Содержание

[Введение 4](#_Toc796322)

[1. Лабораторная работа «Знакомство с платформой .NET. Реализация консольных приложений» 5](#_Toc796323)

[1.1 Задание на лабораторную работу 5](#_Toc796324)

[1.2 Ход работы: 5](#_Toc796325)

[1.3 Контрольные вопросы 6](#_Toc796326)

[2. Лабораторная работа «Разработка сервера для клиент-серверного приложения с использованием платформы .NET» 7](#_Toc796327)

[2.1 Задание на лабораторную работу 7](#_Toc796328)

[2.2 Ход работы: 7](#_Toc796329)

[2.3 Контрольные вопросы 8](#_Toc796330)

[3. Лабораторная работа «Разработка клиентской части для клиент-серверного приложения с использованием платформы .NET» 9](#_Toc796331)

[3.1 Задание на лабораторную работу: 9](#_Toc796332)

[3.2 Ход работы: 10](#_Toc796333)

[3.3 Контрольные вопросы: 10](#_Toc796334)

[4. Лабораторная работа «Реализация взаимодействия между приложениями на основе технологии xml-rpc» 11](#_Toc796335)

[4.1 Задание на лабораторную работу: 11](#_Toc796336)

[4.2 Ход работы: 11](#_Toc796337)

[4.3 Контрольные вопросы. 11](#_Toc796338)

[5. Лабораторная работа «Проектирование SOA-архитектуры приложения» 13](#_Toc796339)

[5.1 Задание на лабораторную работу 13](#_Toc796340)

[5.2 Ход работы: 13](#_Toc796341)

[5.3 Контрольные вопросы: 13](#_Toc796342)

[6. Лабораторная работа «Создание и развёртывание web-сервисов» 15](#_Toc796343)

[6.1 Задание на лабораторную работу 15](#_Toc796344)

[6.2 Ход работы: 15](#_Toc796345)

[6.3 Контрольные вопросы. 15](#_Toc796346)

[7. Лабораторная работа «Использование com объектов» 17](#_Toc796347)

[7.1 Задание на лабораторную работу 17](#_Toc796348)

[7.2 Ход работы: 17](#_Toc796349)

[7.3 Контрольные вопросы 17](#_Toc796350)

[8. Лабораторная работа «Создание собственного com объекта.» 18](#_Toc796351)

[8.1 Задание на лабораторную работу 18](#_Toc796352)

[8.2 Ход работы 18](#_Toc796353)

[8.3 Контрольные вопросы 18](#_Toc796354)

# Введение

Данные методические указания содержат описание лабораторных работ направленных на закрепление теоретических знаний и приобретение первичных практических навыков в области проектирования и разработки архитектуры программных и информационных систем. Лабораторные работы выполняются студентами на языке C# в среде Microsoft Visual Studio или аналоге.

# Лабораторная работа «Знакомство с платформой .NET. Реализация консольных приложений»

## 1.1 Задание на лабораторную работу

Спроектировать модульную архитектуру и реализовать на языке программирования c# программу для решения следующих задач:

1. Вывести на экран аргументы, переданные в программу при запуске в командной строке.
2. Распечатать лет с 1900 по 2000. Рядом с каждым годом вывести слово «високосный» если этот год високосный или «не високосный» – если нет.
3. Вывести последовательность чисел Фибоначи до заданного числа.
4. Вычислить факториал заданного числа.
5. Вывести все простые числа не превышающие заданное. Для решения использовать алгоритм «решето Эратосфена».

Разработанная программа должна иметь единый консольный интерфейс. Каждая задача реализуется в отдельном модуле. Отчёт должен содержать листинг разработанной программы и описание модульной структуры программы со спецификацией каждого разработанного модуля.

## 1.2 Ход работы:

1. Ознакомиться с теоретическим материалом и изучить синтаксис языка программирования C#.
2. Спроектировать модульную структуру программы.
3. Разработать алгоритмы и реализовать модули для решения приведённых задач.
4. Реализовать и протестировать работу программы.
5. Написать отчёт.
6. Ответить на контрольные вопросы.

## 1.3 Контрольные вопросы

1. Что понимается под термином архитектура программной системы?
2. Перечислите основные разновидности архитектуры программных систем.
3. Какие требования предъявляются к проектированию модульной архитектуры программной системы?
4. Перечислите основные особенности языка программирования C# и платформы .NET.
5. Как происходит компиляция и выполнение программы на языке C#?

# Лабораторная работа «Разработка сервера для клиент-серверного приложения с использованием платформы .NET»

## 2.1 Задание на лабораторную работу

Спроектировать клиент-серверную архитектуру для приложения многопользовательского чата. Реализовать серверную часть приложения чата. Задачами сервера является подключение клиентов к чату, получение и рассылка сообщений клиентов между всеми подключёнными клиентами. Сервер должен организовать прослушивание заданного порта по которому к нему может подключаться несколько клиентов. Для обслуживания одновременно подключения от нескольких клиентов сервер должен использовать многопоточную архитектуру. Внутреннюю реализацию сервера рекомендуется сделать с использованием архитектурного шаблона MVC.

