

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)

МЕТОДЫ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

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

Безходарнов Илья Владимирович

Методы и технологии программирования: методические указания к лабораторным работам и организации самостоятельной работы для студентов направления «Программная инженерия» (уровень бакалавриата) / И.В. Безходарнов. – Томск, 2018. – 22 с.

© Томский государственный университет
систем управления и радиоэлектроники, 2018
© Безходарнов И.В., 2018

Содержание

Введение	4
1 Методические указания к проведению лабораторных работ	6
1.1 Лабораторная работа «Анализ проекта/идеи: изучение методики оценки»	6
1.2 Лабораторная работа «Прототипирование интерфейсов» ...	7
1.3 Лабораторная работа «Задачи по программированию с использованием разных парадигм программирования»	8
1.4 Лабораторная работа «Анализ исходного кода»	9
1.5 Лабораторная работа «WEB-frontend»	10
1.6 Лабораторная работа «Небольшое GUI-приложение»	12
1.7 Лабораторная работа «Автоматизированное функциональное тестирование Windows-приложений»	13
1.8 Лабораторная работа «Автоматизированное функциональное тестирование WEB-сайтов»	15
1.9 Лабораторная работа «Continuous Integration – работа с системой автоматизации рутинных задач Jenkins CI»	16
2 Методические указания по организации самостоятельной работы	17
2.1 Общие положения	17
2.2 Проработка лекционного материала, подготовка к контрольным и лабораторным работам	17
2.3 Самостоятельное изучение тем теоретической части курса	18
2.4. Выполнение контрольных работ	21
Рекомендуемая литература	22

ВВЕДЕНИЕ

Цели и задачи дисциплины

Целями дисциплины являются:

1) расширение кругозора технических знаний студентов о различных видах ПО, методах их проектирования, создания и эксплуатации;

2) получение навыков использования различных методов проектирования и создания разных видов ПО;

3) знакомство с технологиями проектирования, создания и эксплуатации ПО. Получение практических навыков их использования.

Задачами преподавания дисциплины являются:

- ознакомление студентов с различными направлениями в отрасли ПО (Windows, Linux, WEB, сетевое и системное программирование, создание пользовательских интерфейсов, автоматизированное тестирование ПО, защита ПО от взлома);

- изучение и получение навыков использования нескольких методологий создания ПО (процедурное программирование, ООП, шаблоны проектирования, графическое проектирование).

- изучение и получение навыков использования технологий, используемых на всем протяжении жизненного цикла ПО (методы оценки и экспертизы проекта, общее представление о жизненном цикле ПО, методы и инструменты автоматизации процессов жизненного цикла ПО);

- получение общего представления о процессах жизненного цикла ПО.

В результате изучения дисциплины **студент должен:**

знать:

- различные методы проектирования, создания и эксплуатации программного обеспечения;

уметь:

- использовать на практике различные методы и технологии проектирования и создания разных видов ПО;

владеть:

- навыками создания ПО для Windows, Linux, WEB;

- навыками применения автоматизированного тестирования;

- навыками проектирования и создания пользовательских интерфейсов;
- навыками создания ПО в процедурной парадигме программирования и парадигме ООП;
- методами оценки и экспертизы проектов;
- методами и инструментами автоматизации процессов жизненного цикла ПО.

1 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

1.1. Лабораторная работа «Анализ проекта/идеи: изучение методики оценки»

Цель

Изучить и научиться применять на практике одну из методик оценки ИТ-проекта с точки зрения его перспективности для разработки. По сути – оценки бизнес-идеи проекта.

Форма проведения

Групповое выполнение заданий на занятии.

Форма отчетности

Выступление перед аудиторией

Методика проведения оценки

Проблемы Топ 3 проблемы	Решения Топ 3 <u>фиши</u>	Уникальное предложение ценности Простое понятное	Фора (нечестное преимущество) Нельзя купить или быстро скопировать	Сегменты заказчиков
	Метрики оценки продукта Метрики для оценки успешности продукта	сообщение о том, почему нужно выбрать именно вас	Каналы продаж Путь к вашим заказчикам	Целевые заказчики или пользователи
Структура затрат Фиксированные издержки Хостинг Зарплаты И т.п.		Потоки прибыли Модели монетизации Прибыль Способы заработка		

Порядок выполнения лабораторной работы (ЛР)

1. В соответствии с методикой, изложенной на лекции оценить один из предложенных вариантов ИТ-проекта.

