

УДК 004.9

doi 10.18101/978-5-9793-0803-6-182-185

СОЗДАНИЕ ПРИЛОЖЕНИЙ ДЛЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ С ПОМОЩЬЮ JAVA

© *Шадрина Наталья Николаевна*, старший преподаватель кафедры вычислительной техники и информатики Бурятского государственного университета Россия, г. Улан-Удэ
E-mail: shadrinann8@yandex.ru

В статье рассматривается технология создания приложений на языке высокого уровня JavaSwing с оболочкой NetBeans на примере решения одной инженерной задачи. Небольшие компактные приложения, позволяющие решать проблему расчетов не только в курсовых, дипломных, выпускных квалификационных работах, но и повседневной жизни, в быту, сегодня особенно популярны. Программы, позволяющие рассчитывать, к примеру, количество калорий, вес багажа и т. д., могут быть установлены на смартфонах, планшетах и других гаджетах. Ориентация студентов на создание прикладных программ, которые не только будут являться частью учебных заданий, но также могут быть предложены потребителю как самостоятельные продукты, сегодня актуальна. В статье рассмотрен алгоритм создания приложения для простого инженерного расчета, включающий в себя этапы создания проекта, заполнения диалоговой формы, прописывания кода и запуска готового приложения.

Ключевые слова: JavaSwing, NetBeans, инженерная задача, технология создания приложений.

CREATING APPLICATIONS FOR ENGINEERING CALCULATIONS USING JAVA

Natalya N. Shadrina, Senior Lecturer at the Department of Computer Sciences and Informatics, Buryat State University Russia, Ulan-Ude

The article discusses the technique of creating applications based on the high-level JavaSwing language with a shell NetBeans on the example of the solution of one engineering problem. Some small, compact applications that support the calculation problem to be solved are especially popular today not only in the students' course works, Diploma papers or graduation qualification works but also in everyday life. The programs that allow calculating, for example, the amount of calories, the weight of luggage and so on can be installed in smartphones, computer tablets and other gadgets. Orientation of students to create application software that will not only be the part of the class assignment, but may also be offered to the consumer as an independent product is very relevant today. The article describes the algorithm for creating an application for a simple engineering calculation, containing the timeline of project creation, the dialogue form fill, the code prescription and the startup of the finished application.

Keywords: JavaSwing, NetBeans, engineering task, the technology to create applications.

Инженерные задачи могут возникать как на протяжении всего периода обучения специальности, так и при написании выпускной квалификационной работы.

Рассмотрим, например, задачу для расчета оперативности телекоммуникационных вычислительных сетей (ТВС). Если надежность и качество сети идеальны, то оперативность ТВС оценивается из теории массового обслуживания по формуле:

$$Q = P_0 + P_1,$$

где P_0 — вероятность того, что сеть свободна и ожидающих нет; P_1 — вероятность того, что сеть занята, но ожидающих нет. Вероятности состояний сети P_0 и P_1 рассчитываются из теории массового обслуживания по формулам:

$$P_0 = \frac{1}{\sum_{k=0}^{N-1} \frac{N!}{(N-k)!} * y^k},$$
$$P_1 = \frac{N * y}{\sum_{k=0}^{N-1} \frac{N!}{(N-k)!} * y^k}$$

где N — число рабочих станций сети; y — нагрузка в сети; k — последовательность чисел $k = 0, 1, 2, \dots, N-1$.

Нагрузка в сети

$$y = \lambda * T_{\text{пер}},$$

где λ — интенсивность вызовов в сети (трафик).

Для решения данной задачи создадим приложение, позволяющее вводить число рабочих станций сети, интенсивность вызовов в сети и время «чистой» передачи данных. В результате работы приложения пользователь получает значение оперативности ТВС, величину которого может использовать в дальнейших расчетах или для качественного анализа.

Запускаем Java. Выбираем «Приложение Java». Далее вводим имя проекта, например, «TVS_efficiency» (имя может быть любым, записано латинскими буквами, не начинается с цифры). Главный класс не создаем, убираем флажок «Создать главный класс». Нажимаем кнопку «Готово».

В окне проектов находим свой проект, выделяем его название, правой кнопкой мыши вызываем контекстное меню, выбираем «Новый», «Форма JFrame». Задаем имя класса, например, «JFrame_TV» (имя может быть любым, записано латинскими буквами, не начинается с цифры). Нажимаем кнопку «Готово».

