

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СЕРВИСА»
(ФГБОУ ВО «ПВГУС»)

Кафедра «Прикладная информатика в экономике»

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплине «Язык программирования Java и средства NetBeans»

Одобрено
Учебно-методическим
Советом университета

Составитель **Малышева Е. Ю.**

Тольятти 2016

УДК 002.52/54(075.8)
ББК -018*32.973я7
Л 12

Рецензент
к.т.н., доц. **Бобровский С. М.**

Л 12 Лабораторный практикум по дисциплине «Язык програм-
мирования Java и средства NetBeans» / сост. Е. Ю. Малышева. –
Тольятти : Изд-во ПВГУС, 2016. – 48 с.

Лабораторный практикум по дисциплине «Язык программирования Java и
средства NetBeans» разработан в соответствии с требованиями ФГОС ВО

УДК 002.52/54(075.8)
ББК -018*32.973я7

© Малышева Е. Ю., составление, 2016
© Поволжский государственный
университет сервиса, 2016

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	4
2. ЛАБОРАТОРНЫЕ РАБОТЫ	5
ЛАБОРАТОРНАЯ РАБОТА № 1. ПРОСТОЕ ПРИЛОЖЕНИЕ В СРЕДЕ NETBEANS	5
ЛАБОРАТОРНАЯ РАБОТА № 2. СОЗДАНИЕ КЛАССОВ И ОБЪЕКТОВ В JAVA	11
ЛАБОРАТОРНАЯ РАБОТА № 3. НАСЛЕДОВАНИЕ КЛАССОВ	16
ЛАБОРАТОРНАЯ РАБОТА № 4. АБСТРАКТНЫЕ КЛАССЫ	24
ЛАБОРАТОРНАЯ РАБОТА № 5. ИНТЕРФЕЙСЫ.	31
3. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ, ОФОРМЛЕНИЯ ОТЧЕТОВ ПО РАБОТЕ.....	39
4. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	39
4.1. ОСНОВНАЯ ЛИТЕРАТУРА	39
4.2. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	39
ПРИЛОЖЕНИЕ 1	41

2. ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1. Простое приложение в среде NetBeans

Цель работы: изучить принципы создания консольных и оконных приложений в NetBeans.

Задание: создать консольное приложение в NetBeans для вычисления простого арифметического выражения в соответствии с вариантом.

Указания к работе:

Построение программ, основанных на объектно-ориентированном программировании (ООП), принципиально отличается от более ранней методики процедурного программирования, в которой основой построения программы служили подпрограммы. ООП - это методология программирования, опирающаяся на три базовых принципа: инкапсуляцию, наследование, полиморфизм. Язык Java является объектно-ориентированным и в полном объеме использует эти принципы.

Программы, написанные в соответствии с принципами процедурного программирования, состоят из набора подпрограмм, причем для решения конкретной задачи программист явно указывает на каждом шагу, что делать и как делать. Эти программы практически полностью (процентов на девяносто) состоят из решения конкретных задач.

Программы, написанные в соответствии с принципами ООП, пишутся совершенно иначе. В них основное время занимает продумывание и описание того, как устроены классы. Код с описанием классов предназначен для многократного использования без внесения каких-либо изменений. И только небольшая часть времени посвящается решению конкретной задачи – написанию такого кода с использованием классов, который в других задачах такого типа не применить. Именно благодаря такому подходу объектное программирование приобрело огромную популярность – при необходимости решения сходных задач можно использовать уже готовый код, модифицировав только ту часть программы, которая относится к решению конкретной задачи. Даже самая простая задача создается в виде класса.

Современное программное обеспечение построено по модульному (блочному) принципу. Программы давно перестали состоять из одного файла. Поэтому вместо слова "программа" лучше употреблять слово "проект".