Отчёт по лабораторной работе должен содержать описание архитектуры клиент-серверного приложения с распределением функций между частями, описание архитектуры серверного приложения, листинг кода реализованного сервера, ответы на контрольные вопросы.

## 2.2 Ход работы:

1. Ознакомиться с лекционным материалом по построению клиент-серверных архитектур приложений.
2. Изучить возможности языка C# и платформы .NET по созданию клиент-серверных приложений.
3. Спроектировать архитектуру приложения чата с распределением функций и задач между клиентом и сервером.
4. Разработать архитектуру и алгоритмы работы серверной части приложения чата.
5. Реализовать серверную часть приложения чата.
6. Оформить отчёт по лабораторной работе.
7. Ответить на контрольные вопросы.

## 2.3 Контрольные вопросы

1. Перечислите основные разновидности клиент-серверной архитектуры.
2. Опишите стек протоколов используемых при организации клиент-серверной архитектуры.
3. Какие компоненты платформы .NET используются для прослушивания входящих подключений на определённом порте?
4. Что такое потоки в программировании?
5. Как реализуются потоки в языке C#?

# Лабораторная работа «Разработка клиентской части для клиент-серверного приложения с использованием платформы .NET»

## 3.1 Задание на лабораторную работу:

Разработать клиентскую часть для приложения многопользовательского чата. Основными задачами клиента является подключение к серверу чата по заданному адресу, отображение списка участников чата, отправка и получение сообщений от участников чата. При подключении к серверу клиент должен идентифицировать себя путём ввода логина. Проверку уникальности логинов и регистрацию пользователей чата реализовывать по желанию. После подключения пользователь должен видеть все сообщения от всех пользователей поступающие в чат. У пользователя должна быть возможность отправить собственное сообщение. Для клиентского приложения разработать графический интерфейс. Примерный вид клиента представлен на рисунке 1.

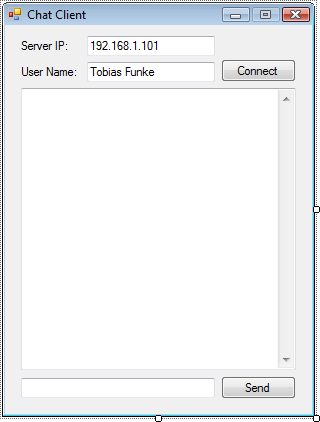


Рисунок 1 – Примерный вид интерфейса клиентского приложения чата.

## 3.2 Ход работы:

1. Ознакомиться с лекционным материалом по построению клиент-серверных архитектур приложений.
2. Изучить возможности языка C# и платформы .NET по созданию графического пользовательского интерфейса.
3. Спроектировать архитектуру и реализовать алгоритмы работы клиентского приложения.
4. Протестировать совместную работу клиентов и сервера.
5. Оформить отчёт по лабораторной работе.
6. Ответить на контрольные вопросы.

## 3.3 Контрольные вопросы:

1. Опишите механизм построения графического пользовательского интерфейса в платформе .NET.
2. Опишите принцип работы делегатов в языке C#.
3. Опишите механизм обновления пользовательского интерфейса из параллельного потока.

# Лабораторная работа «Реализация взаимодействия между приложениями на основе технологии xml-rpc»

## 4.1 Задание на лабораторную работу:

Разработать две программы – сервер и клиент. Клиент принимает от пользователя целочисленную квадратную матрицу и отсылает ее серверу. Сервер принимает целочисленную квадратную матрицу, после этого находит диагональ с минимальным элементом и заменяет элементы этой диагонали нулями, затем возводит в квадрат элементы стоящие ниже этой диагонали. На экран выводится исходная матрица, минимальный элемент диагоналей и результирующая матрица. Выполнение этого действия оформить как удаленную процедуру и воспользоваться механизмом RPC.

## 4.2 Ход работы:

1. Изучить лекционный материал по организации архитектуры на основе удалённого вызова процедур.
2. Разработать архитектуру для описанной в задании системы.
3. Разработать алгоритмы и реализовать серверную часть приложения.
4. Разработать алгоритмы и реализовать клиентскую часть приложения.
5. Протестировать взаимодействие клиента и сервера по средствам удалённого вызова процедур.
6. Оформить отчёт по лабораторной работе.
7. Ответить на контрольные вопросы.

## 4.3 Контрольные вопросы.

1. Опишите основные принципы архитектуры приложений, основанной на удалённом вызове процедур?
2. Опишите механизм работы удалённой процедуры?
3. Опишите структуру протокола XML-RPC.
4. Как реализуется механизм удалённого вызова процедур в платформе .NET?

