

Методические указания к выполнению лабораторных работ по дисциплине
«Технология разработки программного обеспечения»

Направление подготовки
09.04.01 «Информатика и вычислительная техника»

Киров 2019

Содержание

Введение.....	4
1 Базовый курс.....	5
1.1 Лабораторная работа № 1. Знакомство с языком программирования Java и средой разработки Eclipse	5
1.2 Лабораторная работа №2. Использование средств автоматической генерации документации и рефакторинга программного кода.....	7
1.3 Лабораторная работа №3. Организация процесса разработки ПО с применением системы контроля версий.....	8
Установка расширений в Eclipse выполняется с помощью пункта Help – Install New Software.....	9
1.4 Лабораторная работа №4. Разработка библиотеки на языке программирования Java	9
1.5 Лабораторная работа №5. Разработка пользовательского интерфейса с использованием технологии swing.....	10
1.6 Лабораторная работа №6. Знакомство с инструментом автоматизации решения задач Apache Ant.....	11
1.7 Лабораторная работа №7. Использование средств автоматизации тестирования.....	11
1.8 Лабораторная работа №8. Профилирование и оптимизации Java-программ	12
1.9 Лабораторная работа №9. Разработка приложения с использование шаблонов проектирования	13
1.10 Лабораторная работа №10. Автоматизация генерации UML-диаграмм.	13
1.11 Лабораторная работа №11. Разработка базы данных с применением CASE-средств	14
1.12 Лабораторная работа №12. Разработка приложения с применением CASE-средств	15
2 Углубленный курс.....	17

2.1 Лабораторная работа №1. Разработка библиотеки на языке программирования Java	17
2.2 Лабораторная работа №2. Организация процесса разработки ПО с применением системы контроля версий.....	18
2.3 Лабораторная работа №3. Формальная верификация	19
2.4 Лабораторная работа №4. Профилирование и оптимизации Java-программ	20
2.5 Лабораторная работа №5. Разработка пользовательского интерфейса с использованием технологий javaFX.....	21
2.6 Лабораторная работа №6. Знакомство с инструментом автоматизации сборки проектов Apache Maven	21
2.7 Лабораторная работа №7. Декомпиляция приложения на языке программирования Java	22
2.8 Лабораторная работа №8. Защита Java-приложений с помощью обфускации	23
2.9 Лабораторная работа №9. Разработка приложения с использование шаблонов проектирования	23
2.10 Лабораторная работа №10. Автоматизация генерации UML-диаграмм .	24
2.11 Лабораторная работа №11. Разработка базы данных с применением CASE-средств	25
2.12 Лабораторная работа №12. Разработка приложения с применением CASE-средств	26
3 Справочные сведения	27
3.1 Встроенные средства рефакторинга среды разработки Eclipse	27
3.2 Описание дескрипторов JavaDoc.....	28
3.3 Краткие сведения по работе со средством профилирования hprof.....	31
Заключение	33
Список литературы	34

Введение

Представленное пособие содержит методические указания к выполнению лабораторного практикума по дисциплине «Технология разработки программного обеспечения». Данная дисциплина осваивается на протяжении первых двух семестров подготовки магистрантов по направлению 09.04.01 – «Информатика и вычислительная техника» вне зависимости от выбранной направленности.

Курс разбит на две части по выбору – базовый и углубленный. Выбор первой части рекомендуется для студентов, не имеющих специализированных навыков программирования. Вторая часть больше подойдет студентам, которые в процессе освоения программы бакалавриата уделяли программированию значительную часть времени.

1 Базовый курс

Данный курс лабораторных работ предназначен для магистрантов, знакомых только с основами программирования.

1.1 Лабораторная работа № 1. Знакомство с языком программирования Java и средой разработки Eclipse

Целью работы является знакомство с языком программирования Java и интегрированной средой разработки Eclipse.

1.1.1 Задание. В соответствии с выбранным вариантом разработать набор классов на языке программирования Java. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Установить на компьютер виртуальную Java-машину (JDK).
- Установить на компьютер среду разработки Eclipse.
- Настроить рабочее окружение.
- Создать новый проект.
- Реализовать группу классов.
- Продемонстрировать работу классов на подготовленном сценарии.

