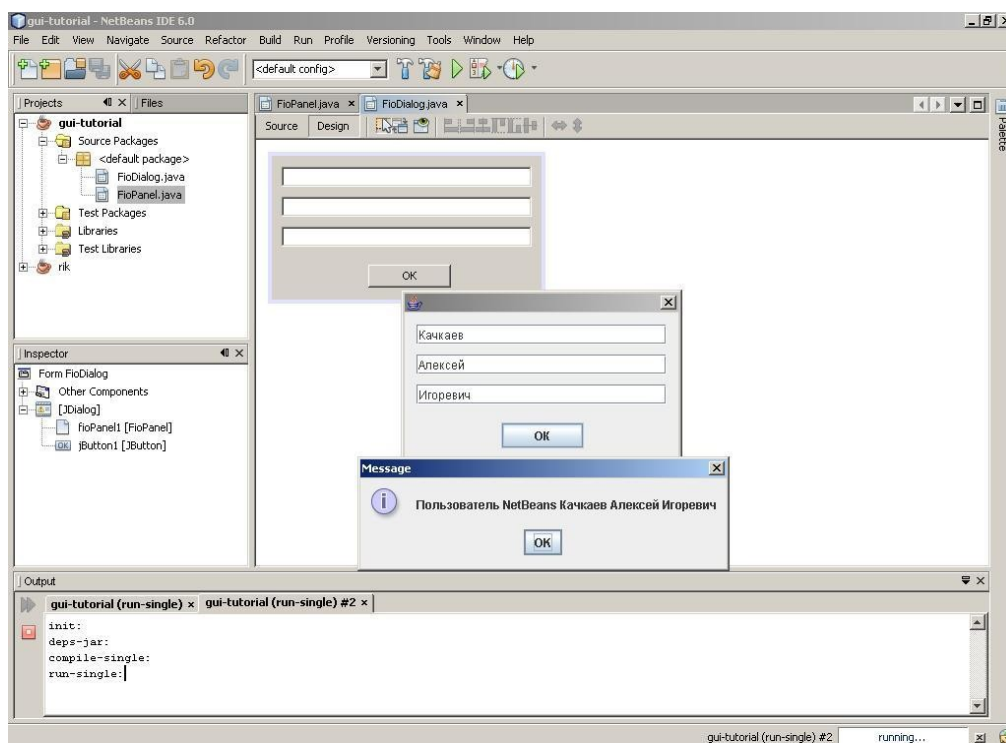


Рисунок 9. Результаты работы программы

Таким образом, нам удалось, используя NetBeans 6.0, достаточно просто создать графический интерфейс для java-приложения.

А сколько еще умеет делать NetBeans!!!

Алексей Качкаев

Центр новых информационных технологий
Рязанского государственного радиотехнического университета
апрель 2008 г.