Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационные системы и анализ данных

Центр программной инженерии

ОТЧЁТ

к лабораторной работе №1

по дисциплине

|  |
| --- |
| «Технологии разработки программных комплексов» |
| Шаблоны проектирования |

наименование темы

Вариант №8

Выполнил cтудент ИСТБ-21-2 Н. В.Зимин

номер группы подпись И. О. Фамилия

дата

Проверил Доцент З.А. Бахвалова

Должность подпись И. О. Фамилия

дата

Иркутск – 2024

# Содержание

[1 Постановка задачи 3](#_Toc116154229)

[2 Структура классов 3](#_Toc116154230)

[3 Таблица спецификаций 4](#_Toc116154231)

[4 Таблица тестов 5](#_Toc116154232)

[5 Результаты тестирования 6](#_Toc116154233)

[Исходный код 8](#_Toc116154234)

[Список использованных источников 9](#_Toc116154235)

# 1 Постановка задачи

Разработать каркас системы в соответствии с выданным вариантом

задания. При проектировании архитектуры необходимо использовать обозначенный в задании шаблон проектирования, иные варианты реализации задания не принимаются.

Программа должна функционировать в соответствии с приведённым в варианте задания примером работы.

Выбранный язык программирования должен поддерживать объектно-ориентированную парадигму. Выбор остального окружения производится студентом самостоятельно.

Индивидуальное задание:

Паттерн «Фасад».

Вы принимаете участие в создании пакета для статистического анализа данных. В ваши задачи входит создание архитектуры модуля проведения описательного анализа. В ходе анализа производится расчёт среднего выборочного, медианы и моды (отдельный класс «средние величины»), расчет значения квартилей (отдельный класс «процентили распределения»), а также расчет дисперсии и среднеквадратического отклонения (отдельный класс «показатели рассеивания»).

Необходимо реализовать каркас модуля. Для обеспечения удобного интерфейса доступа ко всему описательному анализу главный класс модуля должен иметь метод «Произвести описательный анализ», который производит все вышеперечисленные операции. При запуске программа должна вызывать этот метод главного класса модуля и выводить названия операций, которые были произведены. Программа должна быть закрыта для изменения, но открыта для расширения. Вычислений вышеприведенных характеристик производить не требуется, выводятся только названия операций.

Пример работы программы:

Было рассчитано среднее выборочное

Была рассчитана медиана

Была рассчитана мода

Были рассчитаны значения квартилей

Были рассчитаны дисперсия и среднеквадратическое отклонение.

# 2 Характеристика паттерна

Шаблон “фасад” структурный шаблон проектирования, позволяющий скрыть сложность системы путем сведения всех возможных внешних вызовов к одному объекту, делегирующему их соответствующим объектам системы.

Проблема: как обеспечить унифицированный интерфейс с набором разрозненных реализаций или интерфейсов, например, с подсистемой, если нежелательно высокое связывание с этой подсистемой или реализация подсистемы может измениться?

Решение: определить одну точку взаимодействия с подсистемой — фасадный объект, обеспечивающий общий интерфейс с подсистемой, и возложить на него обязанность по взаимодействию с её компонентами. Фасад — это внешний объект, обеспечивающий единственную точку входа для служб подсистемы. Реализация других компонентов подсистемы закрыта и не видна внешним компонентам.