1.1.2 Примерные варианты заданий.

Вариант №1. Разработать класс для генерации массива простых чисел. Класс должен иметь два публичных метода:

- `int getRandomPrime()` – возвращает случайное простое число из диапазона $[2, 10^9]$;
- `int[] getRandomArray(int length)` – возвращает упорядоченный по возрастанию массив простых чисел размерности `length`;

и два внутренних метода:

- `int getNext(int prime)` – возвращает следующее после `prime` простое число из диапазона $[2, 10^9]$;
- `boolean isPrime(int arg0)` – определяет является ли заданный аргумент простым числом.

Продемонстрировать работу класса.

Вариант №2. Разработать пакет, содержащий классы фигура (абстрактный), круг, прямоугольник, квадрат. Каждый класс должен иметь публичный метод для вычисления площади фигуры. Продемонстрировать работу пакета.

Вариант №3. Разработать класс `BigFraction` для работы с дробной длинной арифметикой. Класс должен содержать несколько публичных конструкторов, публичные методы сложения, вычитания, умножения и деления дробей. Продемонстрировать работу класса.

Вариант №4. Разработать класс для работы со строками. Класс должен содержать публичные методы, позволяющие эффективно определять, является ли одна строка префиксом, суффиксом и подстрокой другой строки. Продемонстрировать работу класса.

Вариант №5. Разработать класс, который позволяет удалять из выбранного файла заданные лексемы. Файл, из которого производится удаление, и файл, содержащий лексемы, задаются аргументом командной строки. Реализовать сценарий, демонстрирующий работу класса. Продемонстрировать работу класса.

Вариант №6. Реализовать класс `SmallInteger` для работы с целыми числами. Класс должен содержать публичные методы сложения, вычитания, умножения и деления целых чисел, не превосходящих по абсолютному значению 10^4 . Продемонстрировать работу класса.

Вариант №7. Разработать класс для сбора статистической информации о текстовом файле (уникальные слова, их количество, количество букв и т.д.). Продemonстрировать работу класса.

Вариант №8. Реализовать класс для работы с матрицами. Класс должен содержать публичные методы сложения, вычитания и умножения матриц. Продemonстрировать работу класса.

1.1.3 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, листинг разработанной программы с комментариями и выводы по проделанной работе.

1.2 Лабораторная работа №2. Использование средств автоматической генерации документации и рефакторинга программного кода

Целью работы является знакомство с встроенными в среду разработки Eclipse средствами для автоматической генерации документации и рефакторинга кода.

1.2.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести ряд преобразований программного кода, полученного в ходе выполнения предыдущей работы, с использованием встроенных средств рефакторинга.
- Сопроводить код комментариями с использованием JavaDoc.
- Сгенерировать документацию к разработанным классам.

1.2.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, листинг разработанной программы с комментариями, сгенерированную документацию и выводы по проделанной работе.

1.3 Лабораторная работа №3. Организация процесса разработки ПО с применением системы контроля версий

Целью работы является освоение метода установки расширений для IDE Eclipse, а также получение практических навыков взаимодействия с системой контроля версий Subversion.

1.3.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Установить расширение Subversive для Eclipse.
- Зарегистрировать проект, выполненный в ходе предыдущей лабораторной работы, на ресурсе, предоставляющем пространство для Subversion.
- Добавить ссылку на созданный репозиторий в Eclipse.
- Синхронизировать репозиторий с проектом.
- Изменить функционал проекта.
- Обновить файлы в репозитории.
- Откатить сделанные изменения с помощью информации из репозитория.

1.3.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, детализированное описание хода выполнения лабораторной работы и выводы по проделанной работе.

Пространство для проектов Subversion предоставляется несколькими ресурсами. Например, <http://code.google.com/> или <http://sourceforge.net/>.

Информация по работе с Subversion может быть найдена на ресурсе <http://svnbook.red-bean.com/>.

Информация по работе с Subversive может быть найдена на ресурсе <http://www.eclipse.org/subversive/>.

Установка расширений в Eclipse выполняется с помощью пункта Help – Install New Software.

1.4 Лабораторная работа №4. Разработка библиотеки на языке программирования Java

Целью работы является получение практических навыков разработки эффективных алгоритмических библиотек на языке программирования Java.

