

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Галунин Сергей Александрович
Должность: проректор по учебной работе
Дата подписания: 28.06.2023 11:50:22
Уникальный программный ключ:
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП
«Компьютерное моделирование
и проектирование»



СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования
**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»**
(СПбГЭТУ «ЛЭТИ»)

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

для подготовки бакалавров

по направлению

09.03.01 «Информатика и вычислительная техника»

по профилю

«Компьютерное моделирование и проектирование»

Санкт-Петербург

2022

ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.т.н., доцент Новакова Н.Е.

Рабочая программа рассмотрена и одобрена на заседании кафедры САПР
21.12.2021, протокол № 7

Рабочая программа рассмотрена и одобрена учебно-методической комиссией
ФКТИ, 24.02.2022, протокол № 2

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	САПР
Общая трудоемкость (ЗЕТ)	4
Курс	2
Семестр	3
Виды занятий	
Лекции (академ. часов)	34
Лабораторные занятия (академ. часов)	17
Практические занятия (академ. часов)	17
Иная контактная работа (академ. часов)	3
Все контактные часы (академ. часов)	71
Самостоятельная работа, включая часы на контроль (академ. часов)	73
Всего (академ. часов)	144
Вид промежуточной аттестации	
Дифф. зачет (курс)	2
Курсовая работа (курс)	2

2 АННОТАЦИЯ ДИСЦИПЛИНЫ

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

Рассматриваются основы синтаксиса языка C# и обзор платформы Microsoft .Net: типы-значения, ссылочные типы, выражения, исключения, методы, параметры, классы и объекты. Дисциплина посвящена изучению принципов объектно-ориентированного программирования: использование инкапсуляции, механизм наследования, полиморфизм. Создание и разрушение объектов. Использование конструктора. Наследование. Производные классы, реализация методов, изолированные классы, использование интерфейсов, абстрактные классы. Агрегирование. Применение вложенных классов. Использование агрегирования. Использование пространства имен. Использование модулей и сборок. Пространства имен и область видимости. Операции, делегаты и события. Обзор операций. Перегрузка операций. Создание и использование делегатов. Определение и использование событий. Свойства и индексаторы. Атрибуты.

SUBJECT SUMMARY

«OBJECT ORIENTED PROGRAMMING»

Fundamentals of the C # syntax and an overview of the Microsoft .Net platform are considered: value types, reference types, expressions, exceptions, methods, parameters, classes and objects. The discipline is devoted to the study of the principles of object-oriented programming: us the use of encapsulation, inheritance, polymorphism. The creation and destruction of objects. Using the constructor. Derived classes, the implementation of methods, isolated classes, use interfaces, abstract classes. Aggregation. The use of nested classes. Using aggregation. The use of namespaces. Using modules and assemblies. Namespaces and scope. Operations, delegates and events. Review of Operations. Operator Overloading. Creating and using delegates. Defining and using events. Properties and indexers. Attributes.

3 ОБЩИЕ ПОЛОЖЕНИЯ

3.1 Цели и задачи дисциплины

1. Целью изучения дисциплины является получение знаний в области объектно-ориентированного программирования, формирование навыков создания приложений с применением принципов объектно-ориентированного программирования, умений в области отладки программ, написанных с применением объектно-ориентированной парадигмы.

2. Задачами являются:

-изучение основных принципов объектно-ориентированного программирования: абстрагирование, инкапсуляция, наследование, полиморфизм;

-знакомство с классами и структурами, абстрактными классами, интерфейсами;

-получение знаний проектирования приложений с применением объектно-ориентированной парадигмы.

-формирование умений формализации задачи и получение навыков отладки программ.

3. Дисциплина обеспечивает формирование знаний в области разработки и отладки программ, знакомство с тестированием приложений.

4. Дисциплина обеспечивает умения:

-освоение основ инженерии программного обеспечения

-приобретение умений по разработке пользовательского интерфейса.

5. Дисциплина формирует навыки получения практического применения знания объектно-ориентированного моделирования и проектирования, а также умений применения их на практике.

