

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.Ф. УТКИНА**

Кафедра «Автоматики и информационных технологий в управлении»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДИСЦИПЛИНЫ

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Специальность 12.05.01

«Электронные и оптико-электронные приборы
и системы специального назначения»

ОПОП

«Оптико-электронные информационно-измерительные приборы и системы»

Квалификация выпускника – инженер

Формы обучения – очная

Рязань 2024 г.

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций обучающихся целям и требованиям основной профессиональной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций, приобретаемых обучающимися в соответствии с этими требованиями.

Контроль знаний проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретенных обучающимися в ходе выполнения индивидуальных заданий на практических занятиях и лабораторных работах. При оценивании результатов освоения практических занятий и лабораторных работ применяется шкала оценки «зачтено – не зачтено». Количество лабораторных и практических работ и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

Промежуточный контроль по дисциплине осуществляется проведением зачёта (7 семестр) и экзамена (8 семестр). При оценивании результатов освоения дисциплины на зачете применяется шкала оценки «зачтено – не зачтено», а на экзамене – традиционная балльная система (отлично, хорошо, удовлетворительно, неудовлетворительно).

Форма проведения зачета и экзамена – устный ответ по утвержденным экзаменационным билетам, сформулированным с учетом содержания учебной дисциплины. После выполнения письменной работы обучаемого производится ее оценка преподавателем и, при необходимости, проводится теоретическая беседа с обучаемым для уточнения экзаменационной оценки.

Паспорт оценочных материалов по дисциплине

Модуль 1

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	2	3	4
1	<i>Раздел 1</i> Программирование на языке C#	ОПК-3.2-З ОПК-3.2-У ОПК-3.2-В	Зачет, лабораторная работа

Модуль 2

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	2	3	4
1	<i>Раздел 3</i> Динамическая типизация. Атрибуты. Потоки.	ОПК-3.2-З ОПК-3.2-У ОПК-3.2-В	Экзамен
2	<i>Раздел 4</i> Современные подходы к разработке ПО. Системы контроля версий.	ОПК-3.2-З ОПК-3.2-У ОПК-3.2-В	Экзамен
3	<i>Раздел 5</i> Паттерны проектирования ПО	ОПК-3.2-З ОПК-3.2-У ОПК-3.2-В	Экзамен
4	<i>Раздел 6</i> Создание оконных приложений Windows Forms	ОПК-3.2-З ОПК-3.2-У ОПК-3.2-В	Экзамен

Критерии оценивания компетенций (результатов)

- 1). Уровень усвоения материала, предусмотренного программой.
 - 2). Умение анализировать материал, устанавливать причинно-следственные связи.
 - 3). Ответы на вопросы: полнота, аргументированность, убежденность, умение
 - 4). Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)
 - 5). Использование дополнительной литературы при подготовке ответов.
- Уровень освоения сформированности знаний, умений и навыков по дисциплине оценивается в форме бальной оценки:

«**Отлично**» заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение

свободно выполнять задания, предусмотренные программой, изучивший основную, и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка «отлично» выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

«Хорошо» заслуживает студент, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

«Удовлетворительно» заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

«Неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.

Оценка «зачтено» выставляется студенту, который прочно усвоил предусмотренный программный материал; правильно, аргументировано ответил на все вопросы, с приведением примеров; показал глубокие систематизированные знания, владеет приемами рассуждения и сопоставляет материал из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов.

Обязательным условием выставленной оценки является правильная речь в быстром или умеренном темпе. Дополнительным условием получения оценки «зачтено» могут стать хорошие успехи при выполнении самостоятельной работы, а также уверенная защита лабораторных работ.

Оценка «не зачтено» выставляется студенту, который не справился с 50% вопросов и заданий билета, в ответах на другие вопросы допустил существенные ошибки. Не может ответить на дополнительные вопросы, предложенные преподавателем. Целостного представления об изучаемой дисциплине у студента нет. Оценивается качество устной речи и изложение письменного материала, как и при выставлении положительной оценки.

