

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Методические указания к лабораторным работам
и организации самостоятельной работы для студентов
заочной формы обучения направления «Программная инженерия»
(уровень бакалавриата)

Безходарнов Илья Владимирович

Конструирование программного обеспечения: методические указания к лабораторным работам и организации самостоятельной работы для студентов заочной формы обучения направления «Программная инженерия» (уровень бакалавриата) / И.В. Безходарнов. – Томск: ТУСУР, 2018. – 14 с.

Содержание

Введение	4
1 Методические указания к проведению лабораторных работ	6
1.1 Лабораторная работа «Конструирование бизнес-логики и архитектуры простого приложения с применением объектно-ориентированного проектирования»	6
1.2 Лабораторная работа «Применение шаблонов проектирования программного обеспечения»	7
2 Методические указания по организации самостоятельной работы	9
2.1 Общие положения	9
2.2 Проработка лекционного материала, подготовка к контрольным и лабораторным работам	9
2.3 Самостоятельное изучение тем теоретической части курса	10
2.4. Выполнение контрольных работ	13
Рекомендуемая литература	14

ВВЕДЕНИЕ

Цели и задачи дисциплины

Целью дисциплины «Конструирование программного обеспечения (ПО)» является формирование у студентов знаний по методам, инструментам и процессам конструирования надежно, устойчивого и эффективного ПО для средств вычислительной техники автоматизированных и автоматических систем в рамках современных технологий разработки.

В рамках дисциплины изучается процесс конструирования программных систем, распространенные методики и практики построения надежного программного обеспечения. Дисциплина нацелена на формирование у студента навыков сбора, обработки и представления исходных данных для принятия проектных решений; разработки концептуальных, информационно-логических и функциональных моделей программного обеспечения; объектно-ориентированного анализа и проектирования.

Основными задачами преподавания дисциплины являются:

- изучение методов проектирования программных средств с использованием средств автоматизации проектирования;
- изучение современных инструментальных средств для разработки ПО;
- изучение стандартов по процессам разработки, методам контроля и оценки качества ПО на всех этапах его жизненного цикла;
- изучение эвристических принципов конструирования ПО;
- изучение методов конструирования программ, устойчивых к собственным ошибкам и ошибкам данных,
- изучение принципов верификации и отладки сконструированного ПО;
- изучение методов математического моделирования процессов и объектов для создания эффективной среды отладки сконструированного ПО.

По окончании изучения дисциплины «Конструирование программного обеспечения» студент должен:

знать:

- методы конструирования программного обеспечения;

- методы разработки программных средств для решения практических задач;
- методы разработки компонентов программных комплексов с использованием современных программных средств и технологий разработки алгоритмов и программ, методы отладки;
- методы математического анализа и моделирования, теоретического и экспериментального исследования;

уметь:

- решать задачи, возникающие на различных фазах жизненного цикла программных систем, связанных с конструированием программного обеспечения;
- разрабатывать качественные и гибкие программные системы;
- проводить рефакторинг программных систем;
- обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности;
- использовать нормативно-технические документы;

владеть:

- навыками конструирования программного обеспечения;
- навыками оценки качества конструирования программных систем;
- навыками работы с компьютером как средством проектирования и отладки ПО и управления информацией.

1 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

1.1 Лабораторная работа «Конструирование бизнес-логики и архитектуры простого приложения с применением объектно-ориентированного проектирования»

Цель работы:

Показать умение применять разные подходы для решения задач создания ПО, улучшить навыки.

Форма проведения:

Самостоятельное выполнение задания.

Форма отчетности:

Исходный код, работающая программа, защита отчета по лабораторной работе.

Продолжительность выполнения работы:

12 академических часов

Порядок выполнения работы:

В процессе выполнения работы нужно будет написать две программы. Первая (упрощенная) делается в процедурной парадигме программирования, вторая делается в объектно-ориентированной парадигме программирования.

Варианты:

Используя один из языков программирования: C++, Java, Python нужно написать текстовый редактор для Windows или Linux.

В качестве поля для отображения символов нужно взять графическую поверхность (т.е. фактически пиксельный объект) и весь вывод производить на него. Можно использовать любой шрифт. Не обязательно делать символы разной ширины (можно отображать их в фиксированной сетке).

Программа 1 (процедурная парадигма):

Делаем однострочный вариант редактора. Он должен уметь: отображать курсор (можно нарисовать свой), вводить символы, удалять символы клавишами Backspace и Delete, перемещаться по тексту стрелками влево и вправо, позиционировать курсор с помощью мышки и с помощью клавиш Home и End.

Программа 2 (объектно-ориентированная парадигма):

Делаем многострочный редактор, который должен уметь делать все то, что и программа 1, и плюс к этому: работать с неограниченным количеством строк текста, перемещаться по тексту клавишами вверх, вниз, PgUp, PgDn, Ctrl+Home, Ctrl+End. Правильно обрабатывать переводы строк и выполнять перевод строк в случае их выхода за область редактирования. Выполнять изменение области редактирования, при этом правильно перерисовывая текст.

