

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Луганский государственный университет
имени Владимира Даля»
(ФГБОУ ВО «ЛГУ им. В. Даля»)

Краснодонский факультет инженерии и менеджмента (филиал)
Кафедра информационных технологий и транспорта



УТВЕРЖДАЮ:

Директор

Панайотов К.К.

«21» апреля 2023 года

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

По дисциплине Практикум по программированию
(название дисциплины по учебному плану)

По направлению подготовки 09.03.01 – Информатика и вычислительная техника
(код, название без кавычек)

Профиль подготовки Компьютерные системы и сети

Краснодон 2023

Лист согласования РПУД

Рабочая программа учебной дисциплины «Практикум по программированию» по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиль «Компьютерные системы и сети» – 23 с.

Рабочая программа учебной дисциплины «Практикум по программированию» разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 года № 929.

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):

к.т.н., доц. Бихдрикер А.С.

(ученая степень, ученое звание, должность фамилия, инициалы)

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий и транспорта «15» марта 2023 г., протокол № 7.

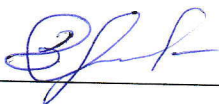
Заведующий кафедрой



Бихдрикер А.С.

Рекомендована на заседании учебно-методической комиссии факультета «20» марта 2023 г., протокол № 8.

Председатель учебно-методической
комиссии факультета



Замота О.Н.

Структура и содержание дисциплины

1. Цели и задачи учебной дисциплины, ее место в учебном процессе

Цель изучения дисциплины – получение практических навыков написания программ для решения прикладных задач, приобретение рациональных качеств мысли, чутья объективности, интеллектуальной честности.

Задачи: сформировать навыками применения алгоритмических языков высокого уровня при решении широкого круга практических задач.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Практикум по программированию» относится к части, формируемой участниками образовательных отношений по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Дисциплина реализуется кафедрой информационных технологий и транспорта.

Основывается на базе дисциплин: программирование; компьютерная логика.

Является основой для изучения следующих дисциплин: структуры и алгоритмы компьютерной обработки данных; архитектура и программирование мобильных устройств; системное программирование; Веб-программирование; проблемно-ориентированные вычислительные системы.

3. Требования к результатам освоения содержания дисциплины

Код и наименование компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1 Способен выбирать современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности ОПК-2.2 Способен выбирать современные информационные технологии, технические и программные средства при решении задач профессиональной деятельности	Знать: понятия объектно-ориентированного программирования: абстрагирование, инкапсуляция, агрегирование, наследование, объектная модель программы; принципы организации проектирования и содержание этапов процесса разработки программных комплексов; методы структурного и объектно-ориентированного программирования; платформу Microsoft .Net Framework для разработки и выполнения программ в операционной системе Windows; компонентный, объектно-ориентированный язык программирования. Уметь: разрабатывать и отлаживать программы с использованием современных технологий программирования; проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к ИС; применять основные принципы объектно-ориентированного программирования, принцип построения классов, критерии проверки правильности построения классов, основные тенденции в области развития

		технологий объектно-ориентированного программирования; разрабатывать объектно-ориентированные модели прикладных программ; выполнять отладку и тестирование научно-прикладных программ. Владеть: навыками программирования в современных средах; навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов; современными методами объектно-ориентированного программирования при кодировании программных систем разного уровня сложности; навыками постановки задач комплексного анализа, связанных с созданием новых информационных технологий и информационных систем; разработкой новых и модернизацией уже существующих информационных технологий и систем в соответствии с техническим заданием.
--	--	---

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (зач. ед.)		
	Очная форма	Очно-заочная форма	Заочная форма
Общая учебная нагрузка (всего)	144 (4 зач. ед)	-	144 (4 зач. ед)
Обязательная аудиторная учебная нагрузка (всего)	68	-	12
в том числе:			
Лекции	17	-	2
Семинарские занятия	-	-	-
Практические занятия	-	-	-
Лабораторные работы	51	-	10
Курсовая работа (курсовой проект)	-	-	-
Другие формы и методы организации образовательного процесса (<i>расчетно-графические работы, групповые дискуссии, ролевые игры, тренинг, компьютерные симуляции, интерактивные лекции, семинары, анализ деловых ситуаций и т.п.</i>)	-	-	-
Самостоятельная работа студента (всего)	76	-	132
Форма аттестации	зачёт, зачёт	-	зачёт, зачёт

4.2. Содержание разделов дисциплины:

Семестр 1

Тема 1. ОПИСАНИЕ КЛАССОВ

Основные элементы классов. Режимы доступа. Поля класса. Инициализация объектов класса.

