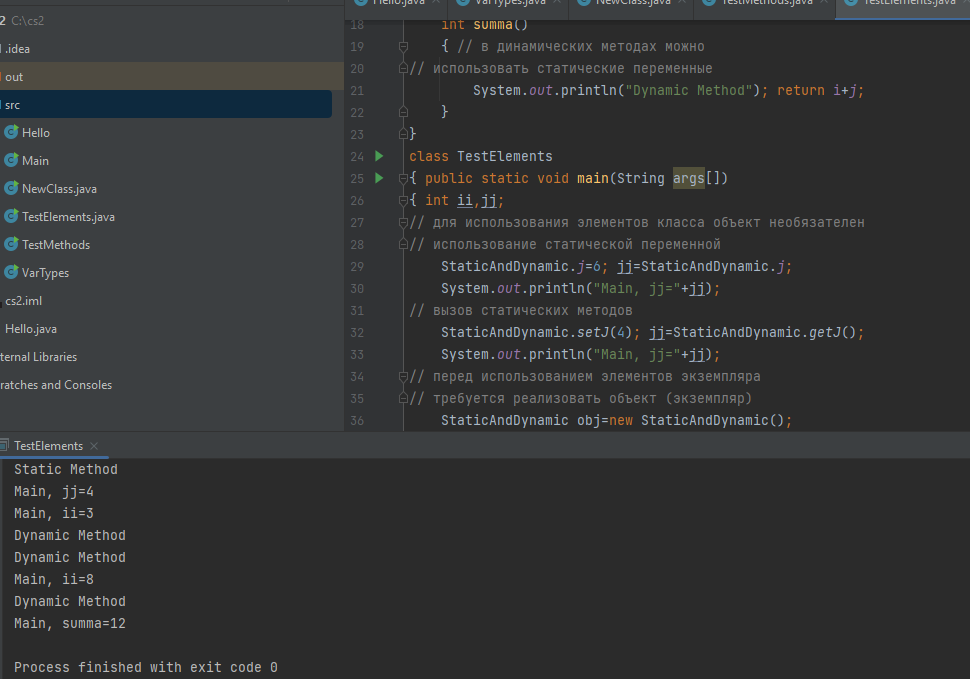
**Лабораторная работа №4**

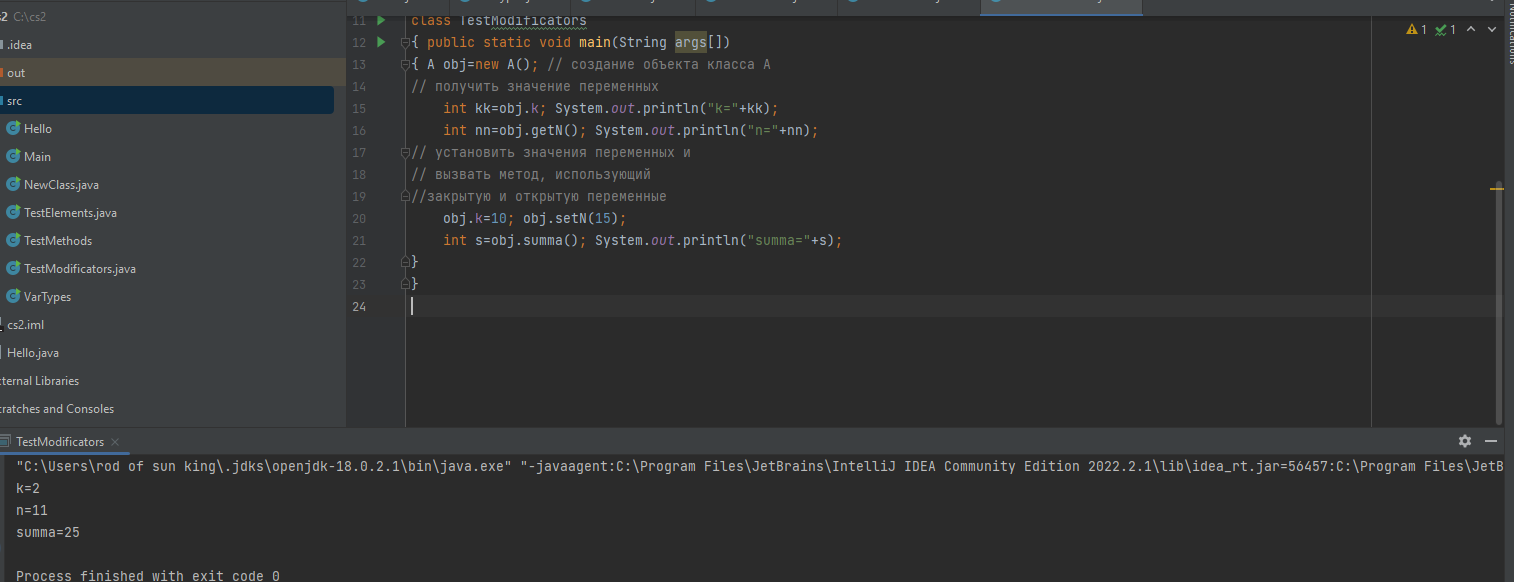
**Основы программирования на Java: классы, объекты, пакеты**

