МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

ФАКУЛЬТЕТ ЭЛЕКТОСВЯЗИ

КАФЕДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЕТЕЙ ТЕЛЕКОММУНИКАЦИЙ

Этапы работы с программой на java

Отчет по лабораторной работе №1

по дисциплине

«Конструирование программ и языки программирования»

|  |  |
| --- | --- |
| Выполнила учащаяся гр. ТП312 | Захарич Татьяна |
| Руководители | Янович Н.И  Малолеткин А.Б. |

Минск 2025

**Цель**: сформировать умения установки JDK, а также компиляции и выполнения программ на языке Java; сформировать умения программирования линейных алгоритмов на языке Java с использованием вывода данных.

**Задание**:

1. Изучить порядок установки среды программирования на Java. теоретические сведения по теме: «Изучение этапов работы с программой на java в системе программирования».

2. Установить среду программирования и выполнить запуск тестовой программы согласно указаниям в теоретических сведениях.

3. Выполнить индивидуальные задания согласно варианту.

4. Создать пакеты и JAR-файлы согласно следующего описания (рисунок 2.1.1)

5. Ответить на контрольные вопросы.

6. Составить отчет о проделанной работе.

**Индивидуальное задание (8 вариант):**

Условие:

Угол *А* задан в градусах, минутах и секундах. Найти его величину в радианах.

Была создана программа для преобразования угла из градусов, минут и секунд в радианы согласно варианту 8 индивидуального задания.

**Алгоритм решения:**  
Изучив работу с числовыми данными и математическими функциями в Java, был реализован следующий алгоритм:

1. Программа последовательно запрашивает у пользователя значения градусов, минут и секунд
2. Выполняется преобразование введенных значений в десятичные градусы по формуле: degrees + (minutes / 60.0) + (seconds / 3600.0)
3. Полученное значение преобразуется в радианы с использованием стандартного метода Math.toRadians()
4. Результаты выводятся на экран в удобочитаемом формате
5. Созданный объект Scanner закрывается для освобождения ресурсов

На рисунке 1 изображена работа программы, а в листинге 1, код.

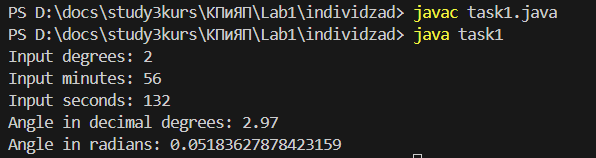
****

Рисунок 1 – Работа программы 8 варианта

import java.util.Scanner;

public class task1 {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.print("Input degrees: ");

        int degrees = in.nextInt();

        System.out.print("Input minutes: ");

        int minutes = in.nextInt();

        System.out.print("Input seconds: ");

        int seconds = in.nextInt();

        // Convert to decimal degrees

        double decimalDegrees = degrees + (minutes / 60.0) + (seconds / 3600.0);

        // Convert to radians

        double radians = Math.toRadians(decimalDegrees);

        System.out.println("Angle in decimal degrees: " + decimalDegrees);

        System.out.println("Angle in radians: " + radians);

        in.close();

    }

}

Листинг 1 – код программы 8 варианта

**Практическое задание:**

В рамках лабораторной работы было освоено применение пакетов для структурирования кода на Java. Все этапы работы выполнялись в строгом соответствии с предоставленными методическими рекомендациями.

*Формирование структуры пакетов*  
Была реализована многоуровневая файловая структура, точно соответствующая заданной системе пакетов. Каждому классу была отведена отдельная директория, путь которой отражал его полное именование в соответствии с пакетной организацией.

*Процесс компиляции*  
Преобразование исходного кода в байт-код осуществлялось с применением утилиты javac с параметром -d, определяющим целевую директорию для размещения скомпилированных файлов. Для классов, требующих наличия других классов, дополнительно применялся параметр -cp, задающий пути к соответствующим зависимостям.

*Формирование JAR-архива*  
С помощью утилиты jar был успешно создан архивный файл для пакета com.zzz.geometry. Сформированный архив содержал все необходимые скомпилированные классы указанного пакета, что свидетельствует о корректности выполнения процедуры архивации.

