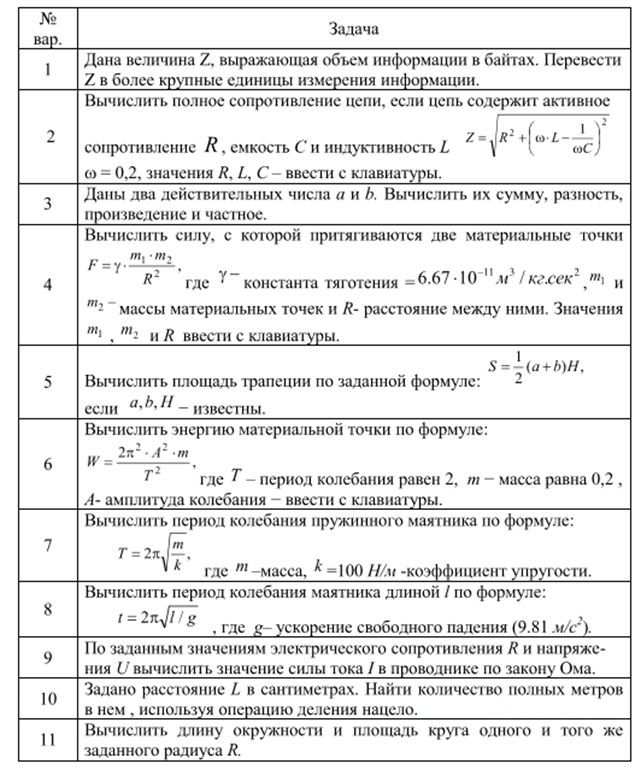
**Лабораторная работа №6**

**Основы программирования на Java: модификаторы доступа и класса**

**Цель работы:** Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки работы с модификаторами доступа в Java.

**Выполнил:** Ковалев Никита Исп22-1



Конец

Начало

|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
| L | int | Ввод |
| Lm | int | Вывод данных |

Код:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Введите L");

int L = in.nextInt();

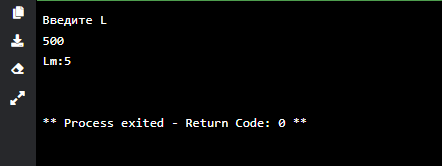
int Lm = L / 100;

System.out.println("Lm:" + Lm);

in.close();

}

}



**Контрольные вопросы для защиты:**

1. Модификатор - это ключевое слово, которое можно использовать для изменения свойств класса, метода, переменной или другого элемента программы, определяя их видимость, доступность и поведение.  
  
2. В Java существуют различные виды модификаторов, например:  
   - public: Предоставляет доступ к элементу из любого места.  
   - private: Ограничивает доступ к элементу только внутри класса.  
   - protected: Предоставляет доступ к элементу в пределах пакета и подклассам.  
   - default (package-private): Если модификатор доступа не указан, элемент доступен только в пределах пакета.   
   - static: Определяет статическое поле или метод, принадлежащее классу, а не экземпляру.  
   - final: Помечает элемент как постоянный. Не может быть изменен после инициализации.  
   - И другие.  
  
3. Модификаторы доступа:  
   - private: Элемент виден только внутри того же класса, в котором он объявлен.  
   - public: Элемент доступен из любого места, где виден класс.  
   - protected: Элемент доступен внутри пакета и подклассам, даже если они находятся в других пакетах.  
  
4. Вспомогательные модификаторы:  
   - static: Определяет статические поля и методы.  
   - final: Определяет постоянные значения.  
   - abstract: Используется в абстрактных классах и методах.  
   - synchronized: Обеспечивает потокобезопасность доступа к методу или кодовому блоку.  
   - volatile: Предотвращает кэширование переменной в потоках.  
   - transient: Исключает переменную из процесса сериализации.  
  
5. В приведенном выше примере кода мы использовали модификаторы public и static для метода main, а также public для метода calculateGeometricMean.