1.4.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать вариант задания и согласовать тему разработки с преподавателем.
- Реализовать библиотеку.
- Разработать несколько типичных сценариев использования библиотеки.

1.4.2 Примерные варианты заданий

Вариант №1. Разработать библиотеку для работы с матрицами, реализовать несколько матричных алгоритмов.

Вариант №2. Разработать библиотеку для работы с графами, реализовать несколько графовых алгоритмов.

Вариант №3. Разработать библиотеку для работы со словарем.

Вариант №4. Разработать библиотеку для работы со строками, реализовать несколько строковых алгоритмов.

Вариант №5. Разработать библиотеку для работы с простыми числами, реализовать несколько алгоритмов теории чисел.

1.4.3 Указания к выполнению работы. Библиотека должна содержать не менее трех асимптотически сложных алгоритма.

Библиотека должна состоять из не менее чем трех классов.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы (назначение и архитектуру, достоинства), листинг разработанной программы, описание нескольких типичных сценариев использования и выводы по проделанной работе.

1.5 Лабораторная работа №5. Разработка пользовательского интерфейса с использованием технологии swing

Целью работы является получение навыков разработки графического пользовательского интерфейса с применением технологии Swing.

1.5.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать и согласовать с преподавателем задачу, для решения которой может быть использована библиотека, разработанная в ходе предыдущей лабораторной работы.

- Разработать программу для решения выбранной задачи (взаимодействие с пользователем должно осуществляться с применением пользовательского интерфейса).

1.5.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, диаграммы, описывающие разработанный интерфейс, программный код, экранные формы, выводы по проделанной работе.

1.6 Лабораторная работа №6. Знакомство с инструментом автоматизации решения задач Apache Ant

Целью работы является знакомство с базовыми возможностями средства автоматизации решения повседневных задач Apache Ant.

1.6.1 Задание. Для выполнения лабораторной работы необходимо написать скрипт для автоматической сборки проекта, реализованного в ходе выполнения предыдущей лабораторной работы.

1.6.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, текст разработанного скрипта, выводы по проделанной работе.

Дополнительные сведения по Apache Ant могут быть найдены по ссылке <http://ant.apache.org/manual/index.html>.

1.7 Лабораторная работа №7. Использование средств автоматизации тестирования

Целью работы является получение практических навыков использования средств автоматизации тестирования.

1.7.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Разработать несколько типичных сценариев использования библиотеки, полученной в ходе лабораторной работы №4.
- Выполнить построение тестов, покрывающих код.
- Провести тестирование с использованием JUnit.

1.7.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы (назначение и архитектуру, достоинства), листинг разработанной программы, описание нескольких типичных сценариев использования, описание хода тестирования, примеры генераторов тестов и выводы по проделанной работе.

1.8 Лабораторная работа №8. Профилирование и оптимизации Java-программ

Целью работы является получение практических навыков профилирования и оптимизации программ.

1.8.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести профилирование разработанных в ходе лабораторной работы №4 сценариев.
- Выявить «узкие» места в работе.
- Проанализировать код используемых функций.
- Выполнить оптимизацию кода.
- Сравнить результаты работы до и после оптимизаций.

1.8.2 Указания к выполнению работы. Профилирование необходимо выполнить с помощью одного из специализированных средств: jconsole, hprof, jvisualvm.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание библиотеки и сценариев, разработанных в ходе предыдущей лабораторной работы, листинг разработанных сценариев, результаты профилирования сценариев, описание выполненных оптимизаций, результаты оптимизаций.

1.9 Лабораторная работа №9. Разработка приложения с использованием шаблонов проектирования

Целью работы является получение практических навыков разработки программ на языке программирования Java с применением шаблонов проектирования, а также закрепление знаний, полученных на лекциях и предыдущих лабораторных работах.

1.9.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать задание в соответствии с целью работы. Согласовать тему разработки с преподавателем.
- Разработать структуру приложения.
- В процессе реализации применить шаблоны проектирования там, где это является целесообразным.

1.9.2 Указания к выполнению работы. Допускается связь разрабатываемого приложения с темой курсового проекта.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы (назначение, структуру, достоинства, использованные в процессе разработки средства), листинг разработанной программы и выводы по проделанной работе.