3.2 Место дисциплины в структуре ОПОП

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Программирование»

и обеспечивает изучение последующих дисциплин:

1. «Учебная практика (технологическая (проектно-технологическая) практика)»

2. «Web-программирование»

3. «Программирование на языке Kotlin»

4. «Сети ЭВМ»

5. «Защита компьютерной информации»

6. «Основы искусственного интеллекта»

3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

Код компетенции/ индикатора компетенции	Наименование компетенции/индикатора компетенции
ПК-2	Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности
<i>ПК-2.1</i>	<i>Анализирует проблемную ситуацию, планирует разработку системы, осуществляет постановку целей</i>
<i>ПК-2.2</i>	<i>Разрабатывает техническое задание, концепцию системы</i>
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение
<i>ПК-3.1</i>	<i>Анализирует требования к программному обеспечению</i>
<i>ПК-3.2</i>	<i>Разрабатывает, изменяет и согласовывает архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения</i>
<i>ПК-3.4</i>	<i>Проектирует программные интерфейсы</i>
СПК-1	Способен разрабатывать компоненты автоматизированных систем, используя современные инструментальные средства и технологии программирования
<i>СПК-1.3</i>	<i>Владеет инструментальными средствами разработки и отладки компонентов автоматизированных систем</i>

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Содержание разделов дисциплины

4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	Лек, ач	Пр, ач	Лаб, ач	ИКР, ач	СР, ач
1	Введение	1		0		
2	Тема 1. Обзор платформы .NET	1				1
3	Тема 2. Система типов	1		1		3
4	Тема 3. Основные языковые конструкции и исключения	1		1		4
5	Тема 4. Методы и параметры	1		1	1	4
6	Тема 5. Массивы	1		1		4
7	Тема 6. Принципы объектно-ориентированного программирования	2	3	1	1	6
8	Тема 7. Использование переменных ссылочного типа	2	2	1		4
9	Тема 8. Создание и уничтожение объектов	1	1	1		2
10	Тема 9. Наследование в C#	2	2	2	1	7
11	Тема 10. Агрегирование	2		1		4
12	Тема 11. Инкапсуляция данных и определение перегруженных операторов	2	1	1		5
13	Тема 12. Делегаты и события	2		1		4
14	Тема 13. Использование коллекций и создание общих типов	2	1	1		4
15	Тема 14. Технологии доступа к данным	1	1	1		2
16	Тема 15. Создание простого приложения Windows Forms	2		1		4
17	Тема 16. Построение пользовательского интерфейса для приложения Windows Forms	2		1		4
18	Тема 17. Проверка данных, введенных пользователем	2	1	1		4
19	Тема 18. Введение в UML. Объектно-ориентированное моделирование	3	3			5
20	Тема 19. Проектирование приложений	2	2			2
21	Заключение	1				
	Итого, ач	34	17	17	3	73
	Из них ач на контроль	0	0	0	0	0
	Общая трудоемкость освоения, ач/зе	144/4				

4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Введение	Диаграммы классов. Отношения между классами. Диаграммы объектов. Диаграммы последовательностей, состояний, деятельностей, коммуникаций. Варианты использования. Диаграммы компонентов, развертывания.
2	Тема 1. Обзор платформы .NET	Создание программы на С#. Общеязыковая среда исполнения (CLR). Библиотека классов .NET Framework. Классы. Структура программы на С#, базовые операции ввода-вывода, компиляция, выполнение и отладка. Пространства имен. Метод Main. Операции ввода/вывода. Комментарии.
3	Тема 2. Система типов	Использование типов-значений. Общая система типов. Типы-значения и ссылочные типы. Встроенные типы и типы, определяемые пользователем. Операторы. Перечисления. Структуры. Преобразование типов.
4	Тема 3. Основные языковые конструкции и исключения	Блоки выражений. Управляющие операторы. Циклы. Операторы перехода. Обработка исключений. Генерация исключений. Проверка переполнений.
5	Тема 4. Методы и параметры	Определение методов. Вызов методов. Возврат управления из методов. Локальные и общие переменные. Области видимости. Механизмы передачи параметров. Рекурсивные методы. Сигнатуры методов. Перегрузка методов.
6	Тема 5. Массивы	Определение массива. Ранг массива. Доступ к элементам массива. Сравнение массивов и коллекций. Создание и инициализация массивов. Невыровненные массивы. Использование массивов.
7	Тема 6. Принципы объектно-ориентированного программирования	Классы и объекты. Сравнение классов и структур. Абстракция. Инкапсуляция. Контроль доступа. Статические данные и методы. Вложенные классы. Наследование. Полиморфизм. Абстрактные классы. Интерфейсы. Раннее и позднее связывание.
8	Тема 7. Использование переменных ссылочного типа	Объявление и освобождение переменных-ссылок. Сравнение переменных-ссылок. Передача ссылок как параметров методов. Класс Exception. Класс String. Иерархия объектов CTS. Рефлексия. Преобразование ссылок. Операторы is и as. Упаковка и распаковка.
9	Тема 8. Создание и уничтожение объектов	Конструкторы. Перегрузка конструкторов. Инициализация данных. Время жизни объекта. Области видимости. Сборка мусора. Деструкторы.
10	Тема 9. Наследование в С#	Производные классы. Виртуальные методы. Замещение методов. Спецификатор sealed. Интерфейсы. Абстрактные классы. Сравнение абстрактных классов и интерфейсов.