Тема 2. МЕТОДЫ КЛАССА

Тело метода. Вызов метода. Конструкторы класса. Деструкторы. Передача объектов методам по ссылке. Использование модификаторов параметров ref и out. Использование переменного числа аргументов. Возврат объектов из методов.

Тема 3. ПЕРЕГРУЗКА МЕТОДОВ

Перегрузка методов. Перегрузка конструкторов. Инициализаторы объектов. Необязательные аргументы. Именованные аргументы. Метод Main (). Применение ключевого слова static. Статические классы.

Тема 4. ПЕРЕГРУЗКА ОПЕРАТОРОВ

Основы перегрузки операторов. Выполнение операций со встроенными в C# типами данных. Перегрузка операторов отношения. Перегрузка операторов true и false. Перегрузка логических операторов. Рекомендации и ограничения по перегрузке операторов.

Тема 5. ИНДЕКСАТОРЫ

Создание одномерных индексаторов. Перегрузка индексаторов. Индексаторы без базового массива. Многомерные индексаторы.

Тема 6. СВОЙСТВА

Автоматически реализуемые свойства. Применение инициализаторов объектов в свойствах. Ограничения, присущие свойствам. Применение модификаторов доступа в аксессорах. Применение индексаторов и свойств.

Семестр 2

Тема 7. НАСЛЕДОВАНИЕ

Основы наследования. Доступ к членам класса и наследование. Конструкторы и наследование. Наследование и сокрытие имен. Создание многоуровневой иерархии классов. Порядок вызова конструкторов. Ссылки на базовый класс и объекты производных классов.

Тема 8. ВИРТУАЛЬНЫЕ МЕТОДЫ

Виртуальные методы и их переопределение. Применение абстрактных классов. Предотвращение наследования с помощью ключевого слова sealed. Класс object.

Тема 9. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Класс System.Exception. Основы обработки исключительных ситуаций. Последствия не перехвата исключений. Обработка исключительных ситуаций - "изящный" способ устранения программных ошибок. Применение нескольких операторов catch. Перехват всех исключений. Вложение блоков try. Генерирование исключений вручную. Использование блока finally. Подробное рассмотрение класса Exception. Получение производных классов исключений. Применение ключевых слов checked и unchecked.

Тема 10. ПРИМЕНЕНИЕ СРЕДСТВ ВВОДА-ВЫВОДА

Организация системы ввода-вывода в C# на потоках. Классы потоков. Консольный ввод-вывод. Класс FileStream и байтовый ввод-вывод в файл. Символьный ввод-вывод в файл. Переадресация стандартных потоков. Файлы с произвольным доступом. Применение класса MemoryStream. Применение классов StringReader и StringWriter. Класс File. Преобразование числовых строк в их внутреннее представление.

Тема 11. СОБЫТИЯ

Примеры событий. Методы экземпляра в сравнении со статическими методами в качестве обработчиков событий. Применение асессоров событий. Применение анонимных

методов и лямбда-выражений вместе с событиями. Рекомендации по обработке событий в среде .NET Framework.

4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1	Описание классов	2	-	0,1
2	Методы класса	2	-	0,1
3	Перегрузка методов	2	-	0,2
4	Перегрузка операторов	2	-	0,2
5	Индексаторы	2	-	0,2
6	Свойства	2	-	0,2
7	Наследование	1	-	0,2
8	Виртуальные методы	1	-	0,2
9	Обработка исключительных ситуаций	1	-	0,2
10	Применение средств ввода-вывода	1	-	0,2
11	События	1	-	0,2
Итого:		17	-	2

4.4. Практические занятия

Планом не предусмотрены.