1.10 Лабораторная работа №10. Автоматизация генерации UML-диаграмм

Целью работы является получение практических навыков использования средств, предназначенных для генерации UML-диаграмм по готовому программному коду.

1.10.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализа доступных средств, позволяющих выполнять генерацию диаграмм классов по готовому программному коду.
- Выбрать, установить и настроить одно из доступных средств.
- Выполнить генерацию диаграммы классов для приложения, разработанного в ходе предыдущей лабораторной работы.
- Отметить на полученной диаграмме классов использованные шаблоны проектирования. Сопроводить изображение необходимыми поясняющими комментариями.

1.10.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, краткий сравнительный анализ средств для автоматической генерации UML-диаграмм, листинг разработанной в ходе предыдущей лабораторной работы программы, сгенерированную диаграмму классов, описание выделенных шаблонов проектирования и выводы по проделанной работе.

1.11 Лабораторная работа №11. Разработка базы данных с применением CASE-средств

Целью работы является получение практических навыков использования CASE-средств, предназначенных для проектирования и создания баз данных.

1.11.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать предметную область и выполнить постановку задания на разработку, согласовав его с преподавателем.

- Провести анализа доступных CASE-средств, позволяющих выполнять проектирование и создание баз данных.
- Выбрать, установить и настроить одно из доступных средств.
- В соответствии с поставленным заданием спроектировать и создать базу данных.

1.11.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, постановку задачи, краткий сравнительный анализ CASE-средств для проектирования и создания баз данных, описание процесса проектирования базы данных и выводы по проделанной работе.

1.12 Лабораторная работа №12. Разработка приложения с применением CASE-средств

Целью работы является получение практических навыков использования CASE-средств, предназначенных для проектирования и создания приложений.

1.12.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализа доступных CASE-средств, позволяющих выполнять проектирование и создание приложений.
- Выбрать, установить и настроить одно из доступных средств.
- В соответствии с поставленным в ходе предыдущей лабораторной работы заданием спроектировать и создать приложение, которое будет использовать полученную ранее базу данных.

1.12.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, постановку задачи, краткий сравнительный анализ CASE-средств для проектирования и создания приложений, описание процесса проектирования, листинг разработанного приложения, экранные формы и выводы по проделанной работе.

2 Углубленный курс

Данный курс лабораторных работ предназначен для магистрантов, программирующих на уровне выше среднего, а также для выпускников бакалавриата по направлению подготовки «Информатика и вычислительная техника», профиль «Программное обеспечение вычислительной техники и автоматизированных систем».

2.1 Лабораторная работа №1. Разработка библиотеки на языке программирования Java

Целью работы является получение практических навыков разработки эффективных алгоритмических библиотек на языке программирования Java.

2.1.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать вариант задания и согласовать тему разработки с преподавателем.
- Реализовать библиотеку.
- Разработать несколько типичных сценариев использования библиотеки.

2.1.2 Примерные варианты заданий.

Вариант №1. Разработать библиотеку для работы с матрицами, реализовать несколько матричных алгоритмов.

Вариант №2. Разработать библиотеку для работы с графами, реализовать несколько графовых алгоритмов.

Вариант №3. Разработать библиотеку для работы со словарем.

Вариант №4. Разработать библиотеку для работы со строками, реализовать несколько строковых алгоритмов.

Вариант №5. Разработать библиотеку для работы с простыми числами, реализовать несколько алгоритмов теории чисел.

2.1.3 Указания к выполнению работы. Библиотека должна содержать не менее трех асимптотически сложных алгоритма.

Библиотека должна состоять из не менее чем трех классов.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы (назначение и архитектуру, достоинства), листинг разработанной программы, описание нескольких типичных сценариев использования и выводы по проделанной работе.

2.2 Лабораторная работа №2. Организация процесса разработки ПО с применением системы контроля версий

Целью работы является получение практических навыков взаимодействия с системой контроля версий Git.

2.2.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Установить и настроить систему контроля версий Git, создать локальный репозиторий для проекта, полученного в ходе выполнения предыдущей лабораторной работы.
- Создать удаленный репозиторий на ресурсе, предоставляющем пространство для Git-проектов, и выполнить синхронизацию с ним.
- Изменить функционал проекта.
- Обновить файлы в репозитории.