*Тестирование работоспособности*  
Завершающая проверка проводилась путем запуска тестового класса Test из пакета mytest. Тестирование выполнялось в двух различных конфигурациях:

С непосредственным указанием путей к директориям с классами

С использованием созданного JAR-архива в сочетании с путями к другим требуемым директориям

Обе конфигурации показали штатную работу программы, что подтвердило правильность реализации пакетной структуры и процедуры создания JAR-архива.

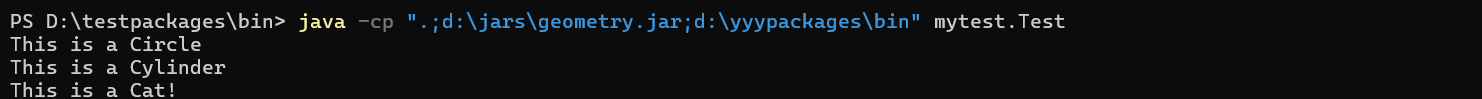


Рисунок 3 – работа программы с jar-файлами

**Контрольные вопросы**

**1. Приведите структуру программы на языке Java.**

java

package имя.пакета; // необязательное объявление пакета

import другой.пакет.Класс; // необязательный импорт внешнего класса

public class ИмяКласса {

private int поле; // переменные (поля) класса

public void метод() {

// тело метода

}

public static void main(String[] args) {

// точка входа — основной код программы

}

}

**2. Под каким именем необходимо сохранять файл с исходным кодом на языке Java?** Файл должен называться точно так же, как и публичный класс внутри него, с расширением .java. Например: если класс называется MyClass, файл должен быть MyClass.java.

**3. Назовите компоненты среды программирования на Java.**

* JDK — комплект для разработки Java-приложений
* JRE — среда выполнения Java-программ
* JVM — виртуальная машина Java, исполняющая байт-код
* Компилятор javac — преобразует исходный код в байт-код
* Интерпретатор java — запускает байт-код на JVM

**4. Изобразите в виде блок-схемы алгоритм разработки и запуска программы на языке Java.** Исходный код .java → (компиляция javac) → байт-код .class → (запуск java) → выполнение на JVM

**5. Назовите средства вывода данных в языке Java.**

* System.out.print() — вывод без переноса строки
* System.out.println() — вывод с переходом на новую строку
* System.out.printf() — форматированный вывод
* System.err.print() — вывод ошибок

**6. Что такое пакеты в Java? Какие опции и для чего используются при компиляции файлов в пакете?** Пакеты — это способ логически организовать классы по группам. При компиляции используются:

* -d — указывает директорию для размещения скомпилированных классов
* -cp или -classpath — задаёт путь к используемым классам и библиотекам

**7. Каково назначение операторов import и package?**

* package — определяет, к какому пакету принадлежит текущий класс
* import — позволяет использовать классы из других пакетов

**8. Что представляют собой, для чего и как используются JAR-файлы?** JAR-файл — это архив, содержащий Java-классы и ресурсы. Применяется для:

* упаковки готовых приложений
* распространения библиотек
* хранения кода в удобной форме

**9. Что делать, если имена классов совпадают?** В таком случае нужно использовать полное имя класса, включая его пакет — это называется квалифицированным именем.

**10. Где JVM ищет классы?** JVM ищет классы в следующих местах:

* текущая рабочая директория
* путь, указанный в переменной CLASSPATH
* системные библиотеки (например, rt.jar)
* подключённые JAR-файлы

**11. Каковы структуры хранения файлов в средах Eclipse и NetBeans?**

* В Eclipse: workspace/проект/src/пакеты/файлы.java
* В NetBeans: проект/src/пакеты/файлы.java Скомпилированные .class-файлы обычно размещаются в bin/ или build/.

**12. Приведите отличия в разработке программ на Java и C++.**

* Java работает через виртуальную машину, C++ — компилируется в машинный код
* В Java память управляется автоматически, в C++ — вручную
* Java строго объектно-ориентирован, C++ поддерживает разные парадигмы
* Java — кроссплатформенный язык, C++ — зависит от платформы и компилятора