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1	Создание класса	3	-	0,5
2	Методы класса	3	-	0,5
3	Конструкторы класса.	3	-	0,5
4	Передача объектов методам по ссылке	3	-	0,6
5	Перегрузка методов. Перегрузка конструкторов	3	-	0,5
6	Статические классы	3	-	0,6
7	Перегрузка операторов	3	-	0,6
8	Индексаторы	3	-	0,5
9	Свойства	3	-	0,5
10	Наследование	3	-	0,5
11	Виртуальные методы	3	-	0,6
12	Интерфейсы	3	-	0,6
13	Структуры и перечисления	3	-	0,5
14	Обработка исключительных ситуаций	2	-	0,5
15	Применение средств ввода-вывода	2	-	0,5
16	Делегаты	2	-	0,5
17	Лямбда-выражения	2	-	0,5
18	События	2	-	0,5
19	Пространства имён, препроцессор, сборки	2	-	0,5
Итого:		51	-	10

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	Очно- заочная форма	Заочная форма
1	Описание классов	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	6		12
2	Методы класса	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
3	Перегрузка методов	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
4	Перегрузка операторов	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
5	Индексаторы	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
6	Свойства	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
7	Наследование	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
8	Виртуальные методы	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
9	Обработка исключительных ситуаций	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12

10	Применение средств ввода-вывода	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
11	События	Подготовка к лабораторным работам, практическим занятиям, к текущему и промежуточному контролю знаний и умений.	7		12
Итого:			76		132

4.7. Курсовые работы/проекты.

Планом не предусмотрены

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

Информационные технологии: использование электронных образовательных ресурсов (электронный конспект, размещенный во внутренней сети, или т.п.) при подготовке к лекциям, практическим занятиям.

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Зайцев М.Г., Объектно-ориентированный анализ и программирование : учебное пособие / Зайцев М.Г. - Новосибирск : Изд-во НГТУ, 2017. - 84 с. - ISBN 978-5-7782-3308-9 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785778233089.html>

2. Иванова Г.С., Технология программирования / Иванова Г.С. - М. : Издательство МГТУ им. Н. Э. Баумана, . - 336 с. (Сер. Информатика в техническом университете) - ISBN 5-7038-2891-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN5703828910.html>

3. Подбельский В.В., Язык C#. Базовый курс : учеб. пособие / В.В. Подбельский. - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2015. - 408 с. - ISBN 978-5-279-03534-2 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785279035342.html>

4. Самохвалов Э.Н., Введение в проектирование и разработку приложений на языке программирования C# : учебное пособие / Э.Н. Самохвалов, Г.И. Ревунков, Ю.Е. Гапанюк - М. : Издательство МГТУ им. Н. Э. Баумана, 2018. - 244 с. - ISBN 978-5-7038-4553-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785703845530.html>

б) дополнительная литература:

1. Комлев Н.Ю., Объектно Ориентированное Программирование. Настольная книга программиста / Комлев Н. Ю. - М. : СОЛОН-ПРЕСС, 2018. - 298 с. - ISBN 978-5-91359-276-7 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785913592767.html>

2. Лисицин Д.В., Объектно-ориентированное программирование / Лисицин Д.В. - Новосибирск : Изд-во НГТУ, 2010. - 88 с. - ISBN 978-5-7782-1454-5 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785778214545.html>

3. Суханов М.В., Основы Microsoft .NET Framework и языка программирования C# : / Суханов М.В. - Архангельск : ИД САФУ, 2014. - 96 с. - ISBN 978-5-261-00934-4 - Текст :

электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785261009344.html>

4. Уйманова Н.А., Основы объектно-ориентированного программирования / Уйманова Н.А. - Оренбург: ОГУ, 2017. - 768 с. - ISBN 978-5-7410-1993-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785741019931.html>

в) интернет-ресурсы:

Министерство образования и науки Российской Федерации – <http://минобрнауки.рф/>
Федеральная служба по надзору в сфере образования и науки – <http://obrnadzor.gov.ru/>