Типовые контрольные задания или иные материалы

Вопросы к зачету по дисциплине

1. Обзор платформы Microsoft .NET. Общеязыковая исполняющая среда Common Language Runtime (CLR). Библиотека базовых классов Framework Class Library (FCL). Стандартная система типов Common Type System (CTS). Общеязыковая спецификация Common Language Specification (CLS). Языки программирования .NET. Компиляция и исполнение .NET приложений.
2. Обзор Visual Studio. Основные окна Visual Studio, используемые при программировании. Интегрированная система документации. Типы приложений. Общая структура программы. Оператор using. Точка входа программы Main(). Пример простой программы. Документирование кода.
3. Иерархия классов для типов данных. Понятие структурных и ссылочных типов. Стек. Куча. Встроенные типы данных. Псевдонимы. Оператор sizeof.
4. Переменная. Правила именования. Область видимости и время существования переменных. Константы. Класс Console. Форматирование строк для вывода на консоль. Класс Math.
5. Преобразование типов (Явное и неявное преобразование, повышающее и понижающее преобразование). Checked и Unchecked. Класс Convert, методы ToString и Parse. Неявно типизированные локальные переменные.
6. Операции. Приоритеты операций. Арифметические операции. Операции отношения. Логические операции. Поразрядные операции. Операции присваивания.
7. Операторы. Условные конструкции (условный оператор if, оператор многозначного выбора switch, тернарный оператор). Операторы перехода (goto, break, continue). Циклические конструкции (while, do-while, for, foreach))
8. Понятие процедурного программирования. Методы. Модификаторы параметров аргументов методов (ref, out, params). Перегрузка методов. Параметры аргументов при вызове методов (позиционные, именованные, опциональные). Рекурсивные методы.
9. Понятие объектно-ориентированного программирования (ООП). Понятие класса. Поля класса. Методы класса. Объекты и экземпляры. Создание экземпляра класса по сильной и слабой ссылке. Передача экземпляра класса в качестве аргумента метода.
10. Парадигмы ООП. Инкапсуляция. Спецификаторы доступа. Свойства класса. Автоматически реализуемые свойства.
11. Инициализация экземпляров класса. Понятие конструктора класса. Использование ключевого слова this для обращения к полям класса. Вызов конструкторов по цепочке. Частичные классы. Частичные методы. Поля только для чтения.
12. Понятие массива. Типы массивов. Одномерный массив. Многомерные массивы. Прямоугольные и зубчатые массивы. Класс Array.
13. Понятие индексатора. Примеры с целочисленными и строковыми типами принимаемых аргументов индексатора.

14. Понятие наследования. Создание многоуровневой иерархии классов. Модификатор доступа `protected`. Вызов конструктора базового класса из производного класса.
15. Понятие полиморфизма. Классический полиморфизм. Приведение к базовому и производному типу. Замещение метода базового класса. Виртуальные методы. Операторы `as` и `is`. Модификатор `sealed`.
16. Понятие абстракции. Понятие абстрактного класса. Абстрактные члены.
17. Понятие интерфейса. Реализация интерфейсов. Множественное наследование интерфейсов. Техника объединения реализации одноименных абстрактных методов. Наследование от класса и интерфейса, у которых совпадают сигнатуры членов.
18. Класс `Char`. Массив символов `char[]`. Класс `String`. Способы создания строк. Строка – неизменяемый тип. Понятие интернирования строк. Класс `StringBuilder`.
19. Регулярные выражения. Класс `Regex`. Метасимволы. Квантификаторы. Примеры шаблонов.
20. Статические классы и их члены. Понятие методов расширения.
21. Понятие структуры. Работа со структурными типами. Отличие структуры от класса. Вложенные структуры и классы. Упаковка (`boxing`) и распаковка (`unboxing`). Структура `DateTime`.
22. Понятие Перечисления. Типы перечислений. Примеры использования перечислений.
23. Понятие делегатов. Создание делегата. Создание экземпляра делегата. Вызов методов, сообщенных с делегатом. Неизменяемость делегатов. Комбинирование делегатов. Анонимные методы. Лямбда-операторы. Лямбда-выражения. Техника предположения делегата.
24. Понятие события. Создание события и его свойства. Обработчики событий. Подписка и отписка от событий. Паттерн проектирования `Model-View-Presenter`.
25. Понятие обобщений. Идентификатор `<T>`. Обобщение и упаковка-распаковка. Ограничения обобщений. Ограничение `new()`. Ограничение `struct`. Ограничение `class`. Ограничение `<base class name>`. Неприкрытое ограничение типа.
26. Понятие коллекции. Интерфейс `IEnumerable`. Интерфейс `IEnumerator`. Ключевое слово `yield`. Циклический оператор `foreach`. Интерфейс `IEnumerable<T>`. Другие интерфейсы для работы с обобщенными и необобщенными коллекциями.
27. Коллекции `ArrayList`, `List<T>`, `Dictionary<TKey, TValue>` и другие. Свойства и методы коллекций.
28. Роль класса `Object`. Методы класса `Object`. Интерфейс `ICloneable`.
29. Перегрузка операторов. Правила перегрузок. Операторы явного и неявного преобразования типа `explicit` и `implicit`.
30. Понятие обработки исключений. Ключевые слова `try`, `catch`, `finally`, `throw`. Обработка нескольких конструкций `catch`. Примеры использования.