**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.

**TestElements.java**



**TestModificators.java**



Контрольные вопросы:

1. Элементы класса - это переменные и методы, объявленные внутри класса. Элементы экземпляра класса - это переменные и методы, которые относятся к каждому конкретному экземпляру класса. Элементы класса объявляются с использованием ключевого слова static, чтобы указать, что они принадлежат классу, а не конкретному экземпляру. Переменные и методы экземпляра не имеют ключевого слова static и относятся к конкретным объектам класса.
2. Модификаторы доступа используются для ограничения доступа к элементам класса, таким как переменные и методы. Существуют четыре модификатора доступа в Java:

public: элемент доступен из любого места в программе.

private: элемент доступен только внутри класса, в котором он определен.

protected: элемент доступен внутри класса, внутри подклассов и внутри того же пакета.

default (отсутствие модификатора): элемент доступен только внутри того же пакета.

1. Процесс наследования позволяет создавать новый класс (подкласс) на основе существующего класса (суперкласс). Подкласс наследует все поля и методы суперкласса и может добавлять свои собственные поля и методы или переопределять унаследованные методы. Наследование обеспечивает механизм повторного использования кода и иерархическую организацию классов.
2. Суперкласс - это класс, от которого происходит наследование. Подкласс - это класс, который наследует свойства и поведение от суперкласса. Подкласс расширяет функциональность суперкласса и может иметь свои собственные дополнительные свойства и методы.
3. Повторное использование кода означает использование существующих классов, методов и компонентов для создания новых программ или компонентов. Это позволяет избежать дублирования кода и упрощает разработку, поддержку и модификацию программного обеспечения.
4. Каждый класс Java содержит заранее определенную переменную this, которая ссылается на текущий экземпляр класса. Она используется для доступа к элементам класса из методов экземпляра.
5. Переменная this позволяет обращаться к элементам класса из методов экземпляра. Она может использоваться для доступа к переменным экземпляра, вызова других методов экземпляра или передачи самого себя в качестве параметра другому методу. Например, this.variableName обращается к переменной экземпляра, а this.methodName() вызывает другой метод экземпляра.

Переменная super используется в подклассе для обращения к элементам суперкласса. Она позволяет вызывать конструктор суперкласса, обращаться к переменным и методам суперкласса и выполнять другие действия, связанные с суперклассом.

1. Скрытие переменной (variable hiding) происходит, когда подкласс объявляет переменную с тем же именем, что и переменная в суперклассе. В результате переменная суперкласса становится недоступной в подклассе, и подкласс использует свою собственную переменную с тем же именем.

Затенение переменной (variable shadowing) происходит, когда переменная в объемлющем блоке скрывает переменную с тем же именем во вложенном блоке. Переменная во вложенном блоке затеняет переменную в объемлющем блоке.

Замещение метода (method overriding) происходит, когда подкласс реализует метод с тем же именем, сигнатурой и возвращаемым типом, что и метод в суперклассе. Замещение метода позволяет переопределить поведение унаследованного метода в подклассе.

1. Для импорта классов из пакетов в Java используется ключевое слово import. Пример:

java

import com.example.MyClass;

import java.util.ArrayList;

В этом примере класс MyClass из пакета com.example и класс ArrayList из пакета java.util импортируются для использования в текущем файле Java.

1. Явные ссылки на классы из пакетов используются для обращения к классам без необходимости использования полного имени класса с пакетом. Для использования явной ссылки на класс вместо полного имени класса с пакетом можно импортировать класс с помощью ключевого слова import и затем использовать имя класса самостоятельно. Например, если импортирован класс MyClass из пакета com.example, то можно использовать только имя класса MyClass вместо полного имени com.example.MyClass.
2. Чтобы добавить класс в пакет, необходимо указать его пакетную директиву в начале файла с исходным кодом класса. Например, если вы хотите добавить класс MyClass в пакет com.example, то в начале файла с исходным кодом должна быть указана следующая пакетная директива:

java

package com.example;

При компиляции и запуске класс будет входить в пакет com.example и станет доступным для импорта и использования других классов в этом пакете.