Отчет по лабораторной работе должен включать ссылки на исходный код обеих программ, диаграмму классов для второй программы, описание работы обеих версий программы.

1.2 Лабораторная работа «Применение шаблонов проектирования программного обеспечения»

Цель работы:

Изучение и получение практических навыков применения шаблонов проектирования.

Форма проведения:

Самостоятельное выполнение задания.

Форма отчетности:

Исходный код, работающая программа, защита отчета по лабораторной работе.

Продолжительность выполнения работы:

4 академических часа.

Порядок выполнения работы:

Нужно доработать программу, сделанную на этапе выполнения предыдущей лабораторной работы.

Варианты:

1. Реализовать механизм «отката последнего действия» с помощью шаблона проектирования, выбрать который нужно самостоятельно.

2. Реализовать механизм аварийного сохранения файла, выбрав для этого подходящий шаблон проектирования.

Отчет по лабораторной работе должен содержать диаграмму классов, ссылку на исходный код и словесное описание работы, а также обоснование почему был выбран тот или иной шаблон проектирования.

2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

2.1 Общие положения

Целями самостоятельной работы являются систематизация, расширение и закрепление теоретических знаний.

Самостоятельная работа студента по дисциплине «Конструирование программного обеспечения» для студентов заочной формы обучения включает следующие виды активности:

- 1) проработка лекционного материала;
- 2) подготовка к лабораторным работам;
- 3) оформление отчетов по лабораторным работам;
- 4) изучение тем теоретической части дисциплины, вынесенных для самостоятельной проработки;
- 5) выполнение контрольной работы.

2.2 Проработка лекционного материала, подготовка к контрольным и лабораторным работам

Проработка лекционного курса является одной из важных активных форм самостоятельной работы. Этот вид самостоятельной работы может быть организован следующим образом:

прочтение конспекта лекции с сопоставлением ваших записи с информацией, представленной на слайдах лекции;

изучение дополнительных учебных материалов, рекомендованных преподавателем;

проработка ответов на контрольные вопросы, которыми, как правило, заканчиваются разделы учебных пособий или учебников;

формулирование вопросов для преподавателя по темам, которые вызывают затруднения.

Не пренебрегайте ведением конспекта лекций. Лекции по дисциплине, как правило, проводятся с использованием слайдов. Но это не означает, что лекцию можно просто слушать. Ведение конспектов значительно повышает качество последующей проработки лекционного материала.

Одной из форм мониторинга уровня знаний по дисциплине являются контрольные работы. Во время изучения дисциплины проводятся контрольные работы двух типов: тестовые опросы на лекции и контрольные работы, в которых студентам необходимо применить полученные знания на практике.

Самостоятельная работа по подготовке к лабораторным работам по дисциплине состоит в изучении методических материалов по темам соответствующих видов аудиторных занятий. Рекомендуется перед выполнением лабораторной работы изучить лекционный и методический материал по теме лабораторной работы. Если при подготовке к занятию остались нерешенные вопросы, обратитесь за консультацией к преподавателю.

2.3 Самостоятельное изучение тем теоретической части курса

Самостоятельное изучение тем теоретической части курса направлено на систематизацию, расширение и закрепление теоретических знаний, полученных в процессе лекционных занятий, а также приобретение навыков практической работы и исследовательской деятельности.

Для самостоятельной проработки предлагаются следующие темы с рекомендацией соответствующей литературы.

Тема Объектно-ориентированное программирование

Перечень вопросов, подлежащих изучению

1. Основы объектно-ориентированного программирования
2. Реализация механизмов ООП в популярных языках программирования
3. Основы объектно-ориентированного проектирования

Рекомендуемые источники

1. Обзор технологии [Электронный ресурс]. – URL: https://ru.wikibooks.org/wiki/Объектно-ориентированное_программирование

2. Виртуальная таблица функций [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Таблица_виртуальных_методов

3. Статья ООП с примерами (часть 1) [Электронный ресурс]. – URL: <https://habrahabr.ru/post/87119/>
4. Статья ООП с примерами (часть 2) [Электронный ресурс]. – URL: <https://habrahabr.ru/post/87205/>
5. Язык программирования C++, обзор [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/C%2B%2B>

Тема «Технологии создания ПО»

Перечень вопросов, подлежащих изучению

1. Разработка приложений для Windows
2. Разработка кроссплатформенных приложений
3. Разработка мобильных приложений
4. Веб разработка

Рекомендуемые источники:

1. Разработка классических приложений на C++ для Windows [Электронный ресурс]. – URL: <https://developer.microsoft.com/ru-ru/windows/desktop/develop>
2. Язык программирования C# [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/C_Sharp
3. Технология разработки ПО .NET [Электронный ресурс]. – URL: <https://www.microsoft.com/net/>, https://ru.wikipedia.org/wiki/.NET_Framework
4. Язык программирования Python [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Python> <https://www.python.org/>
5. Язык программирования Java [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Java> <https://java.com/ru/>
6. Среда разработки Delphi [Электронный ресурс]. – URL: <https://www.embarcadero.com/products/delphi>, [https://ru.wikipedia.org/wiki/Delphi_\(среда_разработки\)](https://ru.wikipedia.org/wiki/Delphi_(среда_разработки))

7. Библиотека для создания интерфейсов Qt [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Qt> <https://www.qt.io/>
8. Библиотека для создания интерфейсов Electron [Электронный ресурс]. – URL: <https://electronjs.org/> <https://ru.wikipedia.org/wiki/Electron>
9. Разработка для платформы Android [Электронный ресурс]. – URL: <https://developer.android.com/index.html>
10. Язык программирования Kotlin [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Kotlin>
11. Разработка для платформы iOS [Электронный ресурс]. – URL: <https://developer.apple.com/documentation/>
12. Язык программирования [Электронный ресурс]. – URL: Objective-C <https://ru.wikipedia.org/wiki/Objective-C>
13. Язык программирования Swift [Электронный ресурс]. – URL: <https://developer.apple.com/swift/>,
[https://ru.wikipedia.org/wiki/Swift_\(язык_программирования\)](https://ru.wikipedia.org/wiki/Swift_(язык_программирования))
14. HTML [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/HTML>,
<https://www.w3schools.com/html/>
15. CSS [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/CSS>, <https://www.w3schools.com/css/>
16. JavaScript [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/JavaScript>,
<https://www.w3schools.com/js/>
17. Стек технологий LAMP [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/LAMP>,
<https://ru.wikipedia.org/wiki/Linux>,
https://ru.wikipedia.org/wiki/Apache_HTTP_Server,
<https://ru.wikipedia.org/wiki/MySQL>, <http://php.net/>,
<https://ru.wikipedia.org/wiki/Perl>

18. Технология ВЕБ разработки Django [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Django>,
<https://www.djangoproject.com/foundation/>

19. Документы, регламентирующие протоколы RFC [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/RFC>

20. ОС Windows [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Windows>

21. Архитектура [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Архитектура_Windows_NT,
<http://www.xnets.ru/plugins/content/content.php?content.110>

Права доступа: [https://msdn.microsoft.com/ru-ru/library/dn408189\(v=ws.11\).aspx](https://msdn.microsoft.com/ru-ru/library/dn408189(v=ws.11).aspx)

2.4 Выполнение контрольных работ

Цель выполнения контрольной работы — приобретение умений и навыков применять полученные теоретические знания для решения прикладных задач. Контрольная работа выполняется студентами самостоятельно с использованием рекомендованных разделов лекционного курса и литературных источников.

Контрольная работы № 1: Шаблоны проектирования

Контрольная работа проводится по материалам раздела конструирование ПО: объектно-ориентированное проектирование. Материалы для изучения указаны в рекомендуемых для самостоятельного изучения источниках и проходятся на лекциях.

Вариант 1

1. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Фабрика».

2. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Адаптер».

3. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Мост».

Вариант 2

1. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Прототип».

2. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Декоратор».

3. Приведите описание, диаграмму классов, пример использования на практике для шаблона проектирования «Одиночка».

Рекомендуемая литература

1. Логинова Ф.С. Объектно-ориентированные методы программирования: учеб. пособие / Ф.С. Логинова. – СПб: ИЭО СПбУТУиЭ, 2012. – 208 с. [Электронный ресурс]. – URL: <https://e.lanbook.com/book/64040>

2. Непейвода Н.Н. Стили и методы программирования: учеб. пособие / Н.Н. Непейвода. – М.; ИНТУИТ, 2016. – 295 с. [Электронный ресурс]. – URL: <https://e.lanbook.com/book/100512>

3. Литвиненко Н.А. Технология программирования на C++. Win32 API-приложения: учеб. пособие / Н.А. Литвиненко. – СПб: БХВ-Петербург, 2010. – 280 с. [Электронный ресурс]. – URL: <http://znanium.com/catalog/product/351463>

4. Розенберг Д. Применение объектного моделирования с использованием UML и анализ прецедентов / Д. Розенберг, К. Скотт; пер. с англ. – М.: ДМК Пресс, 2007. – 160 с. [Электронный ресурс]. – URL: <http://znanium.com/catalog/product/407658>

5. Дейл Н. Программирование на C++ / Н. Дейл, Ч. Уимз, М. Хедингтон; Пер. с англ. – М.: ДМК Пресс, 2007. – 672 с. [Электронный ресурс]. – URL:

<http://znanium.com/catalog/product/407353>