№ п/п	Наименование темы дисциплины	Содержание
11	Тема 10. Агрегирование	Применение вложенных классов. Использование агрегирования. Использование пространства имен. Использование модулей и сборок. Пространства имен и область видимости.
12	Тема 11. Инкапсуляция данных и определение перегруженных операторов	Создание и использование свойств. Создание и использование индексов. Создание и использование свойств. Создание и использование индексов. Перегрузка операторов. Перегрузка операторов.
13	Тема 12. Делегаты и события	Создание и использование делегатов. Использование лямбда выражений. Определение и использование событий.
14	Тема 13. Использование коллекций и создание общих типов	Использование коллекций. Создание общих типов. Общие интерфейсы. Общие методы и делегаты. Реализация пользовательской коллекции.
15	Тема 14. Технологии доступа к данным	Использование LINQ для создания запросов к данным. Использование методов расширений и операторов LINQ. Создание динамических запросов и выражений.
16	Тема 15. Создание простого приложения Windows Forms	Компоненты приложения WinForms. Создание и сохранение проекта приложения WinForms. Формы, свойства форм. Элементы управления. Обработчики событий. Управление конфигурацией. Сборки.
17	Тема 16. Построение пользовательского интерфейса для приложения Windows Forms	Управление формами и диалоговыми окнами. Модальные и немодальные формы. Меню, контекстное меню. Панель инструментов. Полоса прокрутки. Строка состояния. Контекстное окно приложения.
18	Тема 17. Проверка данных, введенных пользователем	Встроенные механизмы проверки. Контроль данных для элемента управления TextBox. Элемент управления MaskedTextBox. Контроль данных на уровне полей. Компонент ErrorProvider. События проверки данных. Проверка данных на уровне форм. Кнопки Ассерт и Cancel.
19	Тема 18. Введение в UML. Объектно-ориентированное моделирование	Диаграммы классов. Отношения между классами. Диаграммы объектов. Диаграммы последовательностей, состояний, деятельностей, коммуникаций. Варианты использования. Диаграммы компонентов, развертывания.
20	Тема 19. Проектирование приложений	Архитектура программных систем. Обзор различных подходов к разработке приложений. Применение паттернов проектирования приложений. Принцип SOLID.
21	Заключение	Рассматриваются перспективы применения объектно-ориентированного подхода к разработке приложений.

4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Обзор языка C#. Использование типов-значений.	1
2. Использование базовых языковых конструкций. Использование методов.	1
3. Создание и использование массивов.	1
4. Создание и использование классов.	2
5. Использование переменных ссылочного типа.	2
6. Применение конструкторов.	2
7. Использование наследования с применением интерфейсов.	2
8. Создание и использование делегатов.	2
9. Использование LINQ.	2
10. Создание приложения Windows Forms	2
Итого	17

4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Объектная модель. Абстрагирование. Передача сообщений. Взаимодействие между объектами. Типизация. Классы. Взаимодействие между классами.	2
2. Основные принципы объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм.	2
3. Основы UML. Диаграммы классов. Отношения между классами. Диаграммы объектов.	3
4. Диаграммы последовательностей, состояний, деятельностей, коммуникаций.	2
5. Проектирование и развертывание приложений. Диаграммы использования. Диаграммы компонентов и развертывания.	2
6. Архитектура программных систем. Обзор различных подходов к разработке приложений. Применение паттернов проектирования приложений.	2
7. Принцип SOLID.	4
8. Технологии доступа к данным.	
Итого	17

4.4 Курсовое проектирование

Цель работы (проекта): закрепление теоретических знаний и практических навыков разработки программного обеспечения на основе объектно-ориентированного подхода. Выполнение курсовой работы должно базироваться на объектной модели, являющейся концептуальной базой для объектно-ориентированной парадигмы. В курсовой работе должны быть отражены ключевые кон-

цепций объектной модели: абстрагирование, передача сообщений, инкапсуляция, модульность, полиморфизм и наследование.

