

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛПУ»)**

Структурное подразделение Институт физико-математического
образования, информационных и обслуживающих технологий
Кафедра информационных образовательных технологий и систем

УТВЕРЖДАЮ

Врио директора ИФМОИОТ

Е.А. Журавлева

«15» Июня 2025 г.



Приложение к рабочей программе учебной дисциплины

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Технологии .Net для сложных информационных систем»**

По направлению подготовки 09.04.04 Программная инженерия

Профиль подготовки Программное обеспечение систем и комплексов

Квалификация выпускника – магистр

Форма обучения очная, заочная

Курс ОФО – 1 курс, ЗФО – 2-3 курс

Разработчик

Швыров В.В.

канд. физ.-мат. наук, доцент, доцент
кафедры информационных
образовательных технологий и систем
Заведующий кафедрой

Д.А. Капустин

Протокол от «14» Июня 2025 г. № 9

Луганск, 2025

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Технологии .Net для сложных информационных систем и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины (модуля).

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриат / специалитет / магистратура по направлению подготовки 09.04.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 932 (с изменениями и дополнениями).

1.3. Перечень компетенций, формируемых в процессе освоения основной образовательной программы

Процесс освоения дисциплины направлен на формирование следующих компетенций и индикаторов их достижения:

Код по ФГОС ВО	Индикатор достижения
Универсальные	
Общепрофессиональные	
Профессиональные	
ПК-6. Понимание существующие подходы к верификации моделей программного обеспечения	ПК-6.1. Знать методы верификации моделей программного обеспечения ПК-6.2. Уметь использовать методы верификации моделей программного обеспечения ПК-6.3. Владеть навыками использования методов верификации моделей программного обеспечения

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Введение. Основные характеристики платформы .Net	ПК-6	Выполнение лабораторных работ
Тема 2. Основные языковые конструкции	ПК-6	Выполнение лабораторных работ

Тема 3. Базовые типы языка	ПК-6	Выполнение лабораторных работ
Тема 4. Переменные и выражения	ПК-6	Выполнение лабораторных работ
Тема 5. Операторы языка	ПК-6	Выполнение лабораторных работ
Тема 6. Процедуры и функции	ПК-6	Выполнение лабораторных работ
Тема 7. Массивы языка C#	ПК-6	Выполнение лабораторных работ
Тема 8. Строки в C#. Работа с регулярными выражениями	ПК-6	Выполнение лабораторных работ
Тема 9. Классы. Структуры и перечисления	ПК-6	Выполнение лабораторных работ
Тема 10. Интерфейсы и делегаты	ПК-6	Выполнение лабораторных работ
Текущая аттестация	ПК-6	Контрольная работа
Промежуточная аттестация	ПК-6	Экзамен (письменный)

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ПК-6. Понимание существующие подходы к верификации моделей программного обеспечения	<p>ПК-6.1. Знает методы верификации моделей программного обеспечения</p> <p>ПК-6.2. Умеет использовать методы верификации моделей программного обеспечения</p> <p>ПК-6.3. Владеет навыками использования методов верификации моделей программного обеспечения</p>

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
2 семестр / 6-7 триместр			
	ОФО	О-ЗФО	ЗФО
Оформление отчетов по лабораторным работам	30 баллов		
Работа на лабораторных занятиях	30 баллов		
Выполнение тестовых заданий	-		
Выполнение заданий самостоятельной работы	10 баллов		
экзамена	30 баллов		
Итого за семестр:	100 баллов		
Всего	100 баллов		

Накопительная система оценивания по 100-балльной шкале

Четырехбал- льная система оценивания экзамена	100- балльная шкала	Буквенная шкала, соответствующая 100- балльной шкале	Система оценивания зачета
Отлично	90–100	A – отлично – 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	Зачтено
Хорошо	83–89	B – очень хорошо – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом в основном сформированы; все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	
Хорошо	75–82	C – хорошо – теоретическое содержание курса освоено полностью; 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	
Удовлетво- рительно	63–74	D – удовлетворительно – теоретическое содержание дисциплины 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, содержат ошибки	
Удовлетво- рительно	50–62	E – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетво- рительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество	Не зачтено

		их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	
Неудовлетворительно	0–20	F – неудовлетворительно – теоретическое содержание курса не освоено; необходимые практические навыки работы не сформированы; все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

