

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»**

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
«Визуальное программирование»**

Направление подготовки – 02.03.03 «Математическое обеспечение и администрирование
информационных систем»

ОПОП академического бакалавриата
«Математическое обеспечение и администрирование информационных систем»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная, очно-заочная

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, закрепленных за дисциплиной.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме экзамена.

Форма проведения экзамена – тестирование, письменный опрос по теоретическим вопросам.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов

1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (экзамен) выносятся тест, два теоретических вопроса. Максимально студент может набрать 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется студенту, который набрал в сумме 9 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических и лабораторных работ.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 6 до 8 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических и лабораторных работ.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 3 до 5 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических и лабораторных работ.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 3 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические или лабораторные работы.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Основы работы с платформой .NET Framework	ОПК-3.1	Экзамен
Объектно-ориентированное программирование в языке C#	ОПК-2.2	Экзамен
Работа с .NET Framework	ОПК-3.1	Экзамен
Создание пользовательских приложений на языке C#	ОПК-2.1 ОПК-3.1	Экзамен

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме экзамена

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности

ОПК-2.1 Применяет современный математический аппарат, связанный с проектированием и оценкой качества программных продуктов в различных областях деятельности

Типовые тестовые вопросы

1. Что не относится к основным элементам приложения WindowsForms?
Форма
Интерфейс
Элемент управления
2. От какого класса должен наследоваться пользовательский элемент управления?
ApplicationException
TextBox
Button
Control
3. Что является способом редактирования элементов управления и форм?
Свойства
События
Конструкторы
Поля
4. Что используется для разработки мобильных приложений?
WPF
Xamarin
ASP.NET
5. Для каких платформ нельзя создавать приложения с использованием Xamarin.Forms?
iOS
Android
Windows
Linux
6. Что позволяет делать Android Design?
Компилировать мобильные приложения
Создавать интерфейс без редактирования XAML
Запускать Android-эмулятор
7. Что не относится к технологиям разработки веб-приложений?
ASP.NET MVC
Razor Pages
Blazor
Blazer Pages
8. Какое расширение имеет страница ASP.NET приложения?
XML
XAML
CSHTML
HTTP

9. Что означает аббревиатура MVC?

Mapping-View-Control
Model-Verify-Controller
Model-View-Controller
Match-Virtual-Control

10. Какой метод вызывается в URL `http://localhost:8080/User/Add` ?

User.Add()
Controller.User.Add()
UserController.Add()
User.Index()

Типовые вопросы открытого типа:

1. Приложение WindowsForms представляет собой ... (**событийно ориентированное приложение, поддерживаемое Microsoft .NET Framework**)
2. Визуальная поверхность, на которой выводится информация для пользователя – это ... (**форма**)
3. Отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных – это ... (**элемент управления**).
4. Платформа с открытым исходным кодом, предназначенная для построения современных производительных приложений для iOS, Android и Windows – это ... (**Xamarin**)
5. Целевая версия Android указывает ... (**версию Android, в которой предназначено запускать приложение**).
6. В инфраструктуре MVC контроллеры – это классы C#, обычно производные от класса ... (**System.Web.Mvc.Controller**)
7. Каждый запрос, поступающий в MVC приложение, обрабатывается ... (**контроллером**)
8. По умолчанию в контроллере веб-приложения вызывается метод ... (**Index()**)
9. Механизм визуализации в веб-приложениях имеет название ... (**Razor**)
10. Компоненты веб-приложения, формирующие пользовательские страницы для отображения данных из модели, называются ... (**представлениями**)

ОПК-2.2 Применяет современный математический аппарат, связанный с разработкой и реализацией программных продуктов и программных комплексов в различных областях деятельности

Типовые тестовые вопросы

1. Какой класс является базовым для всех остальных классов?

System
Object
Class
Value

2. Какой член класса не относится к функциям-членам?

событие
финализатор
конструктор
индексатор

3. Какой механизм полиморфизма использует позднее связывание?

переопределение
перегрузка
перекрытие

4. Можно ли создать экземпляр абстрактного класса?

да
нет
зависит от ситуации

5. Каким ключевым словом в базовом классе помечается метод, помеченный как `override` в производном классе?

abstract
virtual
new
static

6. Допустимо ли в .NET множественное наследование от классов?
- да
 - нет**
 - зависит от ситуации
7. Какое ключевое слово при объявлении класса запрещает наследование от него?
- virtual
 - sealed**
 - static
 - abstract
8. Можно ли создать статический член класса в нестатическом классе?
- да
 - нет
9. Какой модификатор доступа ограничивает доступ текущей сборкой?
- public
 - private
 - internal**
 - protected
10. Какой метод не наследуется из класса object?
- Equals
 - GetEnumerator**
 - ToString
 - GetHashCode

Типовые вопросы открытого типа:

1. Член класса, который вызывается при уничтожении экземпляра класса, называется ... (**финализатором**).
2. Класс, от которого запрещено наследование, называется ... (**запечатанным**).
3. Механизм ООП, при котором в одном классе объявляется несколько методов с одним именем и разной сигнатурой – это ... (**перегрузка**).
4. При перекрытии метода используется механизм ... (**раннего**) связывания.
5. В