Содержание работы (проекта): Задание на курсовую работу состоит из нескольких частей.

Задачи первой части курсовой работы направлены, прежде всего, на закрепление теоретических знаний и практических навыков разработки программ на основе объектно-ориентированного подхода и разработке объектной модели.

Задачи второй части курсовой работы связаны с программной реализацией задач принятия решений (упорядочение объектов по функции полезности, оценки различных критериев, балльные оценки объектов и т.п.). Решение задач должно базироваться на объектной модели.

Темы третьей части представляют собой работу с графами различных типов. Решение задач данного раздела направлено на закрепление теоретических знаний, полученных в рамках ранее изученных дисциплин учебной программы и подготовки к изучению последующих дисциплин. Обязательным требованием решения задач является применение принципов объектно-ориентированного подхода.

Четвертая часть курсовой работы связана с решением задач имитационного моделирования. Программа должна обеспечивать имитацию динамических процессов разной природы: транспортных средств, производственных процессов, задач управления.,

Работа должна содержать не менее 5 источников, пояснительная записка должна быть не менее 30 страниц, сдается в электронном виде и размещается в Moodle. Студенты пишут работу по вариантам.

Темы:

№ п/п	Название темы	Перевод темы
1	Поиск кратчайшего гамильтонова цикла графа с помощью алгоритма ближайшего соседа	Search shortest Hamiltonian cycle of a graph using the nearest neighbor algorithm

4.5 Реферат

Реферат не предусмотрен.

4.6 Индивидуальное домашнее задание

Индивидуальное домашнее задание не предусмотрено.

4.7 Доклад

Доклад не предусмотрен.

4.8 Кейс

Кейс не предусмотрен.

4.9 Организация и учебно-методическое обеспечение самостоятельной работы

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым

образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятельности, которые указывают путь решения проблемы.

Контроль самостоятельной работы студентов осуществляется на лекционных, лабораторных и практических занятиях студентов

Текущая СРС	Примерная трудоемкость, ач
Работа с лекционным материалом, с учебной литературой	6
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	2
Самостоятельное изучение разделов дисциплины	4
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	10
Подготовка к контрольным работам, коллоквиумам	10
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	34
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	7
ИТОГО СРС	73

5 Учебно-методическое обеспечение дисциплины

5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Буч, Грейди. Язык UML [Текст] : руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон ; [пер. с англ. А.А. Слинкин], 2004. -429 с.	73
2	Объектно-ориентированное программирование [Электронный ресурс] : метод. указания к лаб. работам : учеб. электрон. изд. / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) ”ЛЭТИ”, 2014. -1 эл. опт. диск (CD-ROM)	неогр.
3	Горячев, Александр Вадимович. Объектно-ориентированное моделирование [Текст] : учеб. пособие / А.В. Горячев, Д.К. Кравчук, Н.Е. Новакова, 2010. -79 с.	неогр.
Дополнительная литература		
1	Троелсен, Эндрю. С# и платформа .NET [Текст] : монография / Э. Троелсен; Пер. с англ. Р.Михеева, 2002. -795 с.	19
2	Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] : справочное издание / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес; [Пер. с англ. А. Слинкина], 2001. -366 с.	19

5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	Сайт Microsoft http://msdn.microsoft.com/ru-ru/

5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=11330>

6 Критерии оценивания и оценочные материалы

6.1 Критерии оценивания

Для дисциплины «Объектно-ориентированное программирование» предусмотрены следующие формы промежуточной аттестации: зачет с оценкой.

Зачет с оценкой

Оценка	Описание
Неудовлетворительно	Курс не освоен. Студент испытывает серьезные трудности при ответе на ключевые вопросы дисциплины
Удовлетворительно	Студент в целом овладел курсом, но некоторые разделы освоены на уровне определений и формулировок
Хорошо	Студент овладел курсом, но в отдельных вопросах испытывает затруднения. Умеет решать задачи
Отлично	Студент демонстрирует полное овладение курсом, способен применять полученные знания при решении конкретных задач

Особенности допуска

Студент допускается к зачету с оценкой, если была защищена курсовая работа, выполнены и защищены все лабораторные работы.