- Откатить сделанные изменения с помощью информации из репозитория.

- С помощью ветвления смоделировать ряд конфликтных ситуаций и продемонстрировать их автоматическое, полуавтоматическое и ручное разрешение.

2.2.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, детализированное описание хода выполнения лабораторной работы и выводы по проделанной работе.

2.3 Лабораторная работа №3. Формальная верификация

Целью работы является получение практических навыков исследования корректности программ с помощью формальных методов.

2.3.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Разработать несколько типичных сценариев использования библиотеки, полученной в ходе лабораторной работы №1.

- Ознакомиться с принципами работы одного из бесплатных средств формальной верификации.

- Установить и настроить выбранное средство формальной верификации.

- Выполнить построение моделей, соответствующих сценариям использования библиотеки.

- Провести верификацию.

2.3.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы

(назначение и архитектуру, достоинства), листинг разработанной программы, описание нескольких типичных сценариев использования, обзорные сведения по используемому средству формальной верификации, описание моделей, описание хода верификации и выводы по проделанной работе.

2.4 Лабораторная работа №4. Профилирование и оптимизации Java-программ

Целью работы является получение практических навыков профилирования и оптимизации программ.

2.4.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести профилирование разработанных в ходе лабораторной работы №1 сценариев.
- Выявить «узкие» места в работе.
- Проанализировать код используемых функций.
- Выполнить оптимизацию кода.
- Сравнить результаты работы до и после оптимизаций.

2.4.2 Указания к выполнению работы. Профилирование необходимо выполнить с помощью одного из специализированных средств: `jconsole`, `hprof`, `jvisualvm`.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание библиотеки и сценариев, разработанных в ходе предыдущей лабораторной работы, листинг разработанных сценариев, результаты профилирования сценариев, описание выполненных оптимизаций, результаты оптимизаций.

2.5 Лабораторная работа №5. Разработка пользовательского интерфейса с использованием технологий javaFX

Целью работы является получение навыков разработки графического пользовательского интерфейса с применением технологий javaFX.

2.5.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать и согласовать с преподавателем задачу, для решения которой может быть использована библиотека, разработанная в ходе лабораторной работы №1.

- Разработать программу для решения выбранной задачи (взаимодействие с пользователем должно осуществляться с применением пользовательского интерфейса).

2.5.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, диаграммы, описывающие разработанный интерфейс, программный код, экранные формы, выводы по проделанной работе.

2.6 Лабораторная работа №6. Знакомство с инструментом автоматизации сборки проектов Apache Maven

Целью работы является знакомство с базовыми возможностями средства автоматизации сборки проектов Apache Maven.

2.6.1 Задание. Для выполнения лабораторной работы необходимо написать скрипт для автоматической сборки проекта, реализованного в ходе выполнения предыдущей лабораторной работы.

2.6.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, текст разработанного скрипта, выводы по проделанной работе.

Дополнительные сведения по Apache Maven могут быть найдены по ссылке <http://maven.apache.org/guides/index.html>.

2.7 Лабораторная работа №7. Декомпиляция приложения на языке программирования Java

Целью работы является получение практических навыков декомпиляции (преобразования байт-кода в исходный код) приложений на языке программирования Java.

2.7.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализ существующих программных средств декомпиляции байт-кода и выбрать одно из них.
- С помощью выбранного средства преобразовать class-файлы, полученные в ходе выполнения лабораторной работы №1, в файлы исходного кода.
- Провести исследование полученных файлов на предмет их соответствия исходным файлам.

2.7.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, краткий сравнительный анализ инструментов для декомпиляции байт-кода, описание процесса декомпиляции, полученный программный код, результаты сравнения изначального и полученного кода, выводы по проделанной работе.

2.8 Лабораторная работа №8. Защита Java-приложений с помощью обфускации

Целью работы является получение практических навыков обфускации приложений.

2.8.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализ существующих программных средств обфускации Java-программ и выбрать одно из них.
- С помощью выбранного средства выполнить обфускацию исходного кода, полученного в ходе выполнения лабораторной работы №1, и преобразовать его в байт-код.
- Выполнить декомпиляцию полученного байт-кода.
- Провести исследование полученных файлов на предмет их соответствия исходным файлам до обфускации.