2.1. Оценочные средства текущего контроля (типовые)

Вопросы для текущего контроля (темы 1-10):

1. Классификация типов данных. Встроенные типы данных. Преобразование типов.
2. Основные операторы. Чтение данных с клавиатуры. Вывод данных на экран.
3. Понятие исключительной ситуации.
4. Массивы. Размерность массива. Методы и свойства класса Array.
5. Многомерные массивы. Рваные массивы.
6. Регулярные и буквальные строковые литералы. Ссылки типа string. Операции над строками.
7. Методы и свойства класса String. Форматирование строк. Неизменяемость объектов класса String. Тип System.Text.StringBuilder.
8. Функции. Параметры функции. Функции с переменным числом параметров.
9. Функции. Функции с необязательными параметрами. Функции с именованными параметрами. Перегрузка методов.
10. Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
11. Потоки байтов FileStream.
12. Потоки символов StreamWriter и StreamReader.
13. Двоичные файлы BinaryWriter, BinaryReader.
14. Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора.
15. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы.
16. Перегрузка операций (унарные и бинарные операции, приведение типов).
17. Основные свойства ООП. Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.
18. Основные свойства ООП. Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания.
19. Абстрактные классы. Класс object. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET
20. Понятие делегата. Описание и использование. Многоадресная передача.
21. События. События для многоадресной передачи.
22. Механизм обработки исключений. Системные исключения и их обработка. Свойства и методы класса Exception. Исключения, создаваемые программистом.
23. Основные абстрактные структуры данных. Физическое представление данных.
24. Пространство имен System.Collections. Интерфейсы пространства имен System.Collections.

25. Коллекции общего назначения. Основные элементы класса ArrayList.
26. Коллекции общего назначения. Основные элементы класса Hashtable.
27. Коллекции общего назначения. Основные элементы класса SortedList.
28. Коллекции общего назначения. Основные элементы класса Stack.
29. Коллекции общего назначения. Основные элементы класса Queue.
30. Коллекции общего назначения. Доступ к коллекциям с помощью нумератора.
31. Классы-прототипы. Основные преимущества использования обобщений. Ограничения.
32. Параметризованные коллекции библиотеки .NET.
33. Создание собственных классов-прототипов. Обобщенные методы.
34. Что такое интерфейс в языке программирования C#?
35. Как объявить интерфейс в C#?
36. Может ли интерфейс содержать реализацию методов?
37. Какие модификаторы доступа могут использоваться для членов интерфейса?
38. В чем разница между интерфейсом и абстрактным классом в C#?
39. Как класс реализует интерфейс в C#?
40. Может ли один класс реализовывать несколько интерфейсов?
41. Что такое явная реализация интерфейса, и как она используется?
42. Может ли интерфейс наследовать другой интерфейс в C#?
43. Как использовать интерфейсы для достижения множественного наследования в C#?
44. Что такое делегат в C#?
45. Как объявить делегат в C#?
46. В чем отличие между делегатами и интерфейсами?
47. Как присвоить метод делегату?
48. Какие предварительно определенные (built-in) делегаты существуют в C#?
49. Что такое многозадачность с использованием делегатов (multicasting)?
50. Можно ли изменить состояние переменных внутри делегата?
51. Как использовать асинхронные делегаты (async delegates) в C#?
52. В чем отличие между делегатами и событиями (events) в C#?

53. Как создать собственный делегат с параметрами в C#?
54. Какие преимущества предоставляют интерфейсы и делегаты в структуре программы?
55. Каким образом можно использовать интерфейсы и делегаты для обеспечения расширяемости кода?
56. Какие практические примеры использования интерфейсов и делегатов вы можете привести?
57. Как обрабатываются исключения при использовании интерфейсов и делегатов?
58. Каким образом производится тестирование кода, использующего интерфейсы и делегаты в C#?
59. Может ли интерфейс содержать делегат в C#?
60. Как использовать делегаты в интерфейсах?
61. Как реализовать множественное наследование с использованием интерфейсов и делегатов в C#?
62. Как обеспечить вызов метода через делегат, предоставленный интерфейсом?
63. Как использовать делегаты для обратного вызова методов в интерфейсах?
64. Что такое перечисление (enum) в C#?
65. Как объявить перечисление в C#?
66. Каким образом можно связать числовые значения с элементами перечисления?
67. Как использовать перечисления для улучшения читаемости кода?
68. Может ли перечисление содержать методы или свойства?
69. Какие дополнительные атрибуты могут использоваться с перечислениями в C#?
70. Как обратиться к значениям перечисления по их именам?