6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

Вопросы к дифф.зачету

№ п/п	Описание
1	Основные принципы объектно-ориентированного программирования. Классы и объекты.
2	Абстрактные классы. Сравнение абстрактных классов с интерфейсами.
3	Структура программы на языке С#. Метод Main. Использование пространства имен System.
4	Система типов. Использование типов-значений и ссылочных типов. Конвертирование типов данных. Типизация.
5	Управляющие операторы.
6	Циклы.
7	Исключения. Обработка исключений.
8	Использование методов.
9	Использование параметров в методах.
10	Массивы.
11	Основные принципы объектно-ориентированного программирования. Использование инкапсуляции.
12	Определение объектно-ориентированных систем. Наследование. Полиморфизм. Интерфейсы.
13	Использование переменных ссылочного типа. Иерархия объектов. Операции сравнения для переменных-значений и переменных ссылочного типа.
14	Сравнение строк. Рефлексия. Преобразование данных.
15	Создание и уничтожение объектов. Использование конструкторов.
16	Инициализация данных. Инициализация списков.
17	Сборка мусора. Использование деструкторов.
18	Наследование в С#. Производные классы.
19	Наследование в С#. Применение методов.
20	Применение интерфейсов.
21	Пространства имен. Объявление пространства имен.
22	Операторы. Перегрузка операторов.
23	Делегаты. Назначение и применение.
24	События. Управление событиями.
25	Лямбда выражения.
26	Коллекции. Использование коллекций. Методы и свойства коллекций.

27	Использование общих типов.
28	Интерфейсы коллекций. Свойства и методы.
29	Запросы данных. Технология LINQ.
30	Запрос к данным и построение результирующего набора с помощью LINQ. Методы, применяемы для запроса данных, фильтрации, сортировки и группировки данных.
31	Использование операторов запросов в LINQ.
32	Построение динамических LINQ-запросов. Типы выражений.
33	Диаграммы классов UML. Отношения между классами.
34	Диаграммы деятельности UML
35	Принципы объектно-ориентированного проектирования SOLID. Принцип единственности обязанности
36	Принципы объектно-ориентированного проектирования SOLID. Принцип открытости-закрытости.
37	Принципы объектно-ориентированного проектирования SOLID. Принцип разделения интерфейсов.
38	Паттерны проектирования приложений.
39	Создание простого приложения Windows Forms. Компоненты приложения Windows Forms. Компиляция и запуск приложения.
40	Тестирование приложений. Виды тестирования.

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
1	Тема 1. Обзор платформы .NET	Коллоквиум
2	Тема 2. Система типов	Коллоквиум
3	Тема 3. Основные языковые конструкции и исключения	Коллоквиум
4	Тема 4. Методы и параметры	Коллоквиум
5	Тема 5. Массивы	Коллоквиум
6	Тема 6. Принципы объектно-ориентированного программирования	Коллоквиум
7	Тема 7. Использование переменных ссылочного типа	Коллоквиум
8	Тема 8. Создание и уничтожение объектов	Коллоквиум
9	Тема 9. Наследование в C#	Коллоквиум
10	Тема 10. Агрегирование	Коллоквиум
11	Тема 11. Инкапсуляция данных и определение перегруженных операторов	Коллоквиум
12	Тема 12. Делегаты и события	Коллоквиум
13	Тема 13. Использование коллекций и создание общих типов	Коллоквиум
14	Тема 14. Технологии доступа к данным	Коллоквиум
15	Тема 15. Создание простого приложения Windows Forms	Коллоквиум
16	Тема 16. Построение пользовательского интерфейса для приложения Windows Forms	Коллоквиум
17	Тема 18. Введение в UML. Объектно-ориентированное моделирование Заключение	Защита КР / КП

6.4 Методика текущего контроля

Текущий контроль включает в себя контроль посещаемости (не менее 80% занятий), по результатам которого студент получает допуск к итоговому дифференцированному зачету.