Идеология Java подразумевает работу в компьютерных сетях и возможность подгрузки в необходимый момент через сеть требуемых классов и ресурсов, в которых нуждается программа, и которые не были до того загружены. Для обеспечения такого рода работы приложения Java разрабатываются и распространяются в виде большого числа независимых классов. Однако такой способ разработки приводит к чрезвычайно высокой фрагментации про-

граммы. Даже небольшие учебные проекты часто состоят из десятков классов, а реальные проекты – из сотен. При этом каждому общедоступному (public) классу соответствует свой файл, имеющий то же имя. Для того чтобы справиться с таким обилием файлов, в Java предусмотрено специальное средство группировки классов, называемое пакетом (package). Пакеты обеспечивают независимые пространства имен (namespaces), а также ограничение доступа к классам.

Классы всегда задаются в каком-либо пакете. Пакеты могут быть вложенными с произвольным уровнем вложения (ограничивается только операционной системой). Каждому пакету соответствует папка с исходными кодами соответствующих классов, при этом пакету, вложенному в другой, соответствует папка, вложенная в другую.

Для того чтобы поместить класс в пакет, требуется продекларировать имя пакета в начале файла, в котором объявлен класс, в виде

```
package имя_пакета;
```

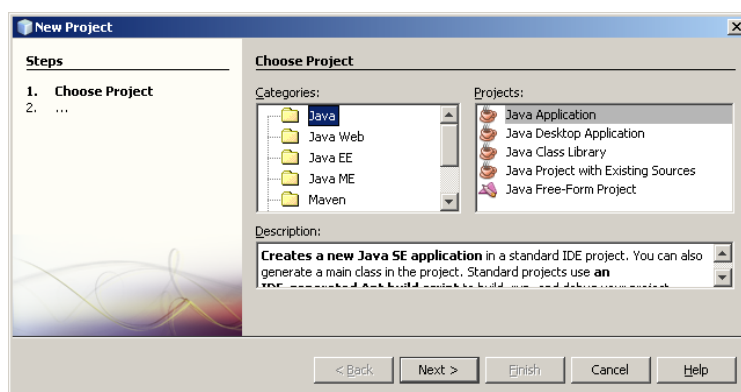
Кроме того, необходимо поместить исходный код класса в папку, соответствующую пакету. Если декларация имени пакета отсутствует, считается, что класс принадлежит пакету с именем default. При создании проекта в среде NetBeans помещение класса в пакет происходит автоматически.

Описание и примеры управляющих конструкций даны в приложении 1.

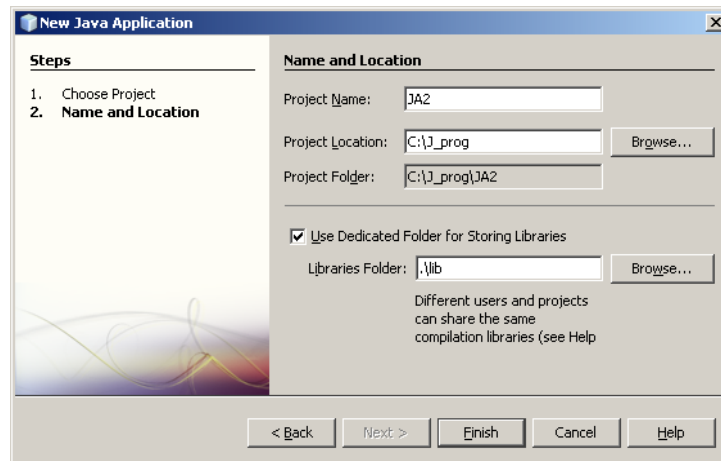
Ход работы:

Этап создания консольного приложения.