2. По той же методике оценить собственную идею ИТ-проекта, либо уже существующую, но не из предложенных для п.1 вариантов.

Для зачета представить результаты оценки двух проектов в письменном виде.

Варианты заданий:

1. 1С-бухгалтерий
2. google.com
3. Microsoft Office
4. Navitel
5. 2gis
6. Skype
7. amazon cloud hosting
8. amazon.com

1.2 Лабораторная работа «Прототипирование интерфейсов»

Цель: изучение и применение на практике упрощенной методики проектирования GUI интерфейсов.

Форма проведения:

Групповое выполнение задания на лабораторном занятии.

Форма отчетности:

Защита результата выполнения преподавателю.

Задание:

1) проработать логику и внешний вид графического пользовательского интерфейса для одного из предложенных вариантов без использования средств автоматизации;

2) найти в интернете любое подходящее бесплатное решение для прототипирования интерфейсов, прорисовать в нем разработанный вариант.

Для зачета представить: электронный и/или печатный вариант макетов, разработанного интерфейса, сделанных в системе прототипирования.

Варианты заданий

1. Разработать интерфейсное решение, для связывания между собой удаленного ВЕБ-сервера и локального приложения. Представить интерфейс со стороны ВЕБ сервера и со стороны приложения.

2. Разработать интерфейсное решение – ВЕБ-форма, для регистрации пользователя на ВЕБ-сайте с одновременным представлением его фотографии (загрузкой фотографии на сервер).

3. Разработать интерфейсное решение, для управления удаленной, автоматизированной видео-камерой, имеющей функции – Zoom, повороты по горизонтали и вертикали, ночной режим съемки, автоматическое определение движения.

1.3 Лабораторная работа «Задачи по программированию с использованием разных парадигм программирования»

Цель работы: показать знания и умение писать простые программы, используя разные парадигмы программирования (процедурную и ООП). Выбрать язык программирования для выполнения следующих лабораторных работ.

Форма проведения:

Индивидуальное (или групповое) самостоятельное выполнение заданий.

Форма отчетности:

Исходные коды программ, работающие приложения, отчет по лабораторной работе.

Порядок выполнения работы

1) определиться с языком программирования, который будет использован в данной и следующих лабораторных работах, установить среду разработки, продемонстрировать умение ее использовать;

2) выполнить указанные преподавателем варианты задач;

3) каждая задача должна быть реализована в двух парадигмах программирования: процедурной и ООП;

4) для сдачи представить исходный код, работающие программы, диаграмму классов.

Варианты:

1. Написать программу, которая решает линейные и квадратные алгебраические уравнения.

2. Простейший калькулятор. Достаточно реализовать сложение, вычитание, умножение и деление.

3. Игра «Крестики Нолики».

4. Простейший минификатор java-скриптов внутри HTML страниц. На вход программе подается имя файла, в котором находится HTML страница с произвольным количеством java-скриптов. На выходе должен получиться новый файл на диске, в котором все java-скрипты минифицированы (достаточно сделать удаление лишних пробелов и переводов строк).

5. Программа для считывания свойств jpeg файла. На вход подается имя файла на диске. Нужно разобрать его внутреннюю структуру и вывести информацию о ней на экран.

6. Простейший архиватор – достаточно реализовать объединение нескольких файлов в один и распаковку их обратно (без сжатия).

1.4 Лабораторная работа «Анализ исходного кода»

Цель работы:

Изучить исходные коды хорошего качества, получить представление о вариантах решения конкретных задач, научиться понимать чужой исходный код.

Форма проведения:

Самостоятельное выполнение задания.

Форма отчетности:

Защита выполненного задания перед преподавателем, отчет по лабораторной работе.