В правой верхней части окна расположено меню «Контейнеры Swing». Мышью выбираем и переносим на форму элемент «Панель». В правой нижней части окна расположено меню «JPanel Свойства». Выбираем в нем свойство «border», нажимаем кнопку «...».

В списке меню выбираем пункт «Рамка с надписью». Вводим заголовок «Расчет оперативности ТВС». Помещаем на панель компонент JLabel (метка), в ее свойство «text» заносим «Интенсивность вызовов в сети». Аналогично формируем метки «Время «чистой» передачи данных» и «Число рабочих станций сети».

Добавляем на форму три текстовых поля JTextField1, JTextField2, JTextField3 для ввода значений. Свойство «text» для этих компонентов нужно очистить.

Помещаем на форму кнопку JButton, задаем ее свойству «text» значение «Вычислить». Под ней располагаем надпись «Оперативность ТВС» и поле для вывода. Можно запустить проект, нажав на зеленую стрелку на панели инструментов или выбрав пункт главного меню «Выполнить». Получим готовую форму.

Выделяем кнопку «Вычислить» и переходим в меню «События» в правом нижнем углу экрана. Выбираем событие «action Performed». Попадаем в раздел «Источник», где находится автоматически сгенерированный код всей нашей формы.

Метод **“private void jButton1ActionPerformed”** уже имеет прописанную сигнатуру, в которую необходимо добавить вычисления.

Вместо закомментированного приглашения Java

// TODO add your handling code here:

вставляем код для обработки входных данных, вычислений и вывода ответа

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    //Объявляем переменные вещественного типа
```

```
    float lambda,N,T,P0,P1,Q;
```

```
    // Из окон ввода получаем значения, преобразовываем их из строкового
```

```
    типа в вещественный и присваиваем объявленным переменным
```

```
    lambda=Float.parseFloat(jTextField1.getText());
```

```
    N=Float.parseFloat(jTextField3.getText());
```

```
    T=Float.parseFloat(jTextField4.getText());
```

```
    //Вводим вспомогательные переменные и
```

```
    //осуществляем необходимые расчеты по формулам
```

```
    float pN=1,p2,p3=1,s=0,y;
```

```
    double p,p1;
```

```
    y=lambda*T;
```

```
    for (int i = 1; i <= N; i++) { pN*=i; }
```

```
    for (int k = 0; k < N; k++) { p2=1;
```

```
    for (int i = 1; i <= N-k; i++) { p2*=i; } s=s+p3/p2; p3*=y; }
```

```
    p=(s/pN)*(1+N*y);
```

```
    //Преобразуем ответ из вещественного типа
```

```
    //в строковый и заносим в окно вывода
```

```
    jTextField5.setText(String.valueOf(p)); }
```

В итоге получаем приложение, которое при вводе исходных данных производит необходимые инженерные расчеты и выдает результат.

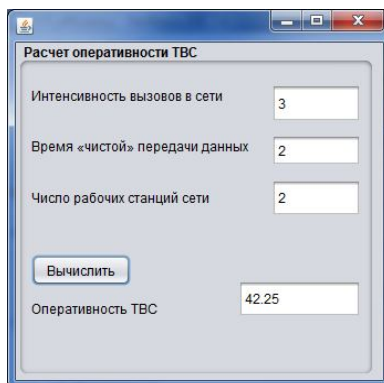


Рис. 1. Запуск готового приложения

Примечания

1. При создании приложений часто возникают ситуации, когда внесенные в окно ввода данные вызывают ошибку работы программы: переполнение, деление на нуль, отсутствие числа. Для корректной работы программы в случае подобных ошибок можно включить в код обработку прерываний.

2. На форму приложения также можно поместить метку, содержащую текст задания, схему или рисунок.

3. Если выполнить программу, используя пункт меню «Выполнить», «Очистить и создать проект», то в результате получим файл с расширением .jar, который будет запускать приложение без загрузки Java.

Литература

1. Герберт Шилдт Полный справочник по Java™. 7-е изд. М.: И Д. Вильямс; СПб.; Киев, 2007. 1025 с.
2. Седжвик Р., Уэйн К. Алгоритмы на Java: пер. с англ. 4-е изд. М.: И. Д. Вильямс, 2013. 848 с.