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](#_Toc163117882)

[2 Характеристика паттерна 4](#_Toc163117883)

[3 Шаги реализация паттерна 5](#_Toc163117884)

[4 UML-диаграмма классов 6](#_Toc163117885)

[5 Результаты тестирования 8](#_Toc163117886)

[6 Исходный код 9](#_Toc163117887)

[Список использованных источников 12](#_Toc163117888)

# 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

1. Определить основную систему в рамках выполняемой задачи.
2. Разделить систему на самостоятельные подсистемы.
3. Реализовать фасадный объект, обеспечивающий точку взаимодействия с подсистемами

# 4 UML-диаграмма классов

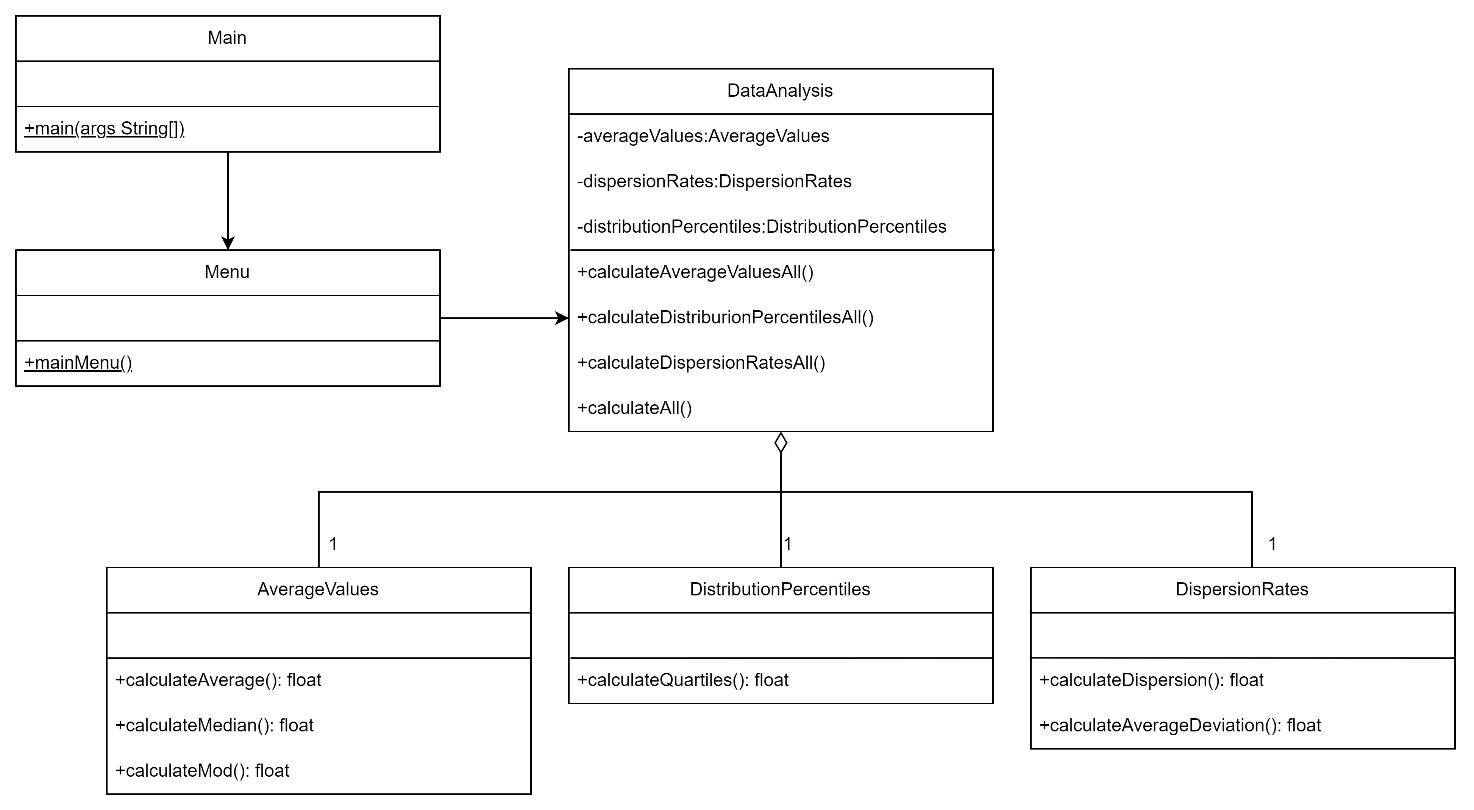


Рисунок 1 – Диаграмма классов программы

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» в рамках системы «статистический анализ».
   1. private AverageValues averageValues – поле класса AverageValues, реализующее доступ к подсистеме «средних величин».
   2. private DispersionRates dispersionRates – поле класса DispersionRates, реализующее доступ к подсистеме «показателей рассеивания».
   3. private DistributionPercentiles distributionPercentiles – поле класса DistributionPercentiles, реализующее доступ к подсистеме «процентилей распределения».
   4. public void calculateAverageValuesAll() – метод, реализующий вызов подсистемы «средних величин» для расчётов.
   5. public void calculateDispersionRatesAll() – метод, реализующий вызов подсистемы «показателей рассеивания» для расчётов.
   6. public void calculateDistriburionPercentilesAll() – метод, реализующий вызов подсистемы «процентилей распределения» для расчётов.
   7. public void calculateAll() – метод, реализующий вызов всех подсистем системы «статистический анализ» для расчётов
4. AverageValues – класс, реализующий подсистему «средних величин».
   1. public float calculateAverage() – метод, реализующий подсчёт средних величин.
   2. public float calculateMedian() – метод, реализующий подсчёт медианы.
   3. public float calculateMod() – метод, реализующий подсчёт моды.
5. DistributionPercentiles – класс, реализующий подсистему «процентилей распределения».
   1. public float calculateQuartiles() – метод, реализующий подсчёт квартилей.
6. DispersionRates – класс, реализующий подсистему «показателей рассеивания».
   1. public float calculateDispersion() – метод, реализующий подсчёт дисперсии.
   2. public float calculateAverageDeviation() – метод, реализующий подсчёт среднеквадратичного отклонения.

# 5 Результаты тестирования работы программы

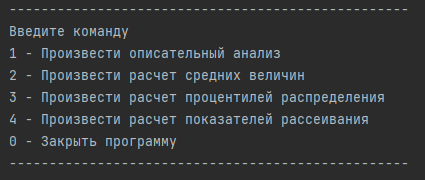


Рисунок 2 – начало работы программы, вывод главного меню

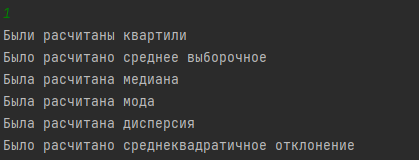


Рисунок 3 – вызов полного описательного анализа

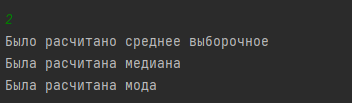


Рисунок 4 – вызов подсистемы «средние величины»



Рисунок 5 – вызов подсистемы «процентилей распределения»

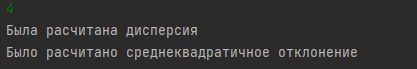