2.8.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, краткий сравнительный анализ инструментов для обфускации Java-программ, описание процесса обфускации, полученный программный код, описание процесса декомпиляции, результаты сравнения изначального (до обфускации) и полученного кода, выводы по проделанной работе.

2.9 Лабораторная работа №9. Разработка приложения с использованием шаблонов проектирования

Целью работы является получение практических навыков разработки программ на языке программирования Java с применением шаблонов

проектирования, а также закрепление знаний, полученных на лекциях и предыдущих лабораторных работах.

2.9.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать задание в соответствии с целью работы. Согласовать тему разработки с преподавателем.
- Разработать структуру приложения.
- В процессе реализации применить шаблоны проектирования там, где это является целесообразным.

2.9.2 Указания к выполнению работы. Допускается связь разрабатываемого приложения с темой курсового проекта.

Отчет по лабораторной работе должен содержать титульный лист, задание, описание разработанной программы (назначение, структуру, достоинства, использованные в процессе разработки средства), листинг разработанной программы и выводы по проделанной работе.

2.10 Лабораторная работа №10. Автоматизация генерации UML-диаграмм

Целью работы является получение практических навыков использования средств, предназначенных для генерации UML-диаграмм по готовому программному коду.

2.10.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализа доступных средств, позволяющих выполнять генерацию диаграмм классов по готовому программному коду.
- Выбрать, установить и настроить одно из доступных средств.

- Выполнить генерацию диаграммы классов для приложения, разработанного в ходе предыдущей лабораторной работы.
- Отметить на полученной диаграмме классов использованные шаблоны проектирования. Сопроводить изображение необходимыми поясняющими комментариями.

2.10.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, краткий сравнительный анализ средств для автоматической генерации UML-диаграмм, листинг разработанной в ходе предыдущей лабораторной работы программы, сгенерированную диаграмму классов, описание выделенных шаблонов проектирования и выводы по проделанной работе.

2.11 Лабораторная работа №11. Разработка базы данных с применением CASE-средств

Целью работы является получение практических навыков использования CASE-средств, предназначенных для проектирования и создания баз данных.

2.11.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Выбрать предметную область и выполнить постановку задания на разработку, согласовав его с преподавателем.
- Провести анализа доступных CASE-средств, позволяющих выполнять проектирование и создание баз данных.
- Выбрать, установить и настроить одно из доступных средств.
- В соответствии с поставленным заданием спроектировать и создать базу данных.

2.11.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, постановку задачи, краткий сравнительный анализ CASE-средств для проектирования и создания баз данных, описание процесса проектирования базы данных и выводы по проделанной работе.

2.12 Лабораторная работа №12. Разработка приложения с применением CASE-средств

Целью работы является получение практических навыков использования CASE-средств, предназначенных для проектирования и создания приложений.

2.12.1 Задание. Для выполнения лабораторной работы необходимо решить следующие задачи.

- Провести анализа доступных CASE-средств, позволяющих выполнять проектирование и создание приложений.
- Выбрать, установить и настроить одно из доступных средств.
- В соответствии с поставленным в ходе предыдущей лабораторной работы заданием спроектировать и создать приложение, которое будет использовать полученную ранее базу данных.

2.12.2 Указания к выполнению работы. Отчет по лабораторной работе должен содержать титульный лист, задание, постановку задачи, краткий сравнительный анализ CASE-средств для проектирования и создания приложений, описание процесса проектирования, листинг разработанного приложения, экранные формы и выводы по проделанной работе.

3 Справочные сведения

В данном разделе представлена дополнительная информация о средствах рефакторинга, генерации документации и профилирования.

3.1 Встроенные средства рефакторинга среды разработки Eclipse

Рефакторингом называется процесс изменения структуры программы без изменения ее функциональности. Eclipse позволяет применять большое количество методов рефакторинга. Ниже рассмотрены некоторые из них.

Переименование (Rename). Позволяет изменить имя переменной, класса, пакета, интерфейса и т.д. Все ссылки на объект будут модифицированы.

