

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ**

УТВЕРЖДАЮ

Директор/Декан
факультета цифровых технологий
Аникуев Сергей Викторович

«__» _____ 20__ г.

Рабочая программа дисциплины

Б1.В.08 Программная инженерия

09.03.02 Информационные системы и технологии

Инженерия информационных систем и цифровые технологии

бакалавр

очная

1. Цель дисциплины

Цель изучения дисциплины: формирование у студентов системы теоретических представлений о современных подходах к проектированию, разработке, тестированию и отладке сложных приложений, а также получении практических навыков программирования приложений с графическим пользовательским интерфейсом.

Задачами дисциплины являются:

- 1) изучение теоретических основ программирования для операционной системы Windows;
- 2) изучение принципов создания графического интерфейса пользователя;
- 3) развитие навыков программирования с использованием технологии .NET Framework;
- 4) изучение механизмов применения и создания библиотек;
- 5) изучение основных принципов тестирования и отладки программного обеспечения;
- 6) изучение возможностей языка высокого уровня для организации доступа к файловой системе;
- 7) изучение технологий доступа к данным.

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

Процесс изучения дисциплины направлен на формирование следующих компетенций ОП ВО и овладение следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.1 Способен создавать программный код, настраивать и конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС	знает умеет владеет навыками
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.2 Способен развертывать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС	знает умеет владеет навыками
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.3 Выполняет интеграцию ИС с существующими ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС	знает умеет владеет навыками

3. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» является дисциплиной части, формируемой участниками образовательных отношений программы.

Изучение дисциплины осуществляется в 8 семестре(-ах).

Для освоения дисциплины «Программная инженерия» студенты используют знания, умения и навыки, сформированные в процессе изучения дисциплин:

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений

Стандартизация, сертификация и контроль качества программного обеспечения

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений

Конфигурирование в ИС

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений

Web-программирование

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений

Разработка серверных веб-приложений

Технологическая (проектно-технологическая) практика
Интернет-технологии
Web-программирование
Конфигурирование в информационных системах
Базы данных
Анализ данных
Управление данными
Технологии разработки web-ресурсов
Системы поддержки принятия решений
Разработка программных приложений
Инженерия данных
Технологическая (проектно-технологическая) практика
Интернет-технологии
Web-программирование
Конфигурирование в информационных системах
Базы данных
Анализ данных
Управление данными
Технологии разработки web-ресурсов
Системы поддержки принятия решений
Разработка программных приложений
Анализ и визуализация данных
Технологическая (проектно-технологическая) практика
Интернет-технологии
Web-программирование
Конфигурирование в информационных системах
Базы данных
Анализ данных
Управление данными
Технологии разработки web-ресурсов
Системы поддержки принятия решений
Разработка программных приложений
Разработка мобильных приложений
Технологическая (проектно-технологическая) практика
Интернет-технологии
Web-программирование
Конфигурирование в информационных системах
Базы данных
Анализ данных
Управление данными
Технологии разработки web-ресурсов
Системы поддержки принятия решений
Разработка программных приложений
Разработка программных приложений
Технологическая (проектно-технологическая) практика
Интернет-технологии
Web-программирование
Конфигурирование в информационных системах
Базы данных
Анализ данных
Управление данными
Технологии разработки web-ресурсов
Системы поддержки принятия решений
Разработка программных приложений
Информационные системы управления предприятием

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений Автоматизированные системы в управлении

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений Технологическая (проектно-технологическая) практика

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений Геопространственный анализ

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений Интернет вещей

Технологическая (проектно-технологическая) практика

Интернет-технологии

Web-программирование

Конфигурирование в информационных системах

Базы данных

Анализ данных

Управление данными

Технологии разработки web-ресурсов

Системы поддержки принятия решений

Разработка программных приложений Разработка игровых приложений

Освоение дисциплины «Программная инженерия» является необходимой основой для последующего изучения следующих дисциплин:

2.1.	Объектно-ориентированное программирование	8	6	2	4	10	КТ 1	Собеседование, Защита лабораторной работы	
2.2.	Наследование	8	6	2	4	10	КТ 1	Собеседование, Защита лабораторной работы	
2.3.	Файловый ввод-вывод	8	6	2	4	10	КТ 2	Собеседование, Защита лабораторной работы	
2.4.	Графический интерфейс на основе Windows Forms	8	14	2	12	10	КТ 2	Собеседование, Защита лабораторной работы	
2.5.	Графический интерфейс на основе Windows Presentation Foundation	8	8		8	10	КТ 2	Собеседование, Защита лабораторной работы	
2.6.	Потоки, задачи и синхронизация	8	6	2	4		КТ 2	Собеседование, Защита лабораторной работы	
2.7.	Многомодульные приложения	8	2		2		КТ 2	Собеседование, Защита лабораторной работы	
3.	3 раздел. Экзамен								
3.1.	Экзамен	8					КТ 3	Собеседование, Практико-ориентированные задачи и ситуационные задачи	
	Промежуточная аттестация		Эк						
	Итого		180	16		48	80		
	Итого		180	16		48	80		

5.1. Лекционный курс с указанием видов интерактивной формы проведения занятий

Тема лекции (и/или наименование раздел) (вид интерактивной формы проведения занятий)/ (практическая подготовка)	Содержание темы (и/или раздела)	Всего, часов / часов интерактивных занятий/ практическая подготовка
Основы языка C#	Система типов .NET	2/2
Исполнение программ	Управление ходом выполнения программы	2/2
Массивы	Массивы	2/-
Объектно-ориентированное программирование	Классы и структуры	2/-
Наследование	Наследование реализации и наследование	2/-

	интерфейсов	
Файловый ввод-вывод	Иерархия типов для файлового ввода-вывода	2/-
Графический интерфейс на основе Windows Forms	Принципы построения графического интерфейса	2/-
Потоки, задачи и синхронизация	Асинхронные и многопоточные приложения	2/-
Итого		16

5.2.2. Лабораторные занятия с указанием видов проведения занятий в интерактивной форме

Наименование раздела дисциплины	Формы проведения и темы занятий (вид интерактивной формы проведения занятий)/(практическая подготовка)	Всего, часов / часов интерактивных занятий/ практическая подготовка	
		вид	часы
Основы языка C#	Разработка консольного приложения	лаб.	2
Основы языка C#	Перенаправление потоков ввода-вывода	лаб.	2
Исполнение программ	Управление потоком выполнения в программе	лаб.	2
Массивы	Одномерные массивы	лаб.	2
Массивы	Многомерные массивы	лаб.	2
Объектно-ориентированное программирование	Классы. Объектное моделирование	лаб.	2
Объектно-ориентированное программирование	Конструктор класса. Перегрузка методов класса	лаб.	2
Наследование	Проектирование иерархии классов	лаб.	2
Наследование	Полиморфизм на основе интерфейсов	лаб.	2
Файловый ввод-вывод	Основы работы с файлами	лаб.	2
Файловый ввод-вывод	Обработка данных в файлах	лаб.	2
Графический интерфейс на основе Windows Forms	Основы проектирования приложений Windows Forms	лаб.	2
Графический интерфейс на основе Windows Forms	Простейшие элементы управления и события	лаб.	2
Графический интерфейс на	Меню и строка состояния в приложениях Windows Forms	лаб.	2