Порядок выполнения лабораторных работ, подготовки отчетов и их защит

В процессе обучения по дисциплине студент обязан выполнить 10 лабораторных работ. Под выполнением лабораторных работ подразумевается подготовка к работе, проведение экспериментальных исследований, подготовка отчета и его защита на коллоквиуме.

Лабораторные работы защищаются студентами индивидуально. Каждый

студент получает вопрос по теоретической части, или по процедуре проведения экспериментальных исследований, или по последующей обработке результатов, после чего ему предоставляется время для подготовки ответа. При обсуждении ответа преподаватель может задать несколько уточняющих вопросов. В случае если студент демонстрирует достаточное знание вопроса, работа считается защищенной.

Порядок выполнения практических работ, подготовки отчетов и их защит

В процессе обучения по дисциплине студент обязан выполнить 8 практических работ. Под выполнением практических работ подразумевается подготовка к работе, проведение экспериментальных исследований, подготовка отчета и его защита на коллоквиуме.

Практические работы защищаются студентами индивидуально. Каждый студент получает вопрос по теоретической части, или по процедуре проведения экспериментальных исследований, или по последующей обработке результатов, после чего ему предоставляется время для подготовки ответа. При обсуждении ответа преподаватель может задать несколько уточняющих вопросов. В случае если студент демонстрирует достаточное знание вопроса, работа считается защищенной.

Коллоквиум проводится на основе вопросов к дифф. зачету, изученных до момента проведения коллоквиума.

Критерии оценивания на коллоквиуме:

Положительная оценка ("Удовлетворительно" и выше) ставится при доказательстве самостоятельности выполнения работы, полностью выполненного задания и объяснения особенностей выполнения работы, при этом могут иметься заметные проблемы с работой проекта или значительные замечания к отчету.

Оценка "Хорошо" ставится при том, что ответ на вопрос в целом правильный, но имеются некоторые неточности в формулировках ответов

Оценка "Отлично" ставится при отсутствии замечаний к ответам студента.

Текущий контроль при выполнении курсовой работы осуществляется в соответствии с методическими указаниями по курсовому проектированию и заданием на курсовую работу.

Оформление пояснительной записки на курсовую работу выполняется в соответствии с требованиями к студенческим работам, принятым в СПбГЭТУ.

Защита курсовой работы осуществляется в соответствии с требованиями «Положения о промежуточной аттестации».

Текущий контроль при выполнении курсового проекта (работы) осуществляется в соответствии с методическими указаниями по курсовому проектированию и заданием на курсовой проект (работу).

Оформление пояснительной записки на курсовой проект (работу) выполняется в соответствии с требованиями к студенческим работам принятым в СПбГЭТУ.

Защита курсового проекта (работы) осуществляется в соответствии с требованиями «Положения о промежуточной аттестации». Защита выполняется очно или дистанционно не ранее, чем через три дня после опубликования окончательного варианта пояснительной записки в соответствующем задании электронного курса в виртуальном образовательном кластере ЛЭТИ (Moodle).

Критерии оценивания курсовой работы:

Положительная оценка ("Удовлетворительно" и выше) ставится при доказательстве самостоятельности выполнения работы, полностью выполненного задания и объяснения особенностей выполнения работы, при этом могут иметься заметные проблемы с работой проекта или значительные замечания к пояс-

нительной записке.

Оценка "Хорошо" ставится при том, что поставленная задача выполнена, но имеются некоторые недочеты в его работы или есть серьезные замечания преподавателя к пояснительной записке.

Оценка "Отлично" ставится при отсутствии замечаний к выполненной задаче и пояснительной записке.

Защита курсовой работы является обязательным видом контроля. Без защиты курсовой работы студент не допускается до дифференцированного зачета.

Все отчетные документы должны быть размещены в Moodle.

Критерии оценивания курсовой работы:

Контроль самостоятельной работы студентов осуществляется на лекционных, лабораторных и практических занятиях студентов по методикам, описанным выше.

7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя – персональный компьютер IBM совместимый Pentium или выше, проектор, экран, маркерная доска	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 2008 и выше
Лабораторные работы	Лаборатория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя – персональный компьютер IBM совместимый Pentium или выше, проектор, экран, маркерная доска	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 2008 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, проектор, экран, учебная доска, персональный компьютер (не менее, чем 1 компьютер на 2 обучающихся)	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 2008 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше

8 Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Дата	Изменение	Дата и номер протокола заседания УМК	Автор	Начальник ОМОЛА