Перемещение (Move). Используется для перемещения объектов (чаще всего классов из одного пакета в другой).

Выделение интерфейса (Extract interface). Позволяет выделить группу методов в интерфейс. Все классы, связанные с данной группой методов будут реализовывать выделенный интерфейс.

Выделение метода (Extract method). Позволяет выделить часть кода в отдельный метод.

Выделение локальной переменной (Extract local variable). Позволяет выделить выражение в локальную переменную. Выражение будет заменено на переменную.

Выделение константы (Extract constant). Аналогично выделению локальной переменной, но выделяет в static final константу неизменяемое выражение.

Введение (Inline...). Обратная к Extract операция.

Инкапсуляция полей (Encapsulate Field). Создает методы-аксессуары для выбранного поля и модифицирует ссылки.

3.2 Описание дескрипторов JavaDoc

Javadoc – генератор документации в HTML-формате из комментариев исходного кода на Java от Sun Microsystems. Javadoc – стандарт для документирования классов Java, предоставляющий API для создания доклетов и тэглетов, которые позволяют программисту анализировать структуру Java-приложения.

Комментарии JavaDoc выполняются следующим образом. После начальной комбинации символов `/**` располагается текст, являющийся главным описанием класса, переменной или метода. Далее разрешается вставлять различные дескрипторы. Каждый дескриптор `@` должен быть первым в своей строке. Несколько дескрипторов одного и того же типа необходимо группировать вместе. Встроенные дескрипторы (начинаются с фигурной скобки) можно помещать внутри любого описания.

Утилита `javadoc` в качестве входных данных принимает файл с исходным кодом программы и генерирует несколько файлов HTML, содержащих документацию по программе. Информация о каждом классе будет сосредоточена в отдельном HTML-файле. Кроме того, будет выполнено построение деревьев индексов и иерархии. Пример документирования представлен ниже.

```
/**
 * Умножает одну матрицу на другую.
 * @author GennadiyChistyakov
 * @param arg0 Первый множитель
 * @param arg1 Второй множитель
 * @return Результирующую матрицу или null, если умножение не возможно
 */
Matrix multiply(Matrix arg0, Matrix arg1) {
    ...
}
```

}

Наиболее распространенными являются следующие дескрипторы.

@author описание. Документирует автора класса. При вызове утилиты javadoc нужно задать опцию -author, чтобы включить поле в HTML документацию.

{@code фрагмент_кода}. Позволяет встраивать в комментарий текст (фрагмент кода). Этот текст будет отображаться с помощью шрифта кода без последующей обработки в HTML.

@deprecated описание. Определяет, что класс, интерфейс или член класса является устаревшим. Рекомендуются включать дескрипторы @see или {@link} для того, чтобы информировать программиста о доступных альтернативных вариантах. Может использоваться для документирования переменных, методов и классов.

{@docRoot}. Определяет путь к корневому каталогу текущей документации.

@exception имя_исключения пояснение. Описывает исключение для данного метода. Здесь имя_исключения указывает полное имя исключения, а пояснение представляет строку, которая описывает, в каких случаях может возникнуть данное исключение. Может использоваться только для документирования методов.

{@inheritDoc}. Наследует комментарий от непосредственного суперкласса.

{@link пакет.класс#элемент текст}. Встраивает ссылку на дополнительную информацию. При отображении текст (если присутствует) используется в качестве имени ссылки.

{@linkplain пакет.класс#элемент текст}. Встраивает ссылку. Ссылка отображается шрифтом основного текста.

{@literal описание}. Позволяет встраивать текст в комментарий. Этот текст отображается «как есть» без последующей обработки HTML.

@param имя_параметра пояснение. Документирует параметр для метода или параметр-тип для класса или интерфейса. Может использоваться только для документирования метода, конструктора, обобщенного класса или интерфейса.

@return пояснение. Описывает возвращаемое значение метода.

@see ссылка, @see пакет.класс#элемент текст. Обеспечивает ссылку на дополнительную информацию.

@serial описание. Определяет комментарий для поля, сериализуемого по умолчанию.

@serialData описание. Документирует данные, записанные с помощью методов `writeObject` и `writeExternal`.