основе Windows Forms			
Графический интерфейс на основе Windows Forms	Диалоговые окна	лаб.	2
Графический интерфейс на основе Windows Forms	Обработка табличных данных	лаб.	2
Графический интерфейс на основе Windows Forms	Технология GDI+ в приложениях Windows Forms	лаб.	2
Графический интерфейс на основе Windows Presentation Foundation	Приложения Windows Presentation Foundation	лаб.	2
Графический интерфейс на основе Windows Presentation Foundation	Ресурсы, стили, триггеры	лаб.	2
Графический интерфейс на основе Windows Presentation Foundation	Механизм привязки WPF	лаб.	2
Графический интерфейс на основе Windows Presentation Foundation	Источники привязки произвольного типа	лаб.	2
Потоки, задачи и синхронизация	Асинхронные делегаты	лаб.	2
Потоки, задачи и синхронизация	Механизмы передачи данных потокам	лаб.	2
Многомодульные приложения	Разработка многомодульного приложения	лаб.	2

5.3. Курсовой проект (работа) учебным планом не предусмотрен

5.4. Самостоятельная работа обучающегося

Темы и/или виды самостоятельной работы	Часы
Система типов .NET	10

Управление ходом выполнения программы	10
Массивы	10
Классы и структуры	10
Наследование реализации и наследование интерфейсов	10
Иерархия типов для файлового ввода-вывода	10
Принципы построения графического интерфейса	10
Приложения Windows Presentation Foundation	10

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Учебно-методическое обеспечение для самостоятельной работы обучающегося по дисциплине «Программная инженерия» размещено в электронной информационно-образовательной среде Университета и доступно для обучающегося через его личный кабинет на сайте Университета. Учебно-методическое обеспечение включает:

1. Рабочую программу дисциплины «Программная инженерия».
 2. Методические рекомендации для организации самостоятельной работы обучающегося по дисциплине «Программная инженерия».
 3. Методические рекомендации по выполнению письменных работ () (при наличии).
 4. Методические рекомендации по выполнению контрольной работы студентами заочной формы обучения (при наличии)
 5. Методические указания по выполнению курсовой работы (проекта) (при наличии).
- Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации:

№ п/п	Темы для самостоятельного изучения	Рекомендуемые источники информации (№ источника)		
		основная (из п.8 РПД)	дополнительная (из п.8 РПД)	метод. лит. (из п.8 РПД)
1	Основы языка C#. Система типов .NET	Л1.1, Л1.2	Л2.2	Л3.1
2	Исполнение программ. Управление ходом выполнения программы	Л1.1	Л2.1, Л2.2	Л3.1
3	Массивы. Массивы	Л1.1, Л1.2	Л2.2, Л2.3	Л3.1
4	Объектно-ориентированное программирование. Классы и структуры	Л1.1, Л1.2	Л2.2, Л2.3	Л3.1
5	Наследование. Наследование реализации и наследование интерфейсов	Л1.3	Л2.2, Л2.3	Л3.2
6	Файловый ввод-вывод. Иерархия типов для файлового ввода-вывода	Л1.3	Л2.2, Л2.4	Л3.1
7	Графический интерфейс на основе Windows Forms. Принципы построения графического интерфейса	Л1.3	Л2.2, Л2.3, Л2.4	Л3.1
8	Графический интерфейс на основе Windows Presentation Foundation. Приложения Windows Presentation Foundation	Л1.3	Л2.2, Л2.3, Л2.4	Л3.1, Л3.2

7. Фонд оценочных средств (оценочных материалов) для проведения промежуточной аттестации обучающихся по дисциплине «Программная инженерия»

7.1. Перечень индикаторов компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Индикатор компетенции (код и содержание)	Дисциплины/элементы программы (практики, ГИА), участвующие в формировании индикатора компетенции	1		2		3		4	
		1	2	3	4	5	6	7	8
ПК-1.1: Способен создавать программный код, настраивать и	Web-программирование						x		
	Автоматизированные системы в управлении						x		

Индикатор компетенции (код и содержание)	Дисциплины/элементы программы (практики, ГИА), участвующие в формировании индикатора компетенции	1		2		3		4	
		1	2	3	4	5	6	7	8
конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС	Анализ и визуализация данных							x	
	Базы данных				x				
	Геопространственный анализ							x	
	Дисциплины (модули) по выбору 3 (ДВ.3)						x		
	Инженерия данных						x		
	Информационные системы управления предприятием АПК						x		
	Конфигурирование в ИС					x			
	Преддипломная практика								x
	Разработка игровых приложений						x		
	Разработка мобильных приложений						x		
	Разработка программных приложений						x		
	Разработка серверных веб-приложений							x	
	Стандартизация, сертификация и контроль качества программного обеспечения				x				
	Технологии разработки и интеграции								x
	Технологическая (проектно-технологическая) практика				x		x		
ПК-1.2:Способен разворачивать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС	Инженерия данных						x		
	Конфигурирование в ИС					x			
	Преддипломная практика								x
	Разработка мобильных приложений						x		
	Разработка программных приложений						x		
	Разработка серверных веб-приложений							x	
Технологии разработки и интеграции								x	
ПК-1.3:Выполняет интеграцию ИС с существующими ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС	Анализ и визуализация данных							x	
	Инженерия данных						x		
	Интернет вещей							x	
	Конфигурирование в ИС					x			
	Преддипломная практика								x
	Разработка мобильных приложений						x		
	Разработка программных приложений						x		
	Разработка серверных веб-приложений							x	
Технологии разработки и интеграции								x	

7.2. Критерии и шкалы оценивания уровня усвоения индикатора компетенций, опреде-

ляющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Оценка знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций по дисциплине «Программная инженерия» проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль проводится в течение семестра с целью определения уровня усвоения обучающимися знаний, формирования умений и навыков, своевременного выявления преподавателем недостатков в подготовке обучающихся и принятия необходимых мер по её корректировке, а также для совершенствования методики обучения, организации учебной работы и оказания индивидуальной помощи обучающемуся.

Промежуточная аттестация по дисциплине «Программная инженерия» проводится в виде Экзамен.