Министерство образования и науки Луганской Народной Республики – <https://minobr.su>

Народный совет Луганской Народной Республики – <https://nslnr.su>

Портал Федеральных государственных образовательных стандартов высшего образования – <http://fgosvo.ru>

Федеральный портал «Российское образование» – <http://www.edu.ru/>

Информационная система «Единое окно доступа к образовательным ресурсам» – <http://window.edu.ru/>

Федеральный центр информационно-образовательных ресурсов – <http://fcior.edu.ru/>

Электронные библиотечные системы и ресурсы

Электронно-библиотечная система «Консультант студента» – <http://www.studentlibrary.ru/cgi-bin/mb4x>

Электронно-библиотечная система «StudMed.ru» – <https://www.studmed.ru>

Информационный ресурс библиотеки образовательной организации

Научная библиотека имени А. Н. Коняева – <http://biblio.dahluniver.ru/>

7. Материально-техническое обеспечение дисциплины

Освоение дисциплины «Программирование» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Прочее:

- рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
- для проведения лекционных занятий требуется аудитория на курс, оборудованная мультимедийным проектором с экраном;
- для проведения лабораторных работ требуется компьютерный класс, подключенный к Интернет.

Программное обеспечение:

Функциональное назначение	Бесплатное программное обеспечение	Ссылки
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Почтовый клиент	Mozilla Thunderbird	http://www.mozilla.org/ru/thunderbird
Файл-менеджер	Far Manager	http://www.farmanager.com/download.php
Архиватор	7Zip	http://www.7-zip.org/
Графический редактор	GIMP (GNU Image Manipulation Program)	http://www.gimp.org/ http://gimp.ru/viewpage.php?page_id=8 http://ru.wikipedia.org/wiki/GIMP
Редактор PDF	PDFCreator	http://www.pdfforge.org/pdfcreator
Аудиоплеер	VLC	http://www.videolan.org/vlc/

8. Оценочные средства по учебной дисциплине
Паспорт фонда оценочных средств по учебной дисциплине
Практикум по программированию
(наименование учебной дисциплины)

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемо й компетенции	Формулировка контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Контролируемы е разделы (темы) учебной дисциплины (модуля), практики	Этапы формировани я (семестр изучения)
1.	ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессионально й деятельности	ОПК-2.1 Способен выбирать современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессионально й деятельности ОПК-2.2 Способен выбирать современные информационные технологии, технические и программные средства при решении задач профессиональной деятельности	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8. Тема 9. Тема 10. Тема 11.	2,3

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Индикаторы достижений компетенций (по реализуемой дисциплине)	Перечень планируемых результатов	Контролируемые разделы (темы) учебной дисциплины (модуля), практики	Наименование оценочного средства
1	ОПК-2	ОПК-2.1 ОПК-2.2	Знать: понятия объектно-ориентированного программирования : абстрагирование, инкапсуляция, агрегирование, наследование, объектная модель программы; принципы организации проектирования и содержание этапов процесса разработки программных комплексов; методы структурного и объектно-ориентированного программирования ; платформу Microsoft .Net Framework для разработки и выполнения программ в операционной системе Windows; компонентный, объектно-ориентированный язык программирования . Уметь: разрабатывать и отлаживать эффективные алгоритмы и программы с	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8. Тема 9. Тема 10. Тема 11.	Собеседование (устный или письменный опрос), контрольная работа, тесты

			использованием современных технологий программирования ; проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к ИС; применять основные принципы объектно-ориентированного программирования , принцип построения классов, критерии проверки правильности построения классов, основные тенденции в области развития технологий объектно-ориентированного программирования ; разрабатывать объектно-ориентированные модели прикладных программ; выполнять отладку и тестирование научно-прикладных программ. Владеть: навыками программирования в современных средах; навыками работы с инструментальными средствами моделирования предметной		
--	--	--	--	--	--

			области, прикладных и информационных процессов; современными методами объектно- ориентированного программирования при кодировании программных систем разного уровня сложности; навыками постановки задач комплексного анализа, связанных с созданием новых информационных технологий и информационных систем; разработкой новых и модернизацией уже существующих информационных технологий и систем в соответствии с техническим заданием.		
--	--	--	---	--	--