Порядок выполнения работы:

Нужно разобрать исходный код чужой программы. Объем разбираемого кода: 1–2 тыс. строк, 3–5 классов или 5–7 функций.

Разобрав код нужно представить:

- 1) диаграмму (классов если это ООП, функций если это процедурное программирование);
- 2) описание программы на естественном языке;
- 3) дополнительные диаграммы если сочтете нужным.

Варианты:

C++

<https://github.com/shayanraouf/Movie-Rental-Inventory-App/archive/master.zip>

<https://github.com/vshwartskop/internship/archive/master.zip>

Java:

<https://github.com/JodaOrg/joda-time/blob/master/src/main/java/org/joda/time/DateTime.java> - библиотека большая, достаточно разобрать несколько классов

C++, C#:

<https://code.msdn.microsoft.com/> - библиотека примеров от независимых разработчиков

C:

<http://www.gnu.org/software/coreutils/coreutils.html>

Можно взять любой другой код, выполнив требования: код должен быть не ваш, объем кода должен быть 1–2 тысячи строк и содержать 3–5 классов или 5–7 функций.

1.5 Лабораторная работа «ВЕБ-frontend»

Цель работы:

Изучение и получение навыков практического применения создания активных WEB-страниц.

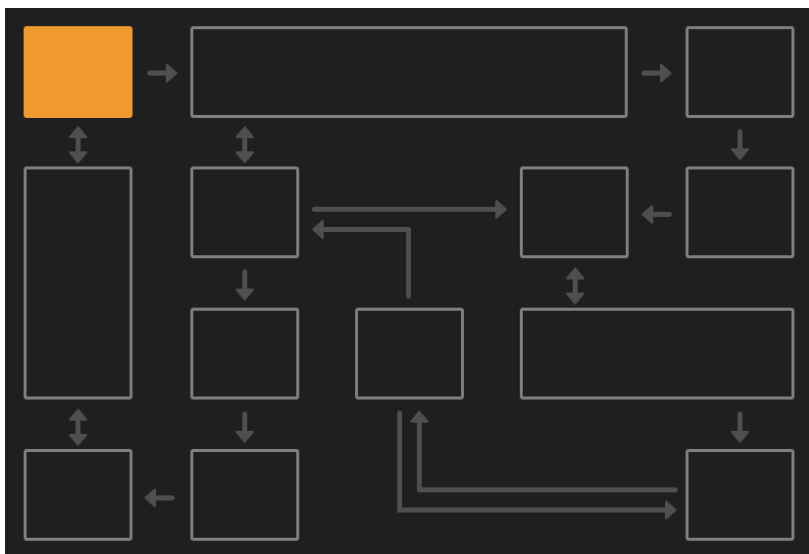
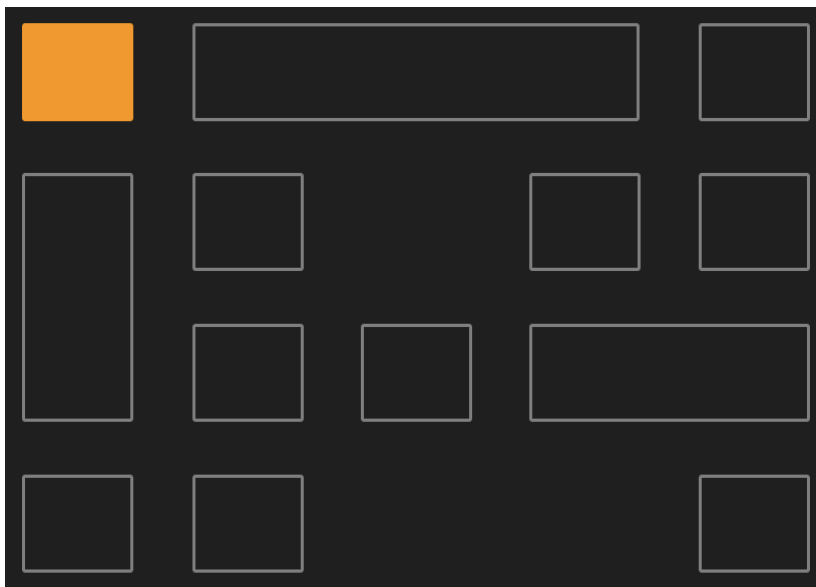
Форма проведения:

Самостоятельное выполнение задания.

