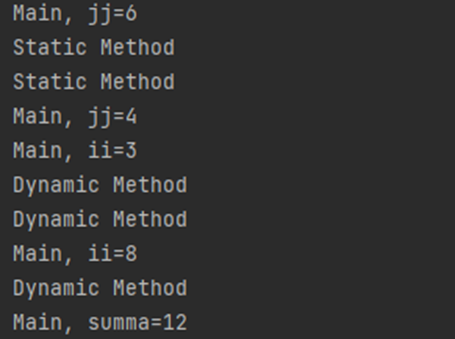
**Лабораторная работа №4**

**Основы программирования на Java: классы, объекты, пакеты**

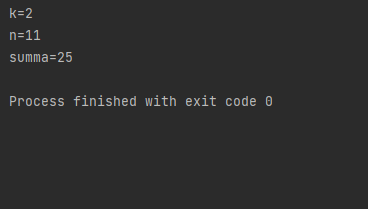
**Цель работы:** Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Используя примеры программ познакомиться с основными приемами в программировании на Java. Приобрести навыки в использовании системы помощи для поиска нужной информации по различным классам Java.

**Приложение TestElements:**



Это приложение демонстрирует работу статического и динамического класса путем ввода и визуального вывода значений. В этой программе сначала определяется класс, включающий статические и динамические методы, а затем его методы используются в функции main().

**Приложение TestModificators:**



Это программа демонстрирует функции public, protected, private типов классов. Использование этих методов помогает в защищенности кода, т.к. protected ограничивает доступ, а private полностью его блокирует.

**Ответы на КР вопросы:**

1. Язык Java позволяет создавать несколько методов с одинаковы ми именами, но различным списком параметров. Такая техника называется **совмещением методов (method overloading).**
2. В Java переменные и методы класса могут быть либо элементами класса, либо элементами экземпляра (объекта) класса, в котором они объявлены, что определяется присутствием или отсутствием модификатора **static.** **Статические методы** и переменные связаны с классом, а не с экземпляром класса, и поэтому имеют название элементов класса.
3. **Модификаторы доступа** используются для управления доступностью элементов класса из других частей программы (в других классах). Применение ключевых слов **public (открытый),** **protected (защищенный), private (закрытый)** к элементам помогает управлять способностью других объектов пользоваться ими.
4. **Наследование** (inheritance), в свою очередь, упрощает практическое использование классов, так как позволяет **расширять** уже написанные и отлаженные классы, **добавляя** к ним новые свойства и возможности.
5. **Подклассом** класс, созданный с помощью механизма наследования от другого класса. Класс, который при этом наследуется (расширяется), называется **суперклассом.**
6. При расширении какого-либо класса имеется возможность использования всех написанных и отлаженных методов этого класса.
7. Каждый класс Java содержит три заранее определенные переменные, которыми можно пользоваться: null, this, super.
8. This - передает ссылку на текущий объект. Super - ссылается на суперкласс объекта.
9. Затенение переменных происходит, когда переменная во внутренней области видимости объявляется с таким же именем, с каким уже существует переменная во внешней области видимости. В данном случае переменная во внутренней области видимости затемняет переменную во внешней области видимости. Сокрытие переменной происходит в случае, когда у переменной, объявленной в дочернем классе, имя такое же, как и у переменной, объявленной в родительском классе.
10. Существует ряд способов доступа к классам в пакетах, основным из которых является импортирование пакета в начале программ:

***import ИмяПакета .\****

или

***import ИмяПакета.ИмяКлассаПакета***

1. А
2. Для создания пакета, используется следующее предложение в качестве первой строки программы в исходном тексте программы: package ИмяПакета. При компиляции этих файлов получающиеся в результате файлы с расширением .class будут помещены в каталоге, соответствующем имени пакета. (изменено)