Фонды оценочных средств по дисциплине «Практикум по программированию»

Перечень вопросов для собеседования (устный или письменный опрос)

1. Основные принципы технологии .NET.
2. Что представляет собой платформа Visual Studio.NET?
3. Как создать консольное приложение?
4. Принципы объектно-ориентированного программирования.
5. Что представляют собой методы?
6. Как объявляется метод?
7. Какова область действия параметров метода?
8. Как вызываются методы?
9. Общие (статические) методы класса.
10. Как описываются классы в C#?
11. Что относится к членам класса?
12. Что такое статические члены класса?
13. Данные: поля и константы.
14. Спецификаторы полей и констант класса.
15. Как передаются параметры в методы?
16. Для чего предназначен параметр params?

17. Что представляет собой конструктор? Для чего он используется?
18. Какие бывают конструкторы?
19. Может ли класс не иметь конструктора?
20. Для чего предназначена система сбора мусора?
21. Что понимается под массивом?
22. Каковы возможные способы описания массивов (одномерных и многомерных)?
23. В каких случаях целесообразно описывать двумерный массив с помощью одномерных?
24. Какие типы допустимы для описания индексов массивов?
25. Какие типы могут использоваться в качестве базовых для описания массивов?
26. Как осуществляется ввод и вывод массивов?
27. Для чего предназначен цикл foreach?
28. Можно ли использовать цикл foreach для ввода элементов массива?
29. Как определяется базовый тип индексатора?
30. Что записывается в качестве имени индексатора?
31. Что содержит список параметров индексатора?
32. Что представляет собой перегрузка методов?
33. Что представляет собой перегрузка операций
34. Формат описания операции класса.
35. Какие операции нельзя перегружать?
36. Что является результатом перегрузки унарных операций?
37. Какие параметры могут быть у бинарных операций класса?
38. Как выполняется перегрузка операций отношения?
39. Чем являются строки в C#?
40. Какие операции определены для строк?
41. Как создаются строки?
42. Можно ли изменять значение строки?
43. В чем состоит принцип наследования?
44. Какие члены класса наследуются?
45. Что представляет собой защищенный доступ?
46. Как происходит вызов конструкторов базового класса?
47. Что такое сокрытие имен при наследовании?
48. Как получить доступ к сокрытому члену базового класса?
49. Что означает принцип полиморфизма?
50. Для чего используется позднее связывание?
51. В каких случаях используются виртуальные методы?
52. Какие условия необходимо соблюдать при переопределении виртуального метода?
53. Что представляют собой абстрактные классы? Для чего они предназначены?
54. Могут ли в абстрактном классе быть неабстрактные методы?
55. Как описывается интерфейс? Его назначение.
56. Какие члены может содержать интерфейс?
57. Какие спецификаторы допустимы у методов, реализующих интерфейс?
58. В каких случаях используется явная реализация интерфейса?
59. Как осуществляется наследование интерфейсов?
60. Можно ли явно реализованные методы объявлять виртуальными?
61. Можно ли повторно реализовать интерфейс, указав его имя в списке предков класса наряду с классом-предком?
62. Какие стандартные интерфейсы используются для работы с коллекциями?
63. Чем отличаются интерфейсы IComparable и IComparer?
64. Перечислите основные характеристики программ.