За знания, умения и навыки, приобретенные студентами в период их обучения, выставляются оценки «ЗАЧТЕНО», «НЕ ЗАЧТЕНО». (или «ОТЛИЧНО», «ХОРОШО», «УДОВЛЕТВОРИТЕЛЬНО», «НЕУДОВЛЕТВОРИТЕЛЬНО» для дифференцированного зачета/экзамена)

Для оценивания знаний, умений, навыков и (или) опыта деятельности в университете применяется балльно-рейтинговая система оценки качества освоения образовательной программы. Оценка проводится при проведении текущего контроля успеваемости и промежуточных аттестаций обучающихся. Рейтинговая оценка знаний является интегрированным показателем качества теоретических и практических знаний и навыков студентов по дисциплине.

Состав балльно-рейтинговой оценки студентов очной формы обучения

Для студентов очной формы обучения знания по осваиваемым компетенциям формируются на лекционных и практических занятиях, а также в процессе самостоятельной подготовки.

В соответствии с балльно-рейтинговой системой оценки, принятой в Университете студентам начисляются баллы по следующим видам работ:

№ контрольной точки	Оценочное средство результатов индикаторов достижения компетенций	Максимальное количество баллов	
8 семестр			
КТ 1	Собеседование	15	
КТ 1	Защита лабораторной работы	15	
КТ 2	Собеседование	15	
КТ 2	Защита лабораторной работы	15	
КТ 3	Собеседование	20	
КТ 3	Практико-ориентированные задачи и ситуационные задачи	20	
Сумма баллов по итогам текущего контроля		100	
Посещение лекционных занятий		20	
Посещение практических/лабораторных занятий		20	
Результативность работы на практических/лабораторных занятиях		30	
Итого		170	
№ контрольной точки	Оценочное средство результатов индикаторов достижений компетенций	Максимальное количество баллов	Критерии оценки знаний студентов
8 семестр			
КТ 1	Собеседование	15	15 баллов заслуживает студент, который полно и развернуто ответил на вопрос. 10 баллов заслуживает студент, который полно ответил на вопрос. 5 баллов заслуживает студент, который не полно ответил на вопрос. 0 баллов заслуживает студент, не ответил на вопрос.

КТ 1	Защита лабораторной работы	15	<p>Выполнение заданий на лабораторных работах 15 баллов – за оцененное на «отлично» выполнение практических заданий по всем темам дисциплины, т.е. практические задания выполнены правильно, аккуратно и в установленные преподавателем сроки; 12 баллов – за оцененное на «хорошо» выполнение практических заданий по всем темам дисциплины, практические задания выполнены правильно, аккуратно, но с нарушением установленных преподавателем сроков; 8 баллов - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, практические задания выполнены с незначительными ошибками, не аккуратно, с нарушением установленных преподавателем сроков; 3 балла - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, т.е. практические задания выполнены с существенными ошибками, не аккуратно, с нарушением установленных преподавателем сроков; 1 балл - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, т.е. выполнены не все практические, а выполненные имеют существенные ошибки, не сданы преподавателю в установленные сроки.</p>
КТ 2	Собеседование	15	<p>15 баллов заслуживает студент, который полно и развернуто ответил на вопрос. 10 баллов заслуживает студент, который полно ответил на вопрос. 5 баллов заслуживает студент, который не полно ответил на вопрос. 0 баллов заслуживает студент, не ответил на вопрос.</p>

КТ 2	Защита лабораторной работы	15	<p>Выполнение заданий на лабораторных работах 15 баллов – за оцененное на «отлично» выполнение практических заданий по всем темам дисциплины, т.е. практические задания выполнены правильно, аккуратно и в установленные преподавателем сроки; 12 баллов – за оцененное на «хорошо» выполнение практических заданий по всем темам дисциплины, практические задания выполнены правильно, аккуратно, но с нарушением установленных преподавателем сроков; 8 баллов - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, практические задания выполнены с незначительными ошибками, не аккуратно, с нарушением установленных преподавателем сроков; 3 балла - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, т.е. практические задания выполнены с существенными ошибками, не аккуратно, с нарушением установленных преподавателем сроков; 1 балл - за оцененное на «удовлетворительно» выполнение практических заданий по всем темам дисциплины, т.е. выполнены не все практические, а выполненные имеют существенные ошибки, не сданы преподавателю в установленные сроки.</p>
КТ 3	Собеседование	20	<p>20 баллов заслуживает студент, который полно и развернуто ответил на вопрос. 15 баллов заслуживает студент, который полно ответил на вопрос. 10 баллов заслуживает студент, который не полно ответил на вопрос. 0 баллов заслуживает студент, не ответил на вопрос.</p>

КТ 3	Практико-ориентированные задачи и ситуационные задачи	20	20 баллов заслуживает студент, который полно и развернуто выполнил задачу. 15 баллов заслуживает студент, который полно выполнил задачу. 10 баллов заслуживает студент, который не полно выполнил задачу. 0 баллов заслуживает студент, не выполнил задачу.
------	---	----	---

Критерии и шкалы оценивания результатов обучения на промежуточной аттестации

При проведении итоговой аттестации «зачет» («дифференцированный зачет», «экзамен») преподавателю с согласия студента разрешается выставлять оценки («отлично», «хорошо», «удовлетворительно», «зачет») по результатам набранных баллов в ходе текущего контроля успеваемости в семестре по выше приведенной шкале.

В случае отказа – студент сдает зачет (дифференцированный зачет, экзамен) по приведенным выше вопросам и заданиям. Итоговая успеваемость (зачет, дифференцированный зачет, экзамен) не может оцениваться ниже суммы баллов, которую студент набрал по итогам текущей и промежуточной успеваемости.

При сдаче (зачета, дифференцированного зачета, экзамена) к заработанным в течение семестра студентом баллам прибавляются баллы, полученные на (зачете, дифференцированном зачете, экзамене) и сумма баллов переводится в оценку.

Критерии и шкалы оценивания ответа на экзамене

Сдача экзамена может добавить к текущей балльно-рейтинговой оценке студентов не более 20 баллов:

Содержание билета	Количество баллов
Теоретический вопрос №1	до 7
Теоретический вопрос №2	до 7
Задача (оценка умений и	до 6
Итого	20

Критерии оценки ответа на экзамене

Теоретические вопросы (вопрос 1, вопрос 2)

7 баллов выставляется студенту, полностью освоившему материал дисциплины или курса в соответствии с учебной программой, включая вопросы рассматриваемые в рекомендованной программой дополнительной справочно-нормативной и научно-технической литературы, свободно владеющему основными понятиями дисциплины. Требуется полное понимание и четкость изложения ответов по экзаменационному заданию (билету) и дополнительным вопросам, заданных экзаменатором. Дополнительные вопросы, как правило, должны относиться к материалу дисциплины или курса, не отраженному в основном экзаменационном задании (билете) и выявляют полноту знаний студента по дисциплине.

5 балла заслуживает студент, ответивший полностью и без ошибок на вопросы экзаменационного задания и показавший знания основных понятий дисциплины в соответствии с обязательной программой курса и рекомендованной основной литературой.

3 балла дан недостаточно полный и недостаточно развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Речевое оформление требует поправок, коррекции.