2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. В чем разница между интерфейсно-ориентированным, объектно-ориентированным и аспектно-ориентированным программированием?
2. Что такое CLR? Что такое IL? Что такое CLS?
3. Что такое managed code?
4. Что такое assembly?
5. Что такое приватные и совместные сборки?
6. Что такое assembly manifest?
7. В чем разница между понятиями namespace и assembly?
8. В чем различие между Value Type и Reference Type?
9. Когда объект удаляется сборщиком мусора?
10. Что такое Code Access Security (CAS)?
11. Что такое attribute?
12. Как обеспечить использование именованных параметров в конструкторе атрибута?
13. В чем различие между Finalize и Dispose?
14. Какие типы можно использовать в предложении foreach?
15. В чем различие между классом и структурой?
16. Что означает модификатор virtual?
17. Чем отличается event от delegate?
18. Может ли класс реализовать два интерфейса, у которых объявлены одинаковые методы? Каким образом?
19. Поддерживает ли C# множественное наследование?
20. Кому доступны переменные с модификатором protected на уровне класса?
21. Наследуются ли переменные с модификатором private?
22. Опишите модификатор “protected internal”
23. Назовите класс .NET, от которого наследуются все классы?
24. Что обозначает термин immutable (неизменяемый)?
25. Какая разница между классами System.String и System.Text.StringBuilder?
26. Какое преимущество использования класса System.Text.StringBuilder перед System.String?
27. Можно ли хранить разные типы данных в объекте класса System.Array?
28. Объясните разницу между System.Array.CopyTo() и System.Array.Clone()?
29. Как отсортировать элементы массива в убывающем порядке?
30. As, is – что это, как применяется?
31. В чем разница между throw ex; и throw;?
32. Как работает return в try-catch-finally?
33. Какой синтаксис используется для указания класса родителя в C#?
34. Можно ли запретить наследование от своего собственного класса?
35. Можно ли разрешить наследование класса, но запретить перекрытие метода?
36. Что такое абстрактный класс?

37. В каком случае вы обязаны объявить класс абстрактным?
38. Что такое интерфейс класса?
39. Почему нельзя указать модификатор видимости для методов интерфейса?
40. Можно ли наследовать от нескольких интерфейсов?
41. Назовите отличия между интерфейсом и абстрактным классом?
42. Назовите различия между структурами и классами.
43. В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами?
44. Dispose(), Finalize() – что это за методы, как используются в .NET?
45. Для чего в .NET используется конструкция using(...) {...}? При чем тут IDisposable?
46. Назовите явное имя параметра, передаваемого в метод set свойства класса?
47. Что обозначает ключевое слово “virtual” для метода или свойства?
48. Чем перекрытый метод отличается от перегруженного метода?
49. Можно ли объявить перекрытый метод статическим, если перекрываемый метод не является статическим?
50. Что такое «сопутствующая сборка» (satellite assembly)?
51. Какая наименьшая исполнимая единица в .NET?
52. Что происходит в памяти при упаковке и распаковке значимого типа?
53. Что такое класс в C#?
54. Как объявить класс в C#?
55. Какие модификаторы доступа могут использоваться для членов класса?
56. Как объявить и использовать конструктор в классе C#?
57. Что такое статический член класса и как он отличается от нестатического?
58. В чем различие между свойствами (properties) и полями (fields) класса?
59. Как осуществляется наследование в C#?
60. Как запретить наследование от класса в C#?
61. Что такое абстрактный класс, и как он используется в C#?
62. Как реализовать интерфейс в C# с использованием класса?
63. Что такое перечисление (enum) в C#?
64. Как объявить перечисление в C#?
65. Каким образом можно связать числовые значения с элементами перечисления?
66. Как использовать перечисления для улучшения читаемости кода?
67. Может ли перечисление содержать методы или свойства?

68. Какие дополнительные атрибуты могут использоваться с перечислениями в C#?
69. Как обратиться к значениям перечисления по их именам?
70. Как перечисление связано с целочисленными типами данных в C#?