65. Приведите существующую классификацию программного обеспечения.
66. Дайте определение и перечислите основные характеристики системного программного обеспечения.
67. Дайте определение и перечислите основные характеристики прикладного программного обеспечения.
68. Дайте определение и охарактеризуйте инструментарий технологии программирования.
69. Расскажите об особенностях создания программного продукта.
70. Что такое жизненный цикл программного обеспечения?
71. Каковы основные свойства каскадной (итерационной) модели жизненного цикла?
72. Из каких этапов состоит модель жизненного цикла UML?
73. Какова стоимость исправления ошибок в ПО на различных стадиях его разработки?
74. Что такое «управление требованиями»?
75. В чем заключается анализ проблемы?
76. Какие виды ограничений на создаваемое ПО необходимо выявить в процессе работы над требованиями?
77. Каковы существующие методы выявления требований к ПО?
78. Приведите эксплуатационные требования к ПО.
79. Перечислите функциональные требования к ПО.
80. Чем определяется выбор архитектуры ПО?
81. Охарактеризуйте статические и полустатические структуры данных.
82. Охарактеризуйте динамические структуры данных.
83. Приведите понятие модуля. Характеристики модуля.
84. Какие существуют методы разработки модулей?
85. Что такое спецификации процессов?
86. Приведите пример диаграммы переходов состояний.
87. Какие бывают функциональные диаграммы?
88. Приведите пример диаграммы потоков данных.
89. Что такое диаграммы «сущность—связь»?
90. Охарактеризуйте понятие UML.
91. Опишите варианты использования системы.
92. Чем описывается поведение системы?
93. Приведите пример структурной схемы ПО.
94. Опишите основные элементы функциональных схем ПО.
95. Охарактеризуйте метод пошаговой детализации.
96. Как составляются структурные карты Константайна?
97. Как составляются структурные карты Джексона?
98. Что такое CASE-технологии?
99. Что такое RAD-технологии?
100. Охарактеризуйте модель проектируемого ПО при объектном подходе.
101. Что такое экстремальное программирование?
102. Какие виды ошибок существуют?
103. Что такое тест? Какими свойствами должен обладать тест?
104. Каковы критерии выбора тестов?
105. Дайте краткую характеристику каждому критерию выбора теста.
106. Опишите последовательность разработки тестов.
107. Что входит в понятие надежности ПО?
108. Какие виды отказов существуют?
109. Каковы количественные характеристики надежности программ?

110. Что представляют собой методы оценки и измерения характеристик надежности ПО?
111. Перечислите достоинства парного программирования.
112. Назовите виды программных документов.
113. Как составляется пояснительная записка?
114. Каким образом составляется руководство пользователя?
115. Как составляется руководство системного администратора?
116. Назовите известные вам языки программирования и перечислите их особенности.
117. Как происходит выбор среды программирования?
118. Сравните технологию .NET с технологией Java.
119. Перечислите основные сведения о защите программных продуктов.
120. Назовите криптографические средства защиты.
121. Дайте определение программным средствам защиты.
122. Какие правовые средства защиты ПП вам известны?
123. Что такое коллективная разработка ПО?
124. Что такое система контроля версий?
125. Расскажите об основных особенностях известных вам систем контроля версий.
126. Перечислите методы оценки стоимости ПО.
127. Опишите линейный метод.
128. Опишите метод функциональных точек.
129. Какие существуют модификации метода функциональных точек?
130. Приведите методы оценки стоимости ПО с использованием эмпирических данных.
131. Охарактеризуйте COCOMO и COCOMO II.
132. Как производится оценка эффективности ПО на этапе эксплуатации?
133. Что такое показатели TCO и ROI?

Критерии и шкала оценивания по оценочному средству собеседование (устный или письменный опрос)

Шкала оценивания (интервал баллов)	Критерий оценивания
5	собеседование (устный или письменный опрос) прошел на высоком уровне (студент в полном объеме осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
4	собеседование (устный или письменный опрос) прошел на среднем уровне (студент в целом осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
3	собеседование (устный или письменный опрос) на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)
2	собеседование (устный или письменный опрос) прошел на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Задания к контрольным работам

1. Создайте проект, в котором разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы без параметров и с параметрами (имена некоторых полей должны совпадать с идентификаторами параметров), методы и свойства. Методы и свойства должны обеспечивать непротиворечивый и удобный интерфейс класса. В программе должна выполняться проверка всех разработанных элементов класса, вывод состояния объекта.

2. Создайте проект, в котором разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы с параметрами и без параметров, методы, свойства, индексаторы. Класс должен реализовывать следующие операции над массивами:

задание произвольной размерности массива при создании объекта;
доступ к элементу по индексам с контролем выхода за пределы массива;
вывод на экран элемента массива по заданному индексу и всего массива.
При возникновении ошибок должны выбрасываться исключения.
В программе должна выполняться проверка всех разработанных элементов класса.