# 3 Шаги реализация паттерна

# 4 UML-диаграмма классов

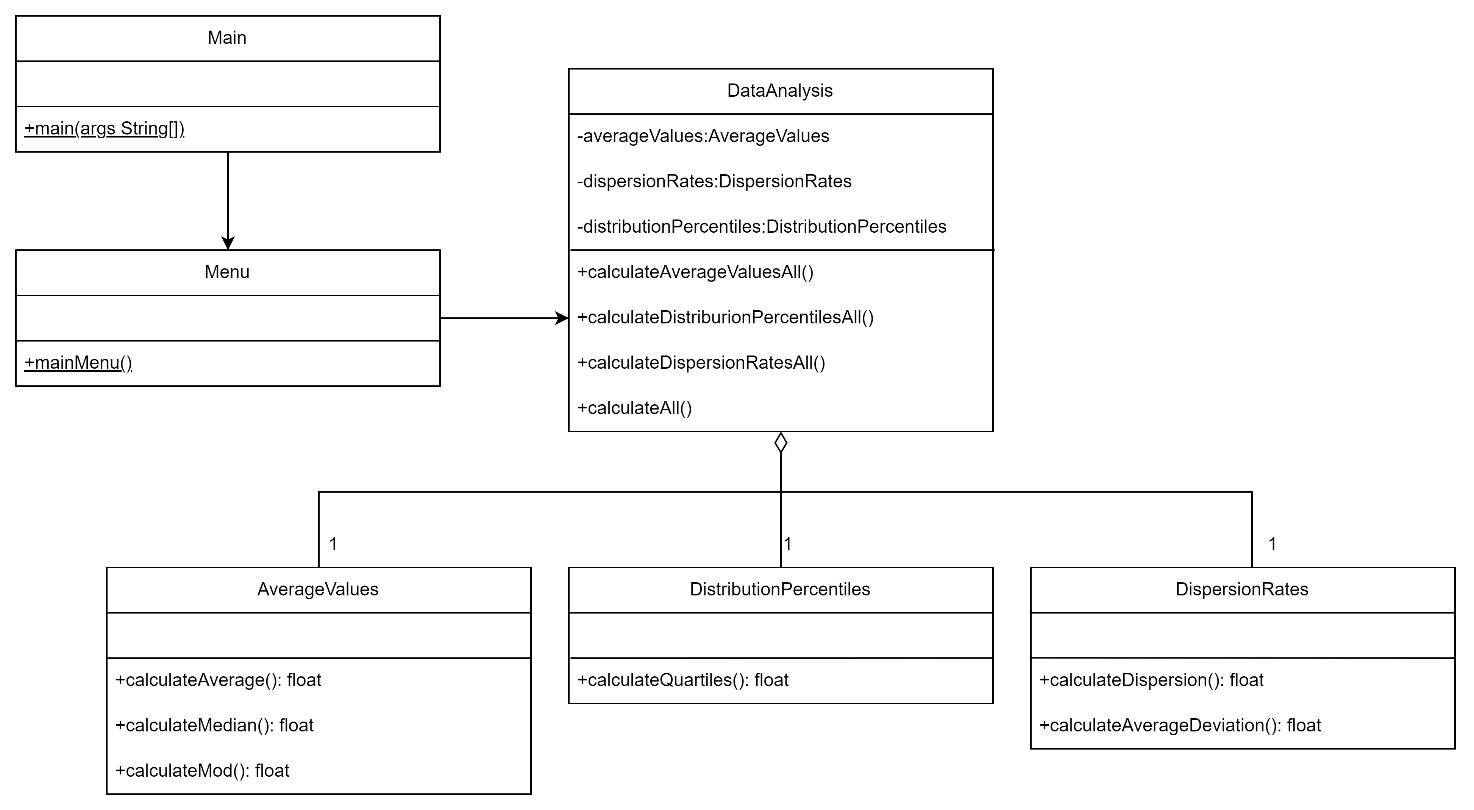


Рисунок 1 – Диаграмма классов программы

1. Main – начальный класс, точка входа в программу.
2. Menu -класс, реализующий интерфейс программы в консоли.
   1. public static void mainMenu() – метод, отображающий главное меню программы и реализующий взаимодействие с объектом класса DataAnalysis.
3. DataAnalysis – класс, реализующий главную точку доступа к подсистемам модуля «статистического анализа» в шаблоне «фасад»
   1. private AverageValues averageValues
   2. private DispersionRates dispersionRates
   3. private DistributionPercentiles distributionPercentiles
   4. public void calculateAverageValuesAll()
   5. public void calculateDispersionRatesAll()
   6. public void calculateDistriburionPercentilesAll()
   7. public void calculateAll()
4. AverageValues – класс, реализующий подсистему «средних величин».
   1. public float calculateAverage()
   2. public float calculateMedian()
   3. public float calculateMod()
5. DistributionPercentiles – класс, реализующий подсистему «процентилей распределения».
   1. public float calculateQuartiles()
6. DispersionRates – класс, реализующий подсистему «показателей рассеивания».
   1. public float calculateDispersion()
   2. public float calculateAverageDeviation()

# 5 Результаты тестирования

# 6 Исходный код

|  |
| --- |
| import Calculate.DataAnalysis;  import java.util.InputMismatchException;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Menu.mainMenu();  }  }  class Menu  {  public static void mainMenu()  {  DataAnalysis dataAnalysis = new DataAnalysis();  Scanner scanner = new Scanner(System.in);  byte command = 0;  while(true)  {  System.out.println("--------------------------------------------------");  System.out.println("Введите команду");  System.out.println("1 - Произвести описательный анализ");  System.out.println("2 - Произвести расчет средних величин");  System.out.println("3 - Произвести расчет процентилей распределения");  System.out.println("4 - Произвести расчет показателей рассеивания");  System.out.println("0 - Закрыть программу");  System.out.println("--------------------------------------------------");    if(scanner.hasNextByte()) command = scanner.nextByte();  else {System.out.println("Ошибка ввода, повторите команду"); scanner.next();continue;}  switch (command) {  case 1: dataAnalysis.calculateAll();break;  case 2: dataAnalysis.calculateAverageValuesAll();break;  case 3: dataAnalysis.calculateDistriburionPercentilesAll();break;  case 4: dataAnalysis.calculateDispersionRatesAll();break;  case 0: System.out.println("Завершение работы");return;  }    }  }  } |

Ссылка на полный проект [https://github.com/CrusaderKleptoman/trainingPatterns.git](https://github.com/CrusaderKleptoman/trainingPatterns.git%20)

# Список использованных источников

1. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер-ДМК, 2001.
2. M. Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley, 1997.
3. М. Фаулер и др. Архитектура корпоративных программных приложений. М.: Вильямс, 2004.