Форма отчетности:

Исходный код, работающая страница, отчет по лабораторной работе.

Порядок выполнения работы:
Необходимо сверстать интерфейс по макетам:



Закрашенный прямоугольник – это компонент или виджет или кнопка или прямоугольник в фокусе. При нажатии на него в консоль разработчика должно выводиться сообщение вида: «Click at rect 42» где 42 это произвольные данные из data атрибута ноды и они могут быть уникальны для каждого прямоугольника.

Далее нужно реализовать логику до любого уровня по нарастающей (см. варианты).

Нужно реализовать любые два из четырех вариантов.

Можно использовать любые фреймворки и библиотеки.

Сопроводить решение обоснованием выбора инструментов.

Тестирование решения будет проходить в браузере Chrome актуальной версии.

Варианты:

1. Макет 1- фокус перемещается между прямоугольниками при наведении курсора мыши.

2. Макет 1 - фокус перемещается между прямоугольниками с помощью клавиш вверх/вправо/вниз/влево, а нажатие на прямоугольник происходит как по клику, так и по нажатию Enter с клавиатуры.

3. Макет 2 - фокус клавиатурой перемещается только в соответствии с указанными стрелками направлениями. Если стрелка однонаправленная, то переход в обратную сторону невозможен.

4. Макет 2 - фокус курсором мыши можно поставить только если есть путь по стрелкам от текущего фокусного прямоугольника до прямоугольника под курсором мыши.

1.6 Лабораторная работа «Небольшое GUI-приложение»

Цель

Улучшить практические навыки создания ПО с большим количеством взаимосвязей и бизнес-логики.

Форма проведения

Самостоятельное выполнение заданий.

Форма отчетности:

Исходный код программы, работающая программа, отчет по лабораторной работе.

Порядок выполнения работы

С использованием любого языка программирования разработать небольшой текстовый редактор. Условия:

1. Используется любой вариант графического пользовательского интерфейса

2. Для решения задачи следует использовать графическую поверхность (например, область окна программы), на которую с помощью функции рисования символов на графической поверхности можно побуквенно выводить текст.

3. Запрещено использование готовых компонентов, выполняющих указанную функциональность.

4. Нужно написать программу, реализующую многострочный ввод текста (т.е. многострочный текстовый редактор).

5. Редактор должен уметь:

a. Ввод текста (т.е. реакция на символьные клавиши)

b. Отрисовка курсора в текущей позиции

c. Удаление текста клавишами Del и Backspace

d. Перемещение курсора клавишами Left, Right, Up, Down, Home, End

6. Программа должна хранить данные (введенный текст) в памяти в любом виде и уметь сохранять их на диск, считывать с диска.

7. Допустимо не использовать прокрутку окна.

8. Строки, которые по длине не входят в окно должны переноситься на следующую строку.

Шаблон программы, от которого можно оттолкнуться (выполнен на языке C в среде разработки MS VisualStudio), размещен : <http://git.gpo2.tomsksoft.com/public-templates/editor-template>

1.7 Лабораторная работа «Автоматизированное функциональное тестирование Windows-приложений»

Цель

Изучение и получение навыков использования библиотеки для автоматизированного тестирования приложений.

Форма проведения

Самостоятельное выполнение задания.

Форма отчетности

Исходный код тестов, работающая система, отчет по ЛР.

Порядок выполнения работы

1. Используя MS VisualStudio, и ее инструмент управления пакетами nuget создать проект для автоматизированного тестирования, подключить к нему библиотеку TestStack на языке C#.

Возможно использование любого другого аналогичного по возможностям пакета на ваше усмотрение.

2. Создать два тест-кейса для тестирования программы калькулятор:

Тест-кейс 1:

Запустить приложение калькулятор

Эмулируя нажатие пользователем мышкой на кнопки калькулятора выполнить операцию: $1111 * 1111 =$

Считать полученный калькулятором результат, и проверить на правильность.

Тест-кейс 2:

Запустить приложение калькулятор

Выполнить операцию $2222 - 1111 =$

Считать полученный калькулятором результат, и сравнить его с заранее неверным ответом вызвать исключение.

Таким образом, тест-кейс 2 должен выдать ошибку.

3. Выполнить один из предложенных вариантов.

Варианты

1. Тестирование программы Notepad, шаги:

- a. Открыть программу
- b. Ввести любой текст длиной 10-100 символов
- c. Сохранить файл на диск
- d. Закрыть программу

2. Тестирование программы Notepad, шаги:

- a. Открыть программу
- b. Открыть текстовый файл с диска
- c. Скопировать содержимое файла в буфер обмена
- d. Добавить содержимое буфера обмена к файлу
- e. Сохранить файл на диск
- f. Закрыть программу

3. Тестирование программы Paint:

- a. Открыть программу
- b. Нарисовать любую фигуру
- c. Сохранить файл на диск
- d. Закрыть программу

1.8 Лабораторная работа «Автоматизированное функциональное тестирование WEB-сайтов»

Цель

Изучение и получение навыков применения технологии Selenium для автоматизированного тестирования WEB-сайтов.

Форма проведения

Самостоятельное выполнение задания.

Форма отчетности

Исходный код тестов, работающая система, отчет по лабораторной работе.

Порядок выполнения работы

1. Используя MS VisualStudio, и ее инструмент управления пакетами nuget создать проект для автоматизированного тестирования, подключить к нему библиотеку Selenium Web-driver. (По желанию, можно использовать любую другую среду/язык программирования совместимый с Selenium Web-driver).
2. Выполнить начальное задание:
 - 1) запустить браузер;
 - 2) выполнить навигацию на любой URL.
3. Выполнить один из предложенных вариантов.

Варианты

1. Поиск конкретных значений в выдаче поисковых сайтов:

- a. Выполнить поисковый запрос на поисковом сайте (например, «шариковые ручки»)
- b. Найти в выданных ответах конкретное значение (например, «Parke»), вывести номер позиции в которой находится это конкретное значение.

2. Тестирование функциональности выбранного самостоятельно сайта: выберите любой сайт, с достаточно разветвленной функциональностью и напишите для него 5 тест-кейсов, в которых будут присутствовать все перечисленные ниже случаи:

- i. Ввод текста в поле ввода
- ii. Нажатие на кнопку
- iii. Выбор элемента списка
- iv. Выбор элемента из радио-группы
- v. Поиск элемента на странице по имени класса
- vi. Поиск элемента на странице по типу тега

Все тест кейсы должны быть независимыми друг от друга (т.е. каждый должен выполняться независимо от результата выполнения других тестов).

1.9 Лабораторная работа «Continuous Integration – работа с системой автоматизации рутинных задач Jenkins CI»

Цель

Изучение и получение навыков работы с системой Jenkins CI.

Форма проведения

Самостоятельное выполнение задания.

Форма отчетности:

Защита результатов выполнения задания, отчет по лабораторной работе.

Порядок выполнения работы

1. Установить пакет Jenkins и настроить авторизированный вход.
2. Создать задачу, которая по расписанию (например, раз в 15 минут) выполняет архивирование заранее заданного каталога на диске.

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 «Методы и технологии программирования» 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](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: Objective-C [Электронный ресурс]. – URL: <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>

22. Права доступа [Электронный ресурс]. – URL: [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 работа «Шаблоны проектирования»

Контрольная работа проводится по разделу 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. – Санкт-Петербург: ИЭО СПбУТУиЭ, 2012. – 208 с. [Электронный ресурс]. – URL: <https://e.lanbook.com/book/64040>

2. Непейвода Н.Н. Стили и методы программирования [Электронный ресурс]: учеб. пособие / Н.Н. Непейвода.– М.: ИНТУИТ, 2016. – 295 с. – Режим доступа: <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>