2 балла дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не

только на поставленный вопрос, но и на другие вопросы дисциплины.

1 балл дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

0 баллов - при полном отсутствии ответа, имеющего отношение к вопросу.

Оценивание задачи

6 баллов Задачи решены в полном объеме с соблюдением необходимой последовательности.

5 баллов

4 балла Задачи решены с небольшими недочетами.

3 балла

2 балла Задачи решены не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы.

1 баллов Задачи решены частично, с большим количеством вычислительных ошибок, объем выполненной части работы не позволяет сделать правильных выводов.

0 баллов Задачи не решены или работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов.

Перевод рейтинговых баллов в пятибалльную систему оценки знаний обучающихся:

для экзамена:

- «отлично» – от 89 до 100 баллов – теоретическое содержание курса освоено полностью, без пробелов необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному;

- «хорошо» – от 77 до 88 баллов – теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками;

- «удовлетворительно» – от 65 до 76 баллов – теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки;

- «неудовлетворительно» – от 0 до 64 баллов - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к существенному повышению качества выполнения учебных заданий

7.3. Примерные оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Программная инженерия»

Лабораторная работа 1. Разработка консольного приложения

1. Какая функция имеет особенное значение при выполнении программы на языках C, C++, C#?

2. Что такое «точка входа» в программе?

3. Как вы понимаете термины «пространства имен», «класс», «метод», «функция»?

Напишите определение каждому термину.

4. Что такое переменная? Как объявляется переменная?

5. Как объявляется константа? Чем константа отличается от переменной?

6. Перечислите целочисленные типы C#.

7. Перечислите отличия типов char и string.

8. Какие из перечисленных идентификаторов нельзя использовать в качестве имен

пользовательских переменных?

_1_01, b100, int, double_1, _b200, MyVar, create-var, 4perem, _5elem, zo0, wodoo, UserCount, system_call, string, System.Double.

9. Опишите назначение управляющих последовательностей: '\n', '\t', '\r'.

10. Имеется список типов: значимые типы, ссылки на функции, ссылочные типы, массивы, делегаты, корреляционные типы, типы по умолчанию. Какие из представленных типов отсутствуют в иерархии типов CTS?

11. К какой ветке дерева типов принадлежат типы int, float, double? Возможные варианты: встроенные типы по значению, перечисления, самоописываемые типы, упакованные типы по значению.

12. На какой тип отображается тип float в библиотеке .NET? Выберите из предложенных вариантов: System.Single, System.Double, System.UInt64, System.UInt32.

13. Среди представленных типов укажите те, которые не предназначены для представления целых чисел: System.Single, System.Int32, System.Byte, System.Int64, System.Char, float, System.String.

Лабораторная работа 2. Перенаправление потоков ввода-вывода

1. Укажите различие между Console.Write() и Console.WriteLine().

2. Опишите метод Console.ReadKey(). Укажите параметры, возвращаемое значение, применение.

3. Укажите стандартные потоки ввода-вывода в консольном приложении .NET Framework.

4. Укажите различия между методами Console.ReadLine() и Console.Read().

5. Программисту необходимо произвести конвертацию значения типа System.String в значение типа float. Напишите метод, который необходимо использовать.

Лабораторная работа 3. Управление потоком выполнения в программе

1. Опишите, каким образом оформляется комментарий в языке C#.

2. Какое ключевое (зарезервированное) слово в условном операторе является обязательным?

3. Дан фрагмент кода:

<фрагмент кода предоставляется преподавателем>

Чему равно значение переменной res после выполнения данного фрагмента?

4. Дан фрагмент кода:

<фрагмент кода предоставляется преподавателем>

Чему равно значение переменной counter после выполнения данного фрагмента?

5. Дан фрагмент кода:

<фрагмент кода предоставляется преподавателем>

Чему равно значение переменной seo после выполнения данного фрагмента?

6. Что будет выведено в консоль в результате выполнения следующей программы?

<фрагмент кода предоставляется преподавателем>

7. Дан листинг программы:

<фрагмент кода предоставляется преподавателем>

Что будет выведено в консоль?

Лабораторная работа 4. Одномерные массивы

1. Дайте определение массива в языке C#.

2. Укажите строки, в которых верно выполнено объявление массива:

<фрагмент кода предоставляется преподавателем>

3. Дан фрагмент кода:

<фрагмент кода предоставляется преподавателем>

Какое значение содержится в переменной res после выполнения фрагмента?

4. Что будет выведено в консоль в результате работы программы?

<фрагмент кода предоставляется преподавателем>

5. Поясните, как связаны величины: количество элементов в массиве, максимальное значение индекса массива, минимальное значение индекса массива.

6. Что будет выведено в консоль в результате работы программы?

<фрагмент кода предоставляется преподавателем>

7. Какое значение будет выведено в результате выполнения программы?

<фрагмент кода предоставляется преподавателем>

8. Что будет выведено в результате выполнения программы?

<фрагмент кода предоставляется преподавателем>

9. Что будет выведено в результате выполнения программы?

<фрагмент кода предоставляется преподавателем>

Лабораторная работа 5. Многомерные массивы

1. Приведите пример объявления массива. Приведите пример инициализации элементов массива.

2. Что будет выведено в консоль в результате работы приложения?

<фрагмент кода предоставляется преподавателем>

3. Что будет выведено в консоль в результате работы приложения?

<фрагмент кода предоставляется преподавателем>

4. Что будет выведено в консоль в результате работы приложения?

<фрагмент кода предоставляется преподавателем>

5. Приведите фрагмент кода, демонстрирующий использование генератора случайных чисел.

6. В чем преимущества и недостатки непосредственного задания значений элементов массива в { } при его объявлении? В чем недостатки такого подхода?

7. Как изменить размер объявленного и инициализированного массива (например, добавить дополнительные элементы)?

8. Что будет выведено в консоль в результате работы приложения?

<фрагмент кода предоставляется преподавателем>

Лабораторная работа 6. Классы. Объектное моделирование

1. Приведите определение понятия «класс».

2. Что такое структура? Чем структура отличается от класса?

3. Чем отличаются определения классов?

<фрагмент кода предоставляется преподавателем>

4. Опишите ошибки компиляции (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

4. Опишите ошибки компиляции (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

5. Как называется переменная типа класс?

6. С использованием какого зарезервированного слова происходит объявление класса?

Какое зарезервированное слово используется для объявления структуры?

7. Опишите ошибки компиляции (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

8. Какие члены класса являются закрытыми?

<фрагмент кода предоставляется преподавателем>

9. Какие поля объявленного класса доступны из объектной переменной?

<фрагмент кода предоставляется преподавателем>

10. К каким полям нельзя обратиться из объектной переменной?

<фрагмент кода предоставляется преподавателем>

Лабораторная работа 7. Конструктор класса. Перегрузка методов класса

1. Что такое статический класс? Что такое статический метод?