Вопросы к экзамену по дисциплине

1. Анонимные типы. Динамическая типизация. Тип `dynamic`.
2. Понятие LINQ. Запросы LINQ. Ключевые слова: `from`, `where`, `select`, `group`, `orderby`, `join`, `let`.
3. Пространства имен. Директива `using`. Подключение новых сборок к проекту. Модификаторы доступа при работе со сборками. Директивы компилятора: `#if`, `#endif`, `#region`, `#else`, `#elif`, `#define`, `#undef`.
4. Понятие атрибута. Предопределенные и пользовательские атрибуты. Класс `Attribute`. `AttributeUsage`, `Conditional`, `Obsolete`.
5. Понятие многозадачности на основе процессов и на основе потоков. Пространство имен `System.Threading`. Классы `Thread`, `Monitor`, делегаты `ThreadStart`, `ParameterizedThreadStart`.
6. Понятие приоритетного и фонового потока. Понятие синхронизации потоков. Синхронный и асинхронный потоки. Ключевое слово `lock`.
7. Понятие технологии разработки ПО. Основные этапы разработки ПО.
8. Модели жизненного цикла ПО. Водопадные модели. V-образная модель. Итеративная модель. Спиральная модель.
9. Современные подходы к разработке ПО.
10. Понятие системы контроля версий. Система `Git`. Работа с удаленным репозиторием. Запись изменений, чтение информации, операции отмены.
11. Понятие системы контроля версий. Система `Git`. Управление ветками: слияние, перебазирование, удаление. Решение конфликтов.
12. Понятие паттерна проектирования. Порождающие паттерны. Паттерн `Abstract Factory`.
13. Понятие паттерна проектирования. Порождающие паттерны. Паттерн `Abstract Builder`.
14. Понятие паттерна проектирования. Порождающие паттерны. Паттерн `Factory Method`.
15. Понятие паттерна проектирования. Порождающие паттерны. Паттерн `Prototype`.
16. Понятие паттерна проектирования. Порождающие паттерны. Паттерн `Abstract Singleton`.
17. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Adapter`.
18. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Bridge`.
19. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Composite`.
20. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Decorator`.
21. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Facade`.
22. Понятие паттерна проектирования. Структурные паттерны. Паттерн `Flyweight`.
23. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн `Chain of Responsibility`.
24. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн `Command`.

25. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Interpreter.
26. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Iterator.
27. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Mediator.
28. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Memento. Паттерн State.
29. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Observer.
30. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Strategy.
31. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Template Method.
32. Понятие паттерна проектирования. Поведенческие паттерны. Паттерн Visitor.
33. Обзор пространства имен System.Windows.Forms. Проект Windows Forms. Класс Application. Класс Control. Класс Form.
34. Проект Windows Forms. Работа с меню (MenuStrip), со строкой состояния (StatusStrip) и таймером (Timer). Создание дочерних окон.
35. Обзор пространства имен GDI+. Обзор пространства имен System.Drawing. Тип Point(F). Тип Rectangle(F). Класс Region.
36. Рисование в окне. Пространство имен System.Windows.Drawing. Класс Graphics. Основные классы для рисования – Pen, Brush.
37. Рисование в окне. Пространство имен System.Windows.Drawing. Класс Graphics. Основные классы для рисования – Color, Font.
38. Элементы управления Windows Forms. Базовые элементы управления. Элементы Label, TextBox, MaskedTextBox, Button.
39. Элементы управления Windows Forms. Базовые элементы управления. Элементы CheckBox, RadioButton, GroupBox, CheckedListBox, ListBox, ComboBox.
40. Элементы управления Windows Forms. Создание изображений. Контроль анимации.
41. Классификация видов тестирования ПО.
42. Понятие Unit-тестирования.

Типовые задания для самостоятельной работы

Модуль 1 (семестр 7)

1. История платформы .NET.
2. Встроенные типы данных на языке C#.
3. Виды условных операторов в C#.
4. Обеспечение инкапсуляции при написании программ на языке C#.
5. Проявление полиморфизма в C#.
6. Сравнение интерфейсов и абстрактных классов в C#.
7. Статические члены в C#.
8. Классы для работы со строками в C#.
9. Сравнение структур и классов в C#.
10. Классификация коллекций в C#.

Модуль 2 (семестр 8)

1. Понятие динамической типизации.
2. Работа с потоками в C#.
3. Модели жизненного цикла ПО.
4. Обзор одного из паттернов проектирования.
5. Обзор метода управления проектами – SCRUM.
6. Создание модулей в Паскале.
7. Принцип работы с оконными приложениями Windows Forms.
8. Обзор видов тестирования ПО.
9. Обзор пространства имен System.Drawing.
10. Правила формирования запроса LINQ.

Лабораторный практикум

Модуль 1

№ п/п	№ раздела дисциплины	Наименование лабораторной работы	Трудоемкость, час
1	1	Основы языка программирования c#	4
2	1	Операции и операторы	4
3	1	Классы	4
4	1	Массивы и индексы. Наследование и полиморфизм	4

Практические занятия

Модуль 1

№ п/п	№ раздела дисциплины	Наименование практического занятия	Трудоемкость, час
1	1	Абстрактные классы. Интерфейсы.	4
2	1	Символы. Строки. Регулярные выражения.	4
3	1	Делегаты. События.	4
4	1	Коллекции	4

Модуль 2

№ п/п	№ раздела дисциплины	Наименование практического занятия	Трудоемкость, час
1	3	Анонимные и динамические типы. LINQ.	4
2	3	Многозадачность на основе потоков.	4
3	4	Системы контроля версий.	4
4	4	Основы тестирования ПО.	4
1	5	Порождающие паттерны.	4
2	5	Структурные паттерны.	4
3	5	Поведенческие паттерны.	4
4	6	Разработка оконных приложений Winsows Forms	4