**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

**Лабораторная работа №4**

по дисциплине: Метрология, стандартизация и сертификация программного обеспечения

тема: «Метрики объектно-ориентированных программных систем»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверили:

ст. пр. Осипов Олег Васильевич

Белгород 2025 г.

**Лабораторная работа №4  
Метрики объектно-ориентированных программных систем.  
Вариант 3**

**Цель работы:** изучить теоретические сведения и получить практические навыки оценки иерархии классов объектно-ориентированных программных систем.

**Задания для выполнения к работе:**

1. Реализовать диаграмму классов собственной объектно-ориентированной программной системы.
2. Для каждого класса указать все его свойства и методы, кратко охарактеризовать их назначение и смысл.
3. Определить значения метрик из набора метрик Чидамбера и Кемерера.
4. Сформулировать рекомендации по модификации составленной иерархии классов на основании вычисленных значений метрик Чидамбера и Кемерера.
5. Определить значения метрик из набора метрик Лоренца и Кидда.
6. Сформулировать рекомендации по модификации составленной иерархии классов на основании вычисленных значений метрик Лоренца и Кидда.

**Выполнение работы**

**Программа “Календарь”**

**Диаграмма классов**

Изображение выглядит как текст, снимок экрана, Параллельный

Автоматически созданное описание

**Описание классов из курсовой работы по ООП**

**Event**

**Описание:** Класс, представляющий событие в календаре.

**Свойства:**

* description – описание события.
* day, month, year – дата события.
* hour, minute – время события.
* period, frequency – период и частота повторения события.

**Методы:**

* operator< – перегрузка оператора для сравнения событий по времени.
* operator== – перегрузка оператора для сравнения событий на равенство.

**EventManager**

**Описание:** Класс для управления событиями, включая их добавление, удаление, обновление и просмотр.

**Свойства:**

* events\_ – мультикарта для хранения событий с ключом по дню.
* fileManager\_ – менеджер файлов для сохранения и загрузки событий.

**Методы:**

* addEvent – добавление события.
* deleteEvent – удаление события.
* viewEvents – просмотр событий на указанную дату.
* viewAllEvents – просмотр всех событий.
* saveEventsToFile – сохранение событий в файл.
* loadEventsFromFile – загрузка событий из файла.
* updateEvent – обновление события.
* getEvents – получение всех событий.

**FileManager**

**Описание:** Класс для работы с файлами, сохранения и загрузки событий.

**Свойства:**  
Отсутствуют.

**Методы:**

* saveEventsToFile – сохранение событий в файл.
* loadEventsFromFile – загрузка событий из файла.

**Calendar**

**Описание:** Класс для отображения календаря и работы с событиями.

**Свойства:**  
Отсутствуют.

**Методы:**

* printCalendar – вывод календаря на указанный месяц и год.
* addEvent – добавление события.
* deleteEvent – удаление события.
* viewEvents – просмотр событий на указанную дату.
* viewAllEvents – просмотр всех событий.
* updateEvent – обновление события.
* showMonthlyCalendar – отображение календаря на месяц.
* isLeapYear – проверка, является ли год високосным.
* daysInMonth – получение количества дней в месяце.
* isHoliday – проверка, является ли день праздничным.

**Notification**

**Описание:** Класс для отправки уведомлений о событиях.

**Свойства:**  
Отсутствуют.

**Методы:**

* notifyEvents – отправка уведомлений о событиях.

**Time**

**Описание:** Класс для работы с временем.

**Свойства:**  
Отсутствуют.

**Методы:**

* displayCurrentDateTime – отображение текущей даты и времени.

**Interface**

**Описание:** Класс для взаимодействия с пользователем через консольный интерфейс.

**Свойства:**  
Отсутствуют.

**Методы:**

* printMenu – вывод меню на экран.
* getUserChoice – получение выбора пользователя.
* processChoice – обработка выбора пользователя.
* addEvent – добавление события через интерфейс.
* deleteEvent – удаление события через интерфейс.
* viewEvents – просмотр событий на дату через интерфейс.
* viewAllEvents – просмотр всех событий через интерфейс.
* updateEvent – обновление события через интерфейс.
* showMonthlyCalendar – показ календаря на месяц через интерфейс.

**Метрики Чидамбера и Кемерера**

| **Класс** | **WMC**  ССМК | **DIT**  ГДН | **NOC**  КПК | **CBO**  СМО | **RFC**  КВВМ | **LCOM**  НМ | **LOCM**  СКСКМ |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Event | 2 | 0 | 0 | 0 | 2 | 1 | 1 |
| EventManager | 8 | 0 | 0 | 2 | 15 | 3 | 5 |
| FileManager | 2 | 0 | 0 | 1 | 2 | 1 | 1 |
| Calendar | 10 | 0 | 0 | 3 | 18 | 4 | 6 |
| Notification | 1 | 0 | 0 | 2 | 1 | 0 | 0 |
| Time | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Interface | 9 | 0 | 0 | 4 | 20 | 5 | 4 |

**Вывод:**

* WMC (Weighted Methods per Class) показывает сложность классов. Наибольшие значения у Calendar и Interface, что указывает на их высокую сложность.
* DIT (Depth of Inheritance Tree) равен 0, так как в проекте отсутствует наследование.
* CBO (Coupling Between Objects) умеренный, что говорит о приемлемой связности классов.
* RFC (Response For a Class) высок у Interface и Calendar, что может усложнить тестирование.
* LCOM (Lack of Cohesion in Methods) в норме, но требует внимания для Interface и Calendar.

**Рекомендации:**

* Упростить классы Calendar и Interface, разделив их на более мелкие компоненты.
* Уменьшить связность между классами, особенно для EventManager и FileManager.

**Метрики Лоренца и Кидда**

| **Класс** | **NOO**  КМК | **NOA**  КАК | **SISI**  РИ | **OSavg**  СРМ | **OCavg**  ССМ | **NPavg**  СКП | **NSS**  КВВ | **NKS**  КИК | **NSUB**  КП |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Event | 0 | 0 | 0 | 0 | 0 | 1.5 | 0 | 0 | 0 |
| EventManager | 6 | 0 | 0 | 0.125 | 1 | 1 | 0 | 0 | 0 |
| FileManager | 0 | 0 | 0 | 0 | 0 | 0.5 | 0 | 0 | 0 |
| Calendar | 8 | 0 | 0 | 0.1 | 0.8 | 1.2 | 0 | 0 | 0 |
| Notification | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Time | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Interface | 7 | 0 | 0 | 0.1 | 0.9 | 1.1 | 0 | 0 | 0 |

**Вывод:**

* NOO (Number of Operations) высок у Calendar и Interface, что подтверждает их сложность.
* NOA (Number of Attributes) в норме.
* SISI (Size of Interface) равен 0, так как интерфейсы не используются.
* OSavg и OCavg в пределах нормы.
* NPavg (Number of Parameters) умеренный, но требует оптимизации для некоторых методов.

**Рекомендации:**

* Уменьшить количество параметров в методах классов Calendar и Interface.
* Оптимизировать взаимодействие между классами для снижения сложности.

**Вывод:** в ходе лабораторной работы изучены теоретические сведения и получены практические навыки оценки иерархии классов объектно-ориентированных программных систем. На основании метрик предложены рекомендации по улучшению структуры кода.