Рисунок 6 - вызов подсистемы «показатели рассеивания»

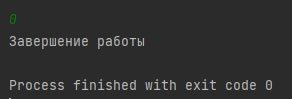


Рисунок 7 – завершение работы программы

# 6 Исходный код

Код класса Main

|  |
| --- |
| public class Main {  public static void main(String[] args) {  Menu.mainMenu();  }  } |

Код класса Menu

import Calculate.DataAnalysis;

import java.util.Scanner;

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;

}

}

}

}

Код класса DataAnalisys

package Calculate;

public class DataAnalysis {

private AverageValues averageValues = new AverageValues();

private DispersionRates dispersionRates = new DispersionRates();

private DistributionPercentiles distributionPercentiles = new DistributionPercentiles();

public DataAnalysis() {

}

public void calculateAverageValuesAll()

{

averageValues.calculateAverage();

averageValues.calculateMedian();

averageValues.calculateMod();

}

public void calculateDispersionRatesAll()

{

dispersionRates.calculateDispersion();

dispersionRates.calculateAverageDeviation();

}

public void calculateDistriburionPercentilesAll()

{

distributionPercentiles.calculateQuartiles();

}

public void calculateAll()

{

calculateDistriburionPercentilesAll();

calculateAverageValuesAll();

calculateDispersionRatesAll();

}

}

Код класса AverageValues

package Calculate;

//класс реализующий подсчёт средних величин статистического анализа

public class AverageValues {

public AverageValues() {

}

//метод возвращающий среднее выборочное

public float calculateAverage()

{

//Реализовать подсчет среднего выборочного

System.out.println("Было расчитано среднее выборочное");

return 0;

}

//метод возвращающий медиану

public float calculateMedian()

{

//Реализовать подсчет медианы

System.out.println("Была расчитана медиана");

return 0;

}

//метод возвращающий моду

public float calculateMod()

{

//Реализовать подсчет моды

System.out.println("Была расчитана мода");

return 0;

}

}

Код класса DistributionPercentiles

package Calculate;

public class DistributionPercentiles {

public DistributionPercentiles() {

}

//метод возвращающий значение квартилей

public float calculateQuartiles()

{

//Реализовать подсчет квартилей

System.out.println("Были расчитаны квартили");

return 0;

}

}

Код класса DispersionRates

package Calculate;

public class DispersionRates {

public DispersionRates() {

}

//метод возвращающий значение дисперсии

public float calculateDispersion()

{

//Реализовать подсчет дисперсии

System.out.println("Была расчитана дисперсия");

return 0;

}

//метод возвращающий значение среднеквадратического отклонения

public float calculateAverageDeviation()

{

//Реализовать подсчет среднеквадратического отклонения

System.out.println("Было расчитано среднеквадратичное отклонение");

return 0;

}

}

Ссылка на полный проект <https://github.com/CrusaderKleptoman/pattern.git>

# 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.