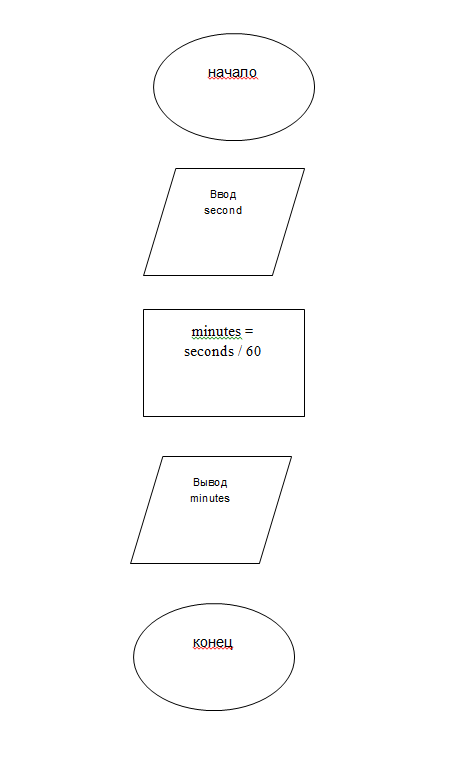
**Лабораторная работа №6**

**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.





|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
| seconds | int | Ввод данных |
| minutes | Вывод данных |

Код:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите количество секунд: ");

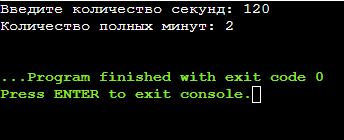
int seconds = scanner.nextInt();

int minutes = seconds / 60;

System.out.println("Количество полных минут: " + minutes);

}

}



**Контрольные вопросы**

1. Модификатор - это ключевое слово в Java, которое изменяет поведение или видимость классов, полей, методов и других элементов программы.  
  
2. В Java существуют следующие модификаторы:  
   - private: делает элемент доступным только внутри того же класса  
   - public: делает элемент доступным из любого места программы  
   - protected: делает элемент доступным внутри того же класса, пакета и для наследников в других пакетах  
   - default (без модификатора): делает элемент доступным только внутри того же пакета  
   - final: указывает, что элемент является константой или не может быть переопределен  
   - static: указывает, что элемент принадлежит классу, а не конкретному экземпляру класса  
  
3. Модификаторы доступа (private, public, protected) определяют уровень доступа к элементу класса:  
   - private: элемент доступен только внутри того же класса  
   - public: элемент доступен из любого места программы  
   - protected: элемент доступен внутри того же класса, пакета и для наследников в других пакетах  
  
4. Вспомогательные модификаторы в Java:  
   - final: указывает, что класс, метод или переменная не может быть изменены после определения  
   - abstract: указывает, что класс может содержать абстрактные методы, которые должны быть реализованы в подклассах  
   - static: указывает, что переменная или метод принадлежат классу, а не конкретному экземпляру класса  
   - synchronized: указывает, что метод или блок кода может быть использован только одним потоком одновременно  
   - volatile: указывает, что переменная является общей для нескольких потоков и должна быть синхронизирована  
  
5. В своей работе я использовал следующие модификаторы:  
   - public для методов, которые должны быть доступны из других классов и пакетов