2. Что такое конструктор класса?

3. Что такое перегрузка методов?

4. Может ли один конструктор класса вызывать другой конструктор? Прежде чем отвечать попробуйте реализовать такой вызов в своем разработанном классе.

5. Сколько перегрузок может иметь метод класса?

6. Для каких целей следует перегружать конструктор класса?

7. Что такое сигнатура метода?

8. Какие ошибки присутствуют в объявлении класса?

<фрагмент кода предоставляется преподавателем>

9. Дано определение класса:

<фрагмент кода предоставляется преподавателем>

и фрагмент использования класса в функции Main():

<фрагмент кода предоставляется преподавателем>

Какие ошибки компиляции существуют в коде?

<фрагмент кода предоставляется преподавателем>

10. Укажите ошибки (если есть) в объявлении класса:

<фрагмент кода предоставляется преподавателем>

Лабораторная работа 8. Проектирование иерархии классов

1. Что такое наследование реализации? Как описать синтаксически наследование реализации?

2. Для чего используется ключевое слово base?

3. Можно ли переопределить метод класса? Свойства класса? Данные класса?

4. Как переопределить метод в производном классе?

5. Для чего используется ключевое слово virtual?

6. Для чего используется ключевое слово override?

7. Как поменять цвет фона в консольном приложении?

8. Укажите ошибки (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

9. Сколько базовых классов может быть у любого класса в C#?

10. Даны определения классов на языке C#.

<фрагмент кода предоставляется преподавателем>

Существуют ли ошибки в данном коде (укажите ошибки, если есть)?

Лабораторная работа 9. Полиморфизм на основе интерфейсов

1. Чем отличается наследование интерфейсов от наследования реализации?

2. Поясните, каким образом проявляется полиморфное поведение объектов при реализации классами интерфейсов?

3. Сколько интерфейсов может реализовывать класс?

4. Как объявляются интерфейсные типы? Для чего используются интерфейсы?

5. Даны определения нескольких типов. Укажите ошибки (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

6. Даны определения нескольких типов. Укажите ошибки (если есть) в представленном фрагменте:

<фрагмент кода предоставляется преподавателем>

Лабораторная работа 10. Основы работы с файлами

1. Какие классы для работы с файловой системой вы знаете?

2. Что такое сборка?

3. Как определить проект по умолчанию в решении Visual Studio?

4. Какие классы отвечают за представление файлов в программе?

5. Что такое поток? Какие типы классов потоков используются при работе с файлами?

6. Опишите последовательность действий при необходимости записать одну строку в файл.

Приведите примеры использования различных классов.

7. Перечислите классы для работы с каталогами.

8. Поясните принцип работы синтаксической конструкции try ... catch.

Лабораторная работа 11. Обработка данных в файлах

1. Дайте определение понятия «файл».

2. Опишите формат файла *.csv.

3. Поясните назначение различных конфигураций (debug или release) при сборке приложения в среде Visual Studio.

4. Приведите пример кода, реализующего различное поведение программы при

использовании различных режимов сборки (debug или release).

5. Что такое поток? Какие типы классов потоков используются при работе с файлами?

6. Опишите последовательность действий при необходимости записать одну строку в файл. Приведите примеры использования различных классов.

Лабораторная работа 12. Основы проектирования приложений Windows Forms

1. Какие файлы описывают класс формы?

2. Какие действия необходимо выполнить для создания обработчика события?

3. Где описывается код обработчика события? В каком файле регистрируется обработчик события (метод привязывается к событию)?

4. Как получить доступ к координатам курсора мыши?

5. Какой класс содержит методы, реализующие математические функции?

Лабораторная работа 13. Простейшие элементы управления и события

1. Какое событие элемента управления Button обрабатывается в программах чаще всего?

2. Для чего предназначен компонент CheckBox? Назовите основные свойства класса CheckBox.

3. Опишите назначение элемента RadioButton. Какой внешний вид может принимать данный компонент? Назовите основные свойства класса RadioButton.

4. Назовите классы компонентов для представления списочной информации.

5. Опишите основные свойства класса ListBox. Чем компонент ListBox отличается от CheckedListBox?

Лабораторная работа 14. Меню и строка состояния в приложениях Windows Forms

1. Какие элементы управления используются для создания главного меню приложения?

2. Какие элементы управления используются для создания контекстного меню элемента управления?

3. Как в режиме разработки указать элементу управления его контекстное меню?

4. Какие типы данных используются для создания пунктов меню?

5. Поясните механизм синхронизации событий от разных элементов управления.

6. Поясните назначение конструкции try ... catch ...

7. Какие классы используются для создания строки состояния в .NET?

8. Какие классы используются для создания панелей строки состояния? Каково назначение каждого из этих классов?

9. Какой класс используется для работы с таймером? Какое событие элемента-таймера необходимо обрабатывать для реагирования на смену системного времени?

10. С помощью какого класса можно получить текущие дату и время? Какие методы содержит данный класс?

11. Что позволяет настраивать свойство Spring элементов-панелей строки состояния?

Лабораторная работа 15. Диалоговые окна

1. Как создаются модальные диалоговые окна?

2. Что такое пространство имен?

3. Какие пространства имен использованы в вашей программе? Какие типы данных из этих пространств использовались?

4. Опишите механизм обработки событий для визуальных элементов управления.

5. С каким модификатором доступности создаются элементы управления формы?

6. Как возвращается информация о нажатой кнопке закрытия модального диалогового окна? Какое свойство диалогового окна используется для этого? Значение какого типа возвращает метод ShowDialog?

7. Для чего предназначен класс OpenFileDialog? Назовите основные методы и свойства класса.

8. Для чего предназначен класс SaveFileDialog? Опишите основные методы и свойства данного класса.

9. Опишите принцип работы с классом ColorDialog. Опишите основные методы и свойства данного класса.

10. Опишите принципы работы с классом FontDialog. Опишите основные методы и свойства данного класса.

Лабораторная работа 16. Обработка табличных данных

1. Какие классы для работы с файловой системой вы знаете?
2. Что такое класс потока? Перечислите классы потоков для работы с файлами?
3. Для чего используются интерфейсные типы? Приведите примеры.
4. Какие классы отвечают за представление файлов в программе?
5. Опишите последовательность действий при необходимости записать одну строку в файл.

Приведите примеры использования различных классов.

6. Опишите принципы работы с байтовым потоком. Приведите пример кода для записи и считывания файла с использованием байтового потока.
7. Чем байтовый поток отличается от символьного?