# Лабораторная работа «Проектирование SOA-архитектуры приложения»

## 5.1 Задание на лабораторную работу

Разработать клиентское приложение, взаимодействующее с сервисом сети интернет на основе SOAP-протокола. В качестве сервера можно использовать любой сервис сети интернет, реализующий протокол SOAP для использования его функциональности (сервисы РЖД, Аэрофлота, сервисы погоды, банковские сервисы и т.п.).

## 5.2 Ход работы:

1. Изучить лекционный материал по организации SOAP архитектуры.
2. Выбрать сервис сети интернет и согласовать с преподавателем функционал приложения.
3. На основе WSDL описания проанализировать методы, предоставляемые выбранным сервисом.
4. Спроектировать и реализовать клиентское приложение, использующее предоставляемый сервисом функционал.
5. Оформить отчёт с описанием механизма взаимодействия разработанного клиента с сервисом.
6. Ответить на контрольные вопросы.

## 5.3 Контрольные вопросы:

1. Опишите принципы построения сервис-ориентированной архитектуры?
2. Опишите структуру протокола SOAP для взаимодействия с web-сервисами?
3. Опишите основные роли участников сервис-ориентированной архитектуры?
4. Каким образом происходит идентификация web-сервисов?
5. Что такое WSDL описание сервиса?
6. Опишите преимущества и недостатки сервис-ориентированной архитектуры?

# Лабораторная работа «Создание и развёртывание web-сервисов»

## 6.1 Задание на лабораторную работу

Разработать web-сервис, предоставляющий информацию о рейсах и наличии билетов на эти рейсы по запросам пользователей. Должна быть возможность просмотреть все рейсы по заданному маршруту, просмотреть наличие билетов, заказать билет на рейс.

Реализовать клиентское приложение для работы с данным сервисом. (Клиентское приложение и web-сервис могут быть реализованы на разных языках программирования). Для реализации web-сервиса рекомендуется использовать библиотеку WCF платформы .NET.

## 6.2 Ход работы:

1. Изучить теоретический материал по инструментарию, предоставляемому платформой .NET для создания web-сервисов.
2. Спроектировать архитектуру разрабатываемого приложения.
3. Реализовать сервер web-сервиса.
4. Развернуть сервис в локальной сети.
5. Разработать алгоритмы и реализовать клиент для взаимодействия с разработанным сервисом.
6. Протестировать взаимодействие клиента и сервиса.
7. Оформить отчёт с описанием разработанной архитектуры и компонентов приложений.
8. Ответить на контрольные вопросы.

## 6.3 Контрольные вопросы.

1. Какие возможности предоставляет платформа .NET для создания web-сервисов?
2. Опишите основные принципы работы платформы WCF.
3. Что такое контракт в терминологии WCF?
4. Опишите механизм построения клиентского приложения на основе описания web-сервиса?
5. Опишите принципы развертывания wcf-службы?

# Лабораторная работа «Использование com объектов»

## 7.1 Задание на лабораторную работу

Разработать приложение для автоматического формирования отчёта по лабораторной работе на основе заданного шаблона и введённых пользователем данных. Результатом работы программы должен быть документ word в который в заданные места на основе шаблона отчёта подставлены данные пользователя. Взаимодействие с word организовать через com объекты.

## 7.2 Ход работы:

1. Изучить лекционный материал по работе с com-объектами.
2. Изучить интерфейсы и структуру объектной модели, предоставляемой Microsoft Word для работы с документами.
3. Спроектировать архитектуру приложения.
4. Разработать шаблон отчёта по лабораторной работе.
5. Разработать алгоритмы и реализовать приложение для генерации отчёта по лабораторной работе.
6. Сгенерировать отчёт по лабораторной работе.

## 7.3 Контрольные вопросы

1. Что такое объект COM?
2. Чем отличаются СОМ объекты от обычных объектов?
3. Объясните основные принципы построения архитектуры приложения на основе com объектов?
4. Расскажите о порядке работы с com-объектами?
5. Опишите преимущества и недостатки архитектуры, основанной на com объектах?

# Лабораторная работа «Создание собственного com объекта.»

## 8.1 Задание на лабораторную работу

Создать COM сервер и клиент, использующий его методы, реализующий функции согласно варианту задания.

## 8.2 Ход работы

1. Получить у преподавателя описание функций, которые должны быть реализованы в соответствующем com-объекте.
2. Изучить теоретический материал по созданию com-объектов с применением платформы .NET.
3. Разработать алгоритмы и реализовать COM сервер.
4. Зарегистрировать созданный COM объект в системе.
5. Разработать алгоритмы и реализовать клиент, использующий созданный com объект.
6. Оформить отчёт по лабораторной работе.
7. Ответить на контрольные вопросы.

## 8.3 Контрольные вопросы

1. Опишите процедуру создания COM объекта?
2. Что такое интерфейс СОМ объекта, его назначение?
3. Назначение библиотеки СОМ. Механизм создания первого экземпляра объекта с помощью библиотеки СОМ и системного реестра.?
4. Что такое СОМ-сервер?