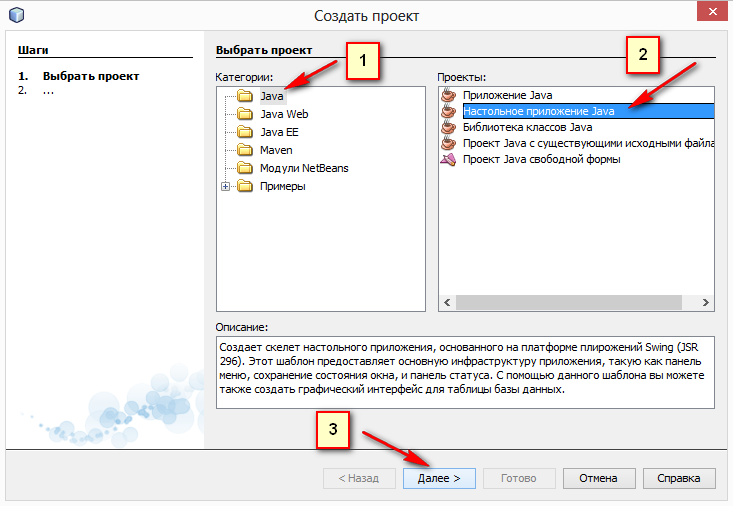
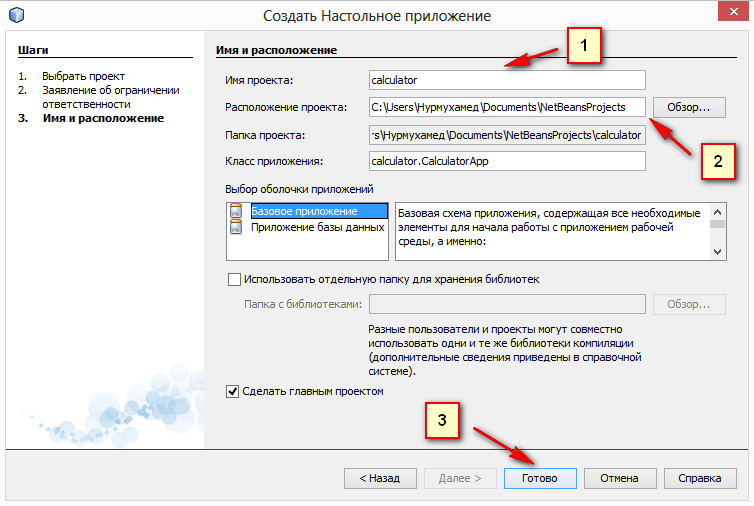
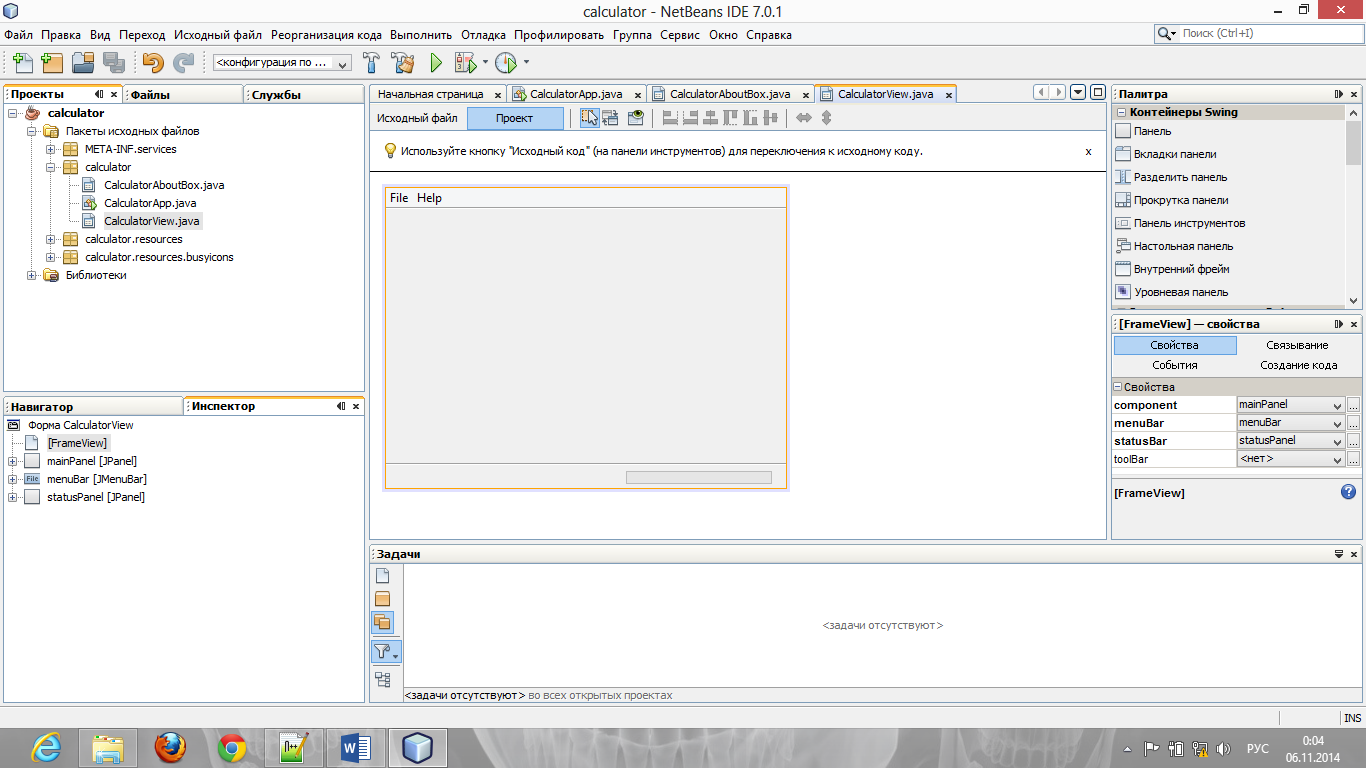
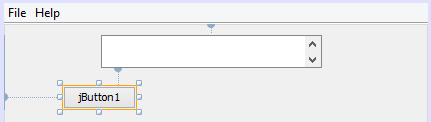
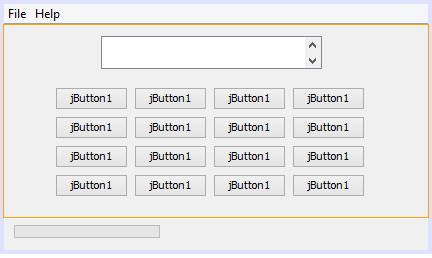
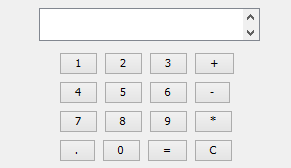
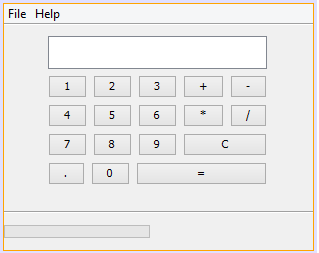
Создание приложения калькулятор

**Цель:** ознакомиться с разработкой приложений с GUI интерфейсом на языке программирование Java в среде разработки NetBeans IDE 7.0.1

1. Запускаем NetBeans.
2. Создаём новый проект. Для этого кликаем по пункту меню «*Файл»* затем по вкладке «*Создать проект»* либо сочетание клавиш *Ctrl+Shift+N.*
3. **В открывшемся меню выбираем тип создоваемого проекта так как указанно ниже
4. Появляется следующее окно с заявлением, жмём далее.
5. Затем даём нашему проекту имя и указываем директорию где он будет сохранён.
6. Открывается рабочая область. Здесь всё стандартно в середине окно редактора, которое можно переключать между режимами *Исходный файл*, где отображается исходный коди режим *Проект* отображающий графический вид проекта. Справа палитра компонентов и свойства слева дерево проекта и окно навигации. Можно создать удобную для себя среду перемещая окна местами изменяя их размер и т. д.
7. Создаем внешний вид будущего калькулятора, для этого перетаскиваем элементы управления *Участок текста* (Text area) и элемент *Кнопка* (Button) на форму из окна *Палитра.*
8. Добавляем ещё кнопок, сколько зависит от функциональности калькулятора в моём приложении их 17 это цифры и след. знаки « **/** », « \* », « + », « - », « = », « . », « С ». Процесс можно ускорить используя *copy paste*.
9. Редактируем текст на кнопках это можно сделать в окне свойств, кликнув правое кнопкой мыши на эл. в контекстном меню выбрав *Редактировать текст* или нажав клавишу F2.



1. Можно изменить размер кнопок их расположение, размер формы так как вам будет более удобно. В свойствах можно изменить цвет фона, кнопок формы
2. Убираем полосу прокрутки с эл. *Text area*. Кликаем по полосе прокрутки в окне *свойства* изменяем значения параметра *horizontalScrollBarPolicy* на *NEVER* тоже самое делаем с параметром *verticalScrollBarPolicy.*
3. Затем прописываем действия при нажатии кнопок кликаем по кнопке с надписью «1» правой кнопкой мыши в контекстном меню последовательно выбираем События→Action→actionPerformed окно редактора переходит в режим *Исходный файл* курсив встаёт в теле функции описывающей событие на нажатие кнопки, которую вы выбрали, в теле этой функции прописываем след. код

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

*// TODO add your handling code here:*

*jTextArea1.append("1");*

*}*

*где jTextArea*1 имя текстового поля append("1") ф-ия передающее значение 1 эл. jTextArea1. Проделаем данную операцию с каждой кнопкой с цифрами изменяя только значение, передаваемое в jTextArea1. Для кнопки «2»

*private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*jTextArea1.append("2");*

*}*

1. Пишем событие на кнопку « . »

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

*// TODO add your handling code here:*

*jTextArea1.append(".");*

*}*

1. Объявляем переменные

*double input\_num1, input\_num2, result;*

*String s;*

*int addc=0, subc=0, multic=0, divc;*

1. В коде пишем след. ф-ию

*public double reader(){*

*double number;*

*String s;*

*s=jTextArea1.getText();*

*number=Double.valueOf(s);*

*return number;*

*}*

1. Задаём действия на кнопку «+»

*private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num1=reader();*

*jTextArea1.setText("");*

*addc=1;*

*subc=0;*

*multic=0;*

*divc=0;*

*}*

1. Задаём действия на кнопку « - »

*private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num1=reader();*

*jTextArea1.setText("");*

*addc=0;*

*subc=1;*

*multic=0;*

*divc=0; }*

1. Задаём действия на кнопку «\*»

*private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num1=reader();*

*jTextArea1.setText("");*

*addc=0;*

*subc=0;*

*multic=1;*

*divc=0;*

*}*

1. Задаём действия на кнопку «/»

*private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num1=reader();*

*jTextArea1.setText("");*

*addc=0;*

*subc=0;*

*multic=0;*

*divc=1;*

*}*

1. Задаём действия на кнопку «С»

*private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num1=0.0;*

*input\_num2=0.0;*

*jTextArea1.setText("");*

*}*

1. Задаём действия на кнопку « = »

*private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {*

*// TODO add your handling code here:*

*input\_num2=reader();*

*if(addc>0)*

*{*

*result=input\_num1+input\_num2;*

*jTextArea1.setText(Double.toString(result));*

*}*

*if(subc>0)*

*{*

*result=input\_num1-input\_num2;*

*jTextArea1.setText(Double.toString(result));*

*}*

*if(multic>0)*

*{*

*result=input\_num1\*input\_num2;*

*jTextArea1.setText(Double.toString(result));*

*}*

*if(divc>0)*

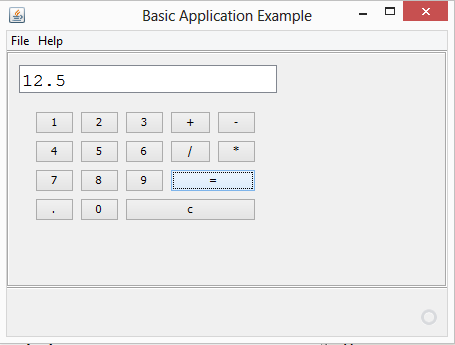
*{*

*result=input\_num1/input\_num2;*

*jTextArea1.setText(Double.toString(result));*

*}*

*}*

1. Пока функции запрещающей ввод букв и не нужных символов в j*TextArea1* нет чтобы обеспечить отказоустойчивость программы в свойствах данного эл. убираем флажок с параметра *Editable* тем самым запретив ввод данных поле с клавиатуры.
2. Нажатием F6 запускаем проект, если всё сделано правильно, то запустится приложение как на картинке ниже