Лабораторная работа 17. Технология GDI+ в приложениях Windows Forms

1. Назовите основные типы, необходимые для работы с GDI+.
2. Какие пространства имен доступны для работы с графикой?
3. Опишите возможные конструкторы класса Point.
4. В чем состоит особенность конструктора класса Region?
5. Как получить доступ к объекту класса Graphics?
6. Опишите способ, с помощью которого можно реализовать перехват события мыши над фигурами GDI+.
7. Для чего предназначен класс GraphicPath?
8. Чем отличаются типы Point и PointF?
9. Какие элементы могут обрабатывать событие Paint?
10. Для чего используется тип PathPointType?
11. Через какой тип происходит рисование в обработчике события Paint?
12. Опишите механизм выбора объекта GDI.
13. Какие события участвуют в процессе интерактивного взаимодействия с GDI объектами графики?
14. Какое событие необходимо задействовать, если добавить в программу отображение контура объекта при перетаскивании?
15. Опишите механизм реализации анимации с использованием GDI+.
16. Опишите назначение основных свойств и методов класса Timer.
17. Для чего предназначено событие Tick класса Timer?
18. Сколько таймеров может присутствовать в программе?

Лабораторная работа 18. Приложения Windows Presentation Foundation

1. Опишите основные свойства класса Window.
2. Что такое контейнер компоновки? Для чего они используются? Опишите свойства класса Panel.
3. Назовите основные контейнеры компоновки. Дайте характеристику и опишите особенности контейнеров компоновки.
4. Что такое присоединенное свойство? Приведите примеры.
5. Какой контейнер применяется в окне приложения как контейнер компоновки по умолчанию?
6. Какие свойства элементов управления влияют на положение элементов внутри контейнера компоновки?

Лабораторная работа 19. Ресурсы, стили, триггеры

1. Что такое стиль?
2. Что такое ресурс?
3. Что такое триггер?
4. Опишите механизмы применения стилей.
5. Опишите механизмы применения триггеров.
6. Какие типы ресурсов вы знаете?

7. У каких элементов WPF может быть коллекция ресурсов?

Лабораторная работа 20. Механизм привязки WPF

1. Что такое привязка WPF?
2. Поясните синтаксис привязки. Какие ключевые слова и для каких целей используются при оформлении привязки с помощью XAML?
3. Какие режимы привязки существуют? Поясните назначение всех режимов привязки.
4. Поясните схему привязки элементов управления. Назовите основные элементы данной схемы.
5. Что означают понятия «целевой элемент», «элемент-источник привязки», «целевое свойство», «свойство-источник»?
6. Какое ограничение на источник привязки существует в WPF?

Лабораторная работа 21. Источники привязки произвольного типа

1. Чем отличается привязка к элементам управления от привязки к произвольным объектам?
2. Поясните синтаксис привязки к данным. Какие ключевые слова и для каких целей используются при оформлении привязки с помощью XAML?
3. Какие режимы привязки существуют? Поясните назначение всех режимов привязки.
4. Поясните схему привязки элементов управления. Назовите основные элементы данной схемы.
5. Что означают понятия «целевой элемент», «элемент-источник привязки», «целевое свойство», «свойство-источник»?
6. Какое ограничение на источник привязки существует в WPF?
7. Какое ограничение существует на целевое свойство привязки?

Лабораторная работа 22. Асинхронные делегаты

1. Что такое тип делегата? Какой аналог типа делегата существует в C++?
2. Опишите основные направления использования делегатов.
3. Какие механизмы технологии Windows Forms реализованы с использованием делегатов?
4. Для чего предназначен тип Action<T>? Чем он отличается от Func<T>?
5. Чем пользовательские делегаты отличаются от библиотечных?
6. Поясните назначение типа IAsyncResult.
7. Для чего используется метод Thread.Sleep()?
8. Поясните механизм возврата значения из метода асинхронного делегата.
9. Как произвести возврат более одного значения из метода?
10. Какая разница существует между библиотечными делегатами, пользовательскими типами делегатов и лямбда-выражениями? Являются ли эти делегаты взаимозаменяемыми при реализации асинхронного вызова методов?

Лабораторная работа 23. Передача данных потокам

1. Для чего применяется тип IAsyncResult?
2. Как реализовать ожидание завершения выполнения асинхронного метода с использованием тайм-аута?
3. Поясните назначение метода WaitOne().
4. Опишите возможные пути передачи параметров в поток.
5. Подумайте, можно ли передать в несколько потоков один и тот же параметр (ссылку на объект)? Ответ обоснуйте.

Лабораторная работа 24. Разработка многомодульных приложений

1. Что такое статический класс?
2. Что такое сборка?
3. Как определить проект по умолчанию в многомодульном решении?
4. Какими способами можно разместить в файлах определение класса?
5. Какой проект начнет выполняться первым если несколько из них в одном решении содержат функцию Main?

6. Что необходимо сделать для того, чтобы появилась возможность использовать классы, определенные в другом проекте?

1. Разработка приложения «Интерактивная карта»
 2. Разработка приложения «Менеджер фотографий»
 3. Разработка приложения «Траектория движения»
 4. Разработка приложения «Игра «Собиратель монет»
 5. Разработка приложения «Свертка изображения»
 6. Разработка приложения «Матричный калькулятор»
 7. Разработка приложения «Аквалайзер»
 8. Разработка приложения «Графический пароль»
 9. Разработка приложения «Игра «Танки»
 10. Разработка приложения «Игра «Змейка»
 11. Разработка приложения «Расчет стоимости поездки»
 12. Разработка приложения «Расчет баллистической траектории»
 13. Разработка приложения «Интерактивный анализ набора данных»
 14. Разработка приложения «Моделирование популяции»
 15. Разработка приложения «Обработка изображения фильтрами»
 16. Разработка приложения «Игра «Цветы против грибов»
 17. Разработка приложения «Фракталы»
 18. Разработка приложения «Искусственный интеллект»
 19. Разработка приложения «Филиалы и сотрудники»
 20. Разработка приложения «Игра «Ракета»
 21. Разработка приложения «Игра «Блиц»
 22. Разработка приложения «Поиск текста в документах»
 23. Разработка приложения «Анализ матричных данных»⁷
 24. Разработка приложения «Игра «Поиск крота»
 25. Разработка приложения «Вычисления со скобками»
1. История развития технологий программирования Windows-приложений
 2. Обобщенная структура .NET Framework
 3. Основные понятия, механизмы и соглашения технологии .NET
 4. Особенности выполнения приложений .NET
 5. Общие сведения о системе типов .NET
 6. Компоновочные блоки, пространства имен, типы
 7. Спецификаторы доступа типов и членов типов
 8. Объявление и инициализация переменных и констант в C#
 9. Условный оператор
 10. Оператор выбора
 11. Оператор цикла for
 12. Цикл с предусловием while
 13. Цикл с постусловием do ... while
 14. Простые массивы
 15. Многомерные массивы
 16. Зубчатые массивы
 17. Классы и структуры
 18. Данные-члены класса
 19. Функции-члены класса
 20. Методы классов
 21. Операции. Перегрузка операций
 22. Наследование реализации
 23. Наследование интерфейсов
 24. Интерфейсы и типы коллекций
 25. Списки
 26. Очередь