@serialField имя тип описание. Для класса, реализующего `Serializable`, дескриптор обеспечивает комментарии для компонента `ObjectStreamField`. Здесь имя представляет имя поля, тип представляет его тип, а описание – комментарий для данного поля.

@since выпуск. Показывает, что класс или элемент класса был впервые представлены в определенном выпуске. Здесь выпуск представляет строку, в которой указан выпуск или версия, начиная с которого эта особенность стала доступной.

@throws имя_исключения пояснение. Имеет то же назначение, что и дескриптор `@exception`.

{@value}. Отображает значение следующей за ним константы, которой должно являться поле `static`.

{@value пакет.класс#поле}. Отображает значение определенного поля `static`.

@version информация. Представляет информацию о версии (как правило, номер). При выполнении утилиты `javadoc` нужно указать опцию `-version`, чтобы этот дескриптор включить в HTML документацию.

Для документирования переменной можно использовать следующие дескрипторы: `@see`, `@serial`, `@serialField`, `{@value}`, `@deprecated`.

Для классов и интерфейсов можно использовать дескрипторы: `@see`, `@author`, `@deprecated`, `@param`, `@version`.

Методы можно документировать с помощью дескрипторов: `@see`, `@return`, `@param`, `@deprecated`, `@throws`, `@serialData`, `{@inheritDoc}`, `@exception`.

Дескрипторы `{@link}`, `{@docRoot}`, `{@code}`, `{@literal}`, `@since`, `{@linkplain}` могут применяться повсеместно.

3.3 Краткие сведения по работе со средством профилирования hprof

Для профилирования с применением средства hprof необходимо осуществить запуск программы из командной строки с помощью команды вида

```
java -agentlib:hprof=[<option>=<value>, ...] someClass
```

Дополнительная информация по возможностям работы с hprof может быть получена с помощью команды

```
java -agentlib:hprof=help
```

Список основных опций представлен в таблице 1.

Таблица 1 – Основные опции hprof

Опция	Описание	Значение по умолчанию
heap=dump sites all	Профилирование кучи	all
cpu=samples times old	Использование процессора	off
monitor=y n	---	n
format=a b	Текстовый или двоичный формат вывода	A

file=<file>	Файл, в который будет осуществлен вывод	java.hprof[{.txt}]
net=<host>:<port>	Послать данные через сокет на удаленный хост	off
depth=<size>	Глубина трассы стека	4
interval=<ms>	Интервал снимков	10
cutoff=<value>	---	0.0001
lineno=y n	Вывод номеров линий трассы	y
thread=y n	Вывод информации о потоках трассы	n
doe=y n	Дамп при выходе	y
msa=y n	---	n
force=y n	---	y
verbose=y n	Печать сообщений о дампе	y

Пример параметров запуска, которые позволяют выполнять сэмплированный анализ с частотой снимков двадцать миллисекунд и глубиной стека вызова три

```
java -agentlib:hprof=cpu=samples,interval=20,depth=3 classname
```

Пример параметров запуска, которые позволяют оценивать объем затраченной памяти из кучи с помощью анализа ссылок

```
java -agentlib:hprof=heap=sites classname
```

Следует заметить, что опция format=b не может быть использована совместно с опциями monitor=y или cpu=old|times.

Заключение

Автор надеется, что предлагаемый курс помог студентам улучшить навыки программирования и освоить часть программных средств, входящих в стек современных технологий разработки ПО.

Отзывы, пожелания и замечания принимаются на электронной почте gennadiychistyakov@gmail.com.

Список литературы

1. Шилдт, Г. Полный справочник по Java SE 6 Edition / Г. Шилдт // М.: «Вильямс», 2007. 1040 с.
2. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес // СПб: «Питер», 2007. 366 с.
3. Git-book [Электронный ресурс]. Режим доступа: <https://git-scm.com/book/en/v2> (дата обращения 20.08.2016).
4. Управление версиями в subversion [Электронный ресурс]. Режим доступа: <http://svnbook.red-bean.com/> (дата обращения 20.08.2016).
5. The Apache Ant project [Электронный ресурс]. Режим доступа: <http://ant.apache.org/> (дата обращения 20.08.2016).
6. Apache Maven project [Электронный ресурс]. Режим доступа: <https://maven.apache.org/> (дата обращения 20.08.2016).