3. Создайте проект, в котором опишите класс должен, содержать следующие элементы: скрытые и открытые поля, конструкторы (один из них должен передавать параметром массив), перегруженные операции.

В программе должна выполняться проверка всех разработанных элементов класса.

4. Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный в таблице метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

5. Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный в таблице метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

6. Составить программу с одним родительским классом и двумя потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов (количество объектов ≥ 5).

7. Составить программу с абстрактным родительским классом и двумя объектами - потомками. Для этого модифицировать задание 1. Составить тестирующую программу с выдачей протокола на экран монитора. В ней нужно реализовать циклический вывод параметров объектов, используя полиморфный контейнер - массив объектов базового класса (количество объектов ≥ 5).

8. Интерфейсы Ix, Iy, Iz, содержат объявления методов с одной и той же сигнатурой следующим образом

```
interface Ix
{
void IxF0(параметр);
void IxF1();
}
interface Iy
{
void F0(параметр);
void F1();
}
interface Iz
{
void F0(параметр);
void F1();
}
```

Эти интерфейсы наследуются в классе TestClass, содержащий член w типа параметр.

В каждом методе задать вывод результата.

Рассмотреть случай:

неявной реализации интерфейсов

явной реализации интерфейса Iz

В программе должна выполняться:

неявная неоднозначная реализация методов интерфейсов Iy и Iz,

вызов функций с явным приведением к типу интерфейса,

вызов метода для объекта посредством интерфейсной ссылки.

9. Выполнить задания, используя для хранения экземпляров разработанных классов стандартные параметризованные коллекции. Во всех классах реализовать интерфейсы IComparable и IComparer перегрузить операции отношения для реализации сравнения объектов по указанному полю. Результат вывести на экран.

Критерии и шкала оценивания по оценочному средству «контрольная работа»

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (зачёт)

1. Определите процедурную и объектную декомпозицию предметной области задачи. Чем они различаются? Назовите достоинства и недостатки этих способов декомпозиции.

2. Назовите семь основных принципов ООП и прокомментируйте, как они использованы.

3. Что такое объект и каким образом объекты соединяются в систему для решения задачи? Чем характеризуется объект?

4. Определите понятие «класс». Чем классы отличаются от других типов данных?
5. Как связаны между собой объект предметной области, класс и программный объект?
6. Каким образом в программных объектах реализуются состояние, поведение и идентификация объектов предметной области?
7. Назовите операции, которые могут быть выполнены над программными объектами.
8. Определите основные средства разработки классов. Почему они названы основными? Охарактеризуйте каждое из перечисленных средств и поясните, в каких ситуациях их целесообразно использовать.
9. Какие дополнительные средства разработки классов появились в последние годы? Для чего они могут быть использованы?
10. Назовите основные этапы разработки программных систем с использованием ООП и расскажите о каждом из них.
 11. Описание классов
 12. Методы класса
 13. Перегрузка методов
 14. Перегрузка операторов
 15. Индексаторы
 16. Свойства
 17. Наследование
 18. Виртуальные методы
 19. Интерфейсы
 20. Структуры и перечисления
 21. Обработка исключительных ситуаций
 22. Применение средств ввода-вывода
 23. Делегаты
 24. Лямба-выражения
 25. События
 26. Пространства имён, препроцессор, сборки

Критерии и шкала оценивания к промежуточной аттестации «зачёт»

Национальная шкала	Характеристика знания предмета и ответов
Зачтено	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
Незачтено	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

9. Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости рабочая программа учебной дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе с применением электронного обучения и дистанционных образовательных технологий.

Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК). В случае необходимости обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося), а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников, например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения – аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной (модулем), за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение электронного обучения, дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение электронного обучения, дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:
- продолжительность сдачи зачёта или экзамена, проводимого в письменной форме, не более чем на 90 минут;
- продолжительность подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, – не более чем на 20 минут; – продолжительность выступления обучающегося при защите курсовой работы – не более чем на 15 минут.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобренны изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)