

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ГОУ ВО ЛНР «ЛГПУ»)

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

Кафедра информационных образовательных технологий и систем



УТВЕРЖДАЮ

Директор ИФМОИОТ

Е.Е. Горбенко Горбенко Е.Е.

2022 г.

Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Программирование для платформы .Net»

По направлению подготовки 44.03.04 Профессиональное обучение (по
отраслям)

Профиль подготовки – Разработка программного обеспечения
образовательных систем

Квалификация выпускника – бакалавр

Форма обучения – очная, заочная

Курс – ОФО – 3 курс (5 семестр), ЗФО – 3 курс (9 триместр)

Разработчик

Швыров В.В.

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

И.о. заведующего кафедрой

Д.А. Капустин Д.А. Капустин

«26» апреля 2022 г.

Луганск, 2022

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

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

Процесс освоения дисциплины направлен на овладение следующими компетенциями:

УК-1 - Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.

ПК-6 - Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения.

ПК-7 - Способность создавать программные интерфейсы.

ПК-9 - Владение навыками использования различных технологий разработки программного обеспечения.

1.2. 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 | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 2. Основные языковые конструкции | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 3. Базовые типы языка | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 4. Переменные и выражения | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 5. Операторы языка | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 6. Процедуры и функции | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 7. Массивы языка C# | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 8. Строки в C#. Работа с регулярными выражениями | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 9. Классы. Структуры и перечисления | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Тема 10. Интерфейсы и делегаты | УК-1; ПК-6; ПК-7; ПК-9 | Выполнение лабораторных работ |
| Форма аттестации | УК-1; ПК-6; ПК-7; ПК-9 | Экзамен (письменный) |

1.3. Описание показателей формирования компетенций

| Код и наименование универсальной компетенции | Код и наименование индикатора достижения универсальной компетенции |
|--|--|
| УК-1 - Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач. | Знать: основные принципы критического анализа, методы критического анализа и оценки современных научных достижений. Уметь: получать новые знания на основе анализа, синтеза и других методов исследования, систематизировать данные по научным проблемам, относящимся к профессиональной области, осуществлять поиск информации и решений на основе теоретического изучения проблемы или экспериментальных действий. Владеть: методами и приемами интеллектуальной деятельности (анализа, синтеза и др.) для исследования профессиональных вопросов. |
| ПК-6 - Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения | Знать основы методики моделирования, анализа и использования формальных методов конструирования программного обеспечения Уметь использовать на практике методы моделирования, анализа и конструирования программного обеспечения Владеть методами формализации, моделирования, анализа и конструирования программного обеспечения |
| ПК-7 - Способность создавать программные интерфейсы | Знать способы, методику и технологию создания программных интерфейсов Уметь создавать интуитивно понятные программные интерфейсы Владеть навыками создания современных программных интерфейсов |
| ПК-9 - Владение навыками использования различных технологий разработки программного обеспечения. | Знать современные методики и технологии разработки программного обеспечения (структурное, объектно-ориентированное). Уметь использовать на практике методики и технологии разработки программного обеспечения Владеть навыками использования различных технологий разработки программного обеспечения |

1.4. Критерии оценивания компетенций на разных этапах их формирования

| Вид текущей учебной работы | Количество баллов |
|--|-------------------|
| 5 семестр / 9 триместр | |
| 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

| Четырехбалльная система оценивания экзамена | 100-балльная шкала | Буквенная шкала, соответствующая 100-балльной шкале | Система оценивания зачета |
|---|--------------------|--|---------------------------|
| Отлично | 90–100 | А – 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 | В – 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 | С – 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 | |
| Удовлетворительно | 63–74 | Д – 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 | |
| Удовлетворительно | 50–62 | Е – 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 | |
| Неудовлетворительно | 21–49 | FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения | Не зачтено |

| | | | |
|---------------------|-------------|--|--|
| | | учебных заданий не выполнено либо качество 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 | |

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ЛНР
ГОУ ВО ЛНР «ЛГПУ»
ИНСТИТУТ ФИЗИКО-МАТЕМАТИЧЕСКОГО ОБРАЗОВАНИЯ, ИНФОРМАЦИОННЫХ
И ОБСЛУЖИВАЮЩИХ ТЕХНОЛОГИЙ

2022 – 2023 учебный год

Направление подготовки (специальность): 44.03.04 Профессиональное обучение (по отраслям)

курс / форма обучения (ОФО, ЗФО): ОФО, ЗФО

Семестр / триместр 5 семестр / 9 триместр

Учебная дисциплина: Программирование для платформы .Net

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Цель и задачи изучения дисциплины «Программирование для платформы .Net». Требования к уровню освоения содержания дисциплины.
2. Типы данных
3. Практическое задание.

Утверждено на заседании кафедры информационных образовательных технологий и систем

Протокол № ____ от ____ г.

И.о. заведующего кафедрой ИОТС _____

Капустин Д.А.

(подпись)

Экзаменатор _____

(подпись)

доцент, Швыров В.В.

(должность, ФИО преподавателя)

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.

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?

Перечень типовых практических заданий к экзамену по дисциплине «Основы разработки C#»:

| № п/п | Перечень типовых практических заданий к экзамену |
|-------|--|
| 1 | <i>Пользователь вводит число в консоль, выводится сообщение, является ли это число чётным.</i> |
| 2 | Пользователь вводит число и вводит степень Р, в которую необходимо возвести число, в консоль выводится результат выполнения операции возведения в степень. |

| | |
|----|---|
| 3 | Пользователь вводит два числа -> в консоль выводятся все символы соответствующие кодам, начиная с первого числа и заканчивая вторым, в формате символ = код. |
| 4 | Пользователь вводит целое, положительное число, в консоль выводится введённое число в двоичной форме. |
| 5 | На вход программы подаётся размерность двумерного массива MxN (значения M и N заданы в коде) -> двумерный массив заполняется числами от 1 до M*N по спирали -> заполненный массив выводится в консоль. |
| 6 | На вход программы подаётся двумерный массив размерностью MxN (значения M и N заданы в коде) -> осуществляется поиск максимального и минимального элементов в массиве -> массив, а так же значения максимального и минимального элементов выводятся в консоль. |
| 7 | На вход программы подаётся строка символов (заполняется в коде, или из консоли) -> определяется количество повторений для каждого символа -> в консоль выводится заданная, и уникальные символы этой строки с количеством их повторений в ней. |
| 8 | На вход программы подаётся строка символов (заполняется в коде, или из консоли) -> пользователь вводит символ -> введённый пользователь символ удаляется из строки -> получившаяся строка выводится в консоль. |
| 9 | Реализовать рекурсивную функцию drawPyramid , выводящую в консоль пирамиду из символов ^, с основанием равным заданному числу |
| 10 | Реализовать заполнение массива случайно сгенерированными числами в заданном диапазоне. |