27. Стек
28. Классы .NET Framework для реализации операций ввода-вывода
29. Классы для работы с каталогами файловой системы
30. Классы для работы с файлами
31. Потоки в системе ввода-вывода
32. Байтовый поток
33. Символьный поток
34. Делегаты
35. Делегаты Action и Func
36. Групповые делегаты
37. Анонимные методы
38. Лямбда-выражения
39. События

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

основная

Л1.1 Тюкачев Н. А., Хлебостроев В. Г. С#. Основы программирования [Электронный ресурс]:учеб. пособие ; ВО - Бакалавриат. - Санкт-Петербург: Лань, 2021. - 272 с. – Режим доступа: <https://e.lanbook.com/book/158960>

Л1.2 Тюкачев Н. А., Хлебостроев В. Г. С#. Алгоритмы и структуры данных. + Электронное приложение [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат, Магистратура, Специалитет. - Санкт-Петербург: Лань, 2022. - 232 с. – Режим доступа: <https://e.lanbook.com/book/229133>

Л1.3 Хорев П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат. - Москва: Издательство "ФОРУМ", 2023. - 200 с. – Режим доступа: <https://znanium.com/catalog/document?id=424788>

дополнительная

Л2.1 Сыромятников В. П. Структуры и алгоритмы обработки данных [Электронный ресурс]:практикум; ВО - Бакалавриат. - Москва: РТУ МИРЭА, 2020. - 244 с. – Режим доступа: <https://e.lanbook.com/book/163915>

Л2.2 Рацеев С. М. Программирование. Лабораторный практикум [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат. - Санкт-Петербург: Лань, 2023. - 104 с. – Режим доступа: <https://e.lanbook.com/book/292907>

Л2.3 Барков И. А. Объектно-ориентированное программирование [Электронный ресурс]:учебник; ВО - Бакалавриат, Магистратура. - Санкт-Петербург: Лань, 2023. - 700 с. – Режим доступа: <https://e.lanbook.com/book/329549>

Л2.4 Белугина С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование [Электронный ресурс]:учеб. пособие для СПО. - Санкт-Петербург: Лань, 2023. - 312 с. – Режим доступа: <https://e.lanbook.com/book/296975>

б) Методические материалы, разработанные преподавателями кафедры по дисциплине, в соответствии с профилем ОП.

Л3.1 Антамошкин О. А. Программная инженерия. Теория и практика [Электронный ресурс]:учебник ; ВО - Бакалавриат, Магистратура, Специалитет. - Красноярск: Сибирский федеральный университет, 2012. - 247 с. – Режим доступа: <http://new.znanium.com/go.php?id=492527>

Л3.2 Турнецкая Е. Л., Аграновский А. В. Программная инженерия. Интеграционный подход к разработке [Электронный ресурс]:учебник; ВО - Бакалавриат. - Санкт-Петербург: Лань, 2023. - 216 с. – Режим доступа: <https://e.lanbook.com/book/352307>

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Наименование ресурса сети «Интернет»	Электронный адрес ресурса
1	Руководство по программированию в Windows Forms	https://metanit.com/sharp/windowsforms/
2	Руководство по программированию WPF	https://metanit.com/sharp/wpf/
3	Полное руководство по языку программирования C# 12 и платформе .NET 8	https://metanit.com/sharp/tutorial/

10. Методические указания для обучающихся по освоению дисциплины

Методические рекомендации по освоению дисциплины «Программная инженерия»
Методические рекомендации по выполнению курсовой работа (проекта).

Лекционные занятия

Основа освоения дисциплины – лекция, целью которой является целостное и логичное рассмотрение основного материала курса. Вместе с тем значимость лекции определяется тем, что она не только способствует выработке логического мышления, но и способствует развитию интереса к пониманию современной действительности.

Задача студентов в процессе умелой и целеустремленной работы на лекциях – внимательно слушать преподавателя, следить за его мыслью, предлагаемой системой логических посылок, доказательств и выводов, фиксировать (записывать) основные идеи, важнейшие характеристики понятий, теорий, наиболее существенные факты. Лекция задает направление, содержание и эффективность других форм учебного процесса, нацеливает студентов на самостоятельную работу и определяет основные ее направления (подготовку к практическим занятиям, выполнение творческих заданий, рефератов, решение контекстных задач).

Записывание лекции – творческий процесс. Запись лекции крайне важна. Это позволяет надолго сохранить основные положения лекции; способствует поддержанию внимания; способствует лучшему запоминанию материала. Важно уметь оформить конспект так, чтобы важные моменты были выделены графически, а главную информацию следует выделять в самостоятельные абзацы, фиксируя ее более крупными буквами или цветными маркерами. Конспект должен иметь поля для заметок. Это могут быть библиографические ссылки и, наконец, собственные комментарии. Для быстрой записи текста можно придумать условные знаки, при этом таких знаков не должно быть более 10–15. Условные обозначения придумывают для часто встречающихся слов (существует, который, каждый, точка зрения, на основании и т.п.).

Перед каждой лекцией необходимо внимательно прочитать материал предыдущей лекции. В рабочей тетради графически выделить: тему лекции, основные теоретические положения. Подготовленный студент легко следит за мыслью преподавателя, что позволяет быстрее запоминать новые понятия, сущность которых выявляется в контексте лекции. Повторение материала облегчает в дальнейшем подготовку к экзамену. Затем надо ознакомиться с материалом темы по учебнику, внести нужные уточнения и дополнения в лекционный материал. После усвоения каждой темы рекомендуется проверять свои знания, отвечая на контрольные вопросы по теме.

Лабораторные занятия

Целью лабораторных занятий является закрепление, расширение, углубление теоретических знаний, полученных на лекциях и в ходе самостоятельной работы, развитие познавательных способностей.

Являясь частью образовательного процесса, семинар преследует ряд основополагающих задач:

- работа с источниками, которая идет на уровнях индивидуальной самостоятельной работы и в ходе коллективного обсуждения;

- формирование умений и навыков индивидуальной и коллективной работы, позволяющих эффективно использовать основные методы исследования, грамотно выстраивать его основные технологические этапы (знакомство с темой и имеющейся по ней информацией, определение основной проблемы, первичный анализ, определение подходов и ключевых узлов механизма ее развития, публичное обсуждение, предварительные выводы);

- анализ поставленных проблем, умение обсуждать тему, высказывать свое мнение, отстаивать свою позицию, слушать и оценивать различные точки зрения, конструктивно полемизировать, учиться думать, говорить, слушать, понимать, находить точки соприкосновения разных позиций, их разумного сочетания;

- формирование установок на творчество;

- диалог, внутренний и внешний; поиск и разрешение проблемы в рамках имеющейся о ней информации;

- поиск рационального зерна в самых противоречивых позициях и подходах к проблеме;

- открытость новому и принципиальную возможность изменить свою позицию и вытекающие из нее решения, в случае получения новой информации и связанных с ней обстоятельств сознательный отход от подготовленного к семинару текста во время своего, построенного на тезисном изложении фактов и мыслей, когда конспект привлекается лишь в том случае, когда надо привести какие-то факты.

Для эффективной работы на практическом занятии студенту необходимо учесть и выполнить следующие требования по подготовке к нему:

1. Внимательно прочитать, как сформулирована тема, определить ее место в учебном плане курса, установить взаимосвязи с другими разделами.

2. Познакомиться с целью и задачами работы на практическом занятии, обратив внимание на то, какие знания, умения и навыки студент должен приобрести в результате активной познавательной деятельности.

3. Проработать основные вопросы и проблемы (задания), которые будут рассматриваться и обсуждаться в ходе практического занятия.

4. Подобрать литературу по теме занятия; найти соответствующий раздел в лекциях и в рекомендуемых пособиях.

5. Добросовестно проработать имеющуюся научную литературу (просмотреть и подобрать информацию, сделать выписки (конспектирование узловых проблем), обработать их в соответствии с задачами практического занятия.

6. Обдумать и предложить свои выводы и мысли на основании полученной информации (предварительное осмысление).

7. Продумать развернутые законченные ответы на предложенные вопросы, предлагаемые творческие задания и контекстные задачи, опираясь на материал лекций, расширяя и дополняя его данными из учебника, дополнительной литературы, составить план ответа, выписать терминологию.

Видами заданий на практических занятиях:

- для овладения знаниями: чтение текста (учебника, первоисточника, дополнительной литературы), работа со словарями и справочниками, ознакомление с нормативными документами, учебно-исследовательская работа, использование аудио- и видеозаписей, компьютерной техники и Интернета и др.

- для закрепления и систематизации знаний: работа с конспектом лекции, обработка текста, повторная работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио и видеозаписей, ответы на контрольные вопросы, аналитическая обработка текста, подготовка мультимедиа сопровождения к защите рефератов, и др.

- для формирования умений: решение контекстных задач, подготовка к деловым играм, выполнение творческих заданий, анализ профессиональных умений с использованием аудио- и видеотехники и др.

Работа с научной и учебной литературой

Важнейшим средством информации, распространения знаний является книга. Работа с книгой состоит в том, чтобы облегчить специалистам возможность добывать из книги необходимые знания, отобрать нужную информацию наиболее эффективно и при возможно меньших затратах времени.

Приступая к изучению дисциплины необходимо внимательно просмотреть список основной и дополнительной литературы, определить круг поиска нужной информации. Если книг на одну тему несколько, то необходимо, прежде всего, просмотреть их, ознакомиться с оглавлением, содержанием предисловием, аннотацией или введением, характером и стилем изложения материала. Выбор необходимой литературы и периодики осуществляется самостоятельно, так как даже опытный библиограф не в состоянии учесть индивидуальные интересы.

Обучающийся должен внимательно изучить электронные каталоги и картотеки. Лаконичные каталожные карточки несут богатую информацию: фамилия автора, название книги, его подзаголовок, научное учреждение, подготовившее издание, название издательства, год выхода книги, количество страниц. Обязательный справочный материал поможет вам в подборе необходимой литературы.

Изучение книги целесообразно начинать с предварительного знакомства с ней: просмотреть введение, оглавление, заключение, библиографию или список использованной литературы. Во введении или предисловии автор обычно формулирует задачи, которые ставятся в книге. Внимательно изучив оглавление, студент узнает общий план книги, содержание ее, а в научных трудах и основные мысли автора. К оглавлению полезно обращаться не только при предварительном знакомстве с книгой, но и в процессе повторного и выборочного чтения, завершения его.

После предварительного знакомства с книгой следует приступить к первому чтению, главная цель которого - понять содержание в целом. Это предварительное чтение - знакомство с книгой и выделение в ней всего того, что наиболее существенно и требует детальной проработки в другое время.

Следующим этапом является повторное чтение или чтение с проработкой материала - это критический разбор читаемого с целью глубокого проникновения в его сущность, конспектирования.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства и информационных справочных систем (при необходимости).

11.1 Перечень лицензионного программного обеспечения

1. Kaspersky Total Security - Антивирус
2. Microsoft Windows Server STDCORE AllLngLicense/Software AssurancePack Academic OLV 16Licenses LevelE AdditionalProduct CoreLic 1Year - Серверная операционная система

11.3 Перечень программного обеспечения отечественного производства

1. Kaspersky Total Security - Антивирус

При осуществлении образовательного процесса студентами и преподавателем используются следующие информационно справочные системы: СПС «Консультант плюс», СПС «Гарант».

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование специальных помещений и помещений для самостоятельной работы	Номер аудитор ии	Оснащенность специальных помещений и помещений для самостоятельной работы
-------	---	------------------	---

1	Учебная аудитория для проведения занятий всех типов (в т.ч. лекционного, семинарского, практической подготовки обучающихся), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Э-160	Специализированная мебель на 180 посадочных мест, персональный компьютер – 1 шт., проектор Panasonic EX620 X6A – 1 шт., интерактивная доска SMART Board 690 – 1 шт., трибуна для лектора – 1 шт., микрофон – 1 шт., мониторы - 3 шт., плазменная панель - 1 шт., учебно-наглядные пособия в виде презентаций, информационные плакаты, подключение к сети «Интернет», выход в корпоративную сеть университета.
		Э-159	Специализированная мебель на 26 посадочных мест, персональный компьютер – 26 шт., телевизор – 1 шт., учебно-наглядные пособия в виде презентаций, подключение к сети «Интернет», доступ в электронную информационно-образовательную среду университета, выход в корпоративную сеть университета.
2	Помещение для самостоятельной работы обучающихся, подтверждающее наличие материально-технического обеспечения, с перечнем основного оборудования		

13. Особенности реализации дисциплины лиц с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- задания для выполнения, а также инструкция о порядке проведения промежуточной аттестации оформляются увеличенным шрифтом;

- задания для выполнения на промежуточной аттестации зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- промежуточная аттестация проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по желанию студента промежуточная аттестация может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента промежуточная аттестация проводится в устной форме.

Рабочая программа дисциплины «Программная инженерия» составлена на основе Федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии (приказ Минобрнауки России от 19.09.2017 г. № 926).

Автор (ы)

_____ доц. , ктн Николаев Е.И.

Рецензенты

_____ доц. , ктн Шлаев Д.В.

_____ доц. , кэн Сорокин А.А.

Рабочая программа дисциплины «Программная инженерия» рассмотрена на заседании Кафедра инжиниринга IT-решений протокол № 1 от 26.08.2024 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии

Заведующий кафедрой _____ Шлаев Д.В.

Рабочая программа дисциплины «Программная инженерия» рассмотрена на заседании учебно-методической комиссии Факультет цифровых технологий протокол № от г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии

Руководитель ОП _____