Создаем новое приложение в NetBeans: File > New Project > Java Application



Даем проекту имя и выбираем папку для сохранения проекта



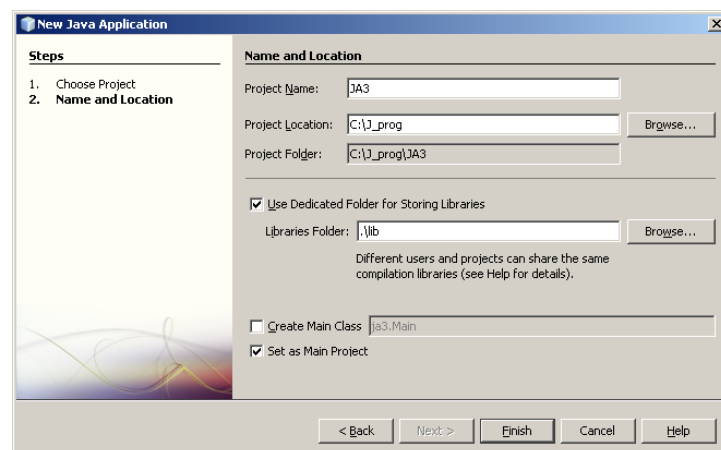
Непосредственно в методе main пишем команды для вычисления выражения и вывода результата:

```
public class Main {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int x=2;
        int y=5;
        System.out.println("4*x+7*y="+ (4*x+7*y));    }
}
```

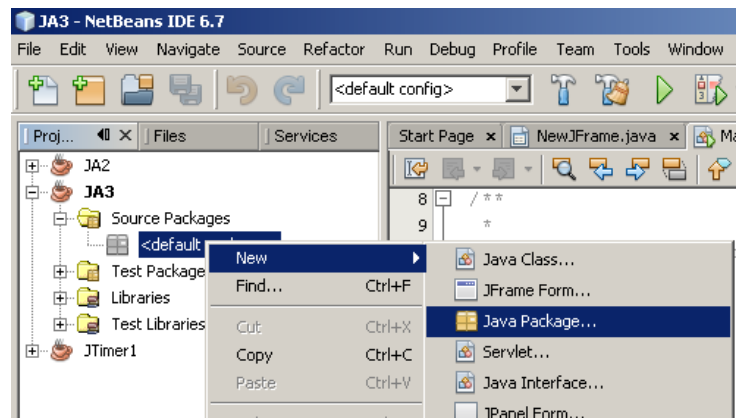
Запускаем приложение (команда Run или кнопка с зеленым треугольником)

Этап создания оконного приложения.

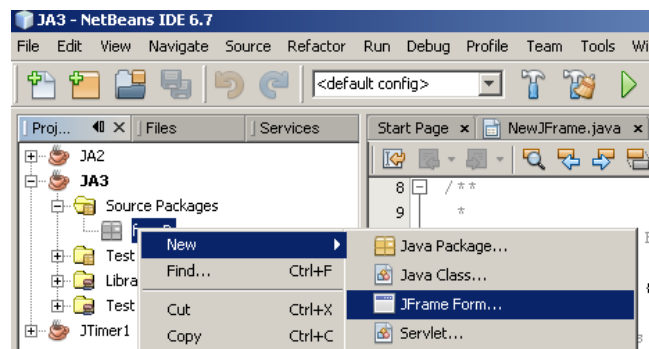
Создаем новое приложение без процедуры main



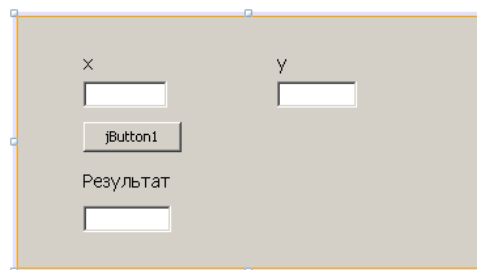
Используя контекстное меню, создаем пакет formP



Используя контекстное меню, создаем класс формы



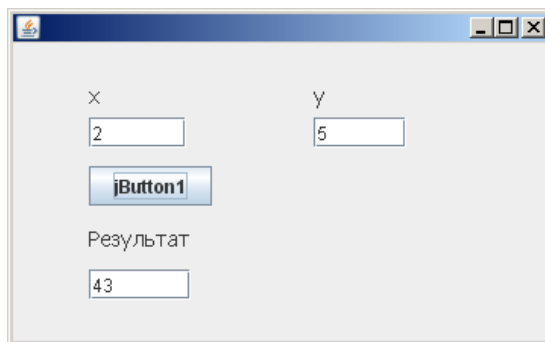
Добавляем на форму панель элементы текста, поля ввода, поле вывода, кнопку для расчета выражения.



Непосредственно в классе обработки события нажатия кнопки пишем команды для вычисления выражения и вывода результата.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int x=Integer.valueOf(jTextField1.getText());
    int y=Integer.valueOf(jTextField2.getText());
    jTextField3.setText(Integer.toString(4*x+7*y));
}
```

Запускаем приложение.



ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

Этап создания консольного приложения для вычисления сложного арифметического выражения.

Создаем новое консольное приложение

Непосредственно в методе `main` пишем команды для вычисления выражения и вывода результата. Синтаксис условных операторов и операторов цикла аналогичен языку C++.

Пример условного оператора:

```
if (a<b) {
    a=a+1;
}
else {
    b=b+1;
}
```

Пример оператора цикла: вычисление суммы последовательно идущих чисел.

Напишем цикл, в котором производится суммирование всех чисел от 1 до 100. Результат будем хранить в переменной `result`.

```
int result=0;
for(int i=1; i<=100; i++){
    result=result+i;
}
```

Варианты арифметических выражений

<i>Вариант</i>	<i>x</i>	<i>y</i>	<i>Выражение</i>
1	2	6	$2x+3y$
2	3	7	$3x-2y$
3	4	8	$4x+3y$
4	2	9	$7x-4y$
5	4	3	$1x+5y$

6	5	5	$4x-6y$
7	4	9	$9x+3y$
8	6	1	$2x-8y$
9	7	4	$4x*3y$
10	5	8	$5x*2y$

Варианты сложных выражений для дополнительного задания

<i>Вариант</i>	<i>x</i>	<i>Выражение</i>
1	2	Если $x < 2$, то $y = \text{сумме чисел от 2 до 13}$, иначе $y = 45$
2	3	Если $x < 5$, то $y = \text{сумме чисел от 4 до 11}$, иначе $y = 45$
3	4	Если $x < 6$, то $y = \text{сумме чисел от 7 до 17}$, иначе $y = 45$
4	2	Если $x < 3$, то $y = \text{сумме чисел от 3 до 12}$, иначе $y = 45$
5	4	Если $x < 7$, то $y = \text{сумме чисел от 8 до 10}$, иначе $y = 45$
6	5	Если $x < 11$, то $y = \text{сумме чисел от 12 до 17}$, иначе $y = 45$
7	4	Если $x < 21$, то $y = \text{произведению чисел от 11 до 13}$, иначе $y = 45$
8	6	Если $x < 13$, то $y = \text{произведению чисел от 1 до 3}$, иначе $y = 45$
9	7	Если $x < 2$, то $y = \text{произведению чисел от 4 до 8}$, иначе $y = 45$
10	5	Если $x < 8$, то $y = \text{сумме чисел от 5 до 13}$, иначе $y = 45$

Вопросы для самоконтроля

1. Что такое среда разработки NetBeans?
2. Что такое консольное приложение?
3. Что такое оконное приложение?
4. Какие типы данных в Java вы знаете?
5. Для чего используется и как обозначается блок в Java?
6. Для чего применяется оператор if?
7. Как используется оператор while?
8. Как используется оператор do-while?
9. Как используется оператор for?

Литература: 1,2,3,5.

Учебное издание

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплине «**Язык программирования Java и средства NetBeans**»

Составитель

Малышева Елена Юрьевна

Издается в авторской редакции.

Подписано в печать с электронного оригинал-макета 30.06.2016.

Бумага офсетная. Печать трафаретная. Усл. печ. л. 3,0.

Тираж 500 экз. Заказ 78/01.

Издательско-полиграфический центр

Поволжского государственного университета сервиса.

445677, г. Тольятти, ул. Гагарина, 4.

rio@tolgas.ru, тел. (8482) 222-650.

Электронную версию этого издания

вы можете найти на сайте ЭБС ПВГУС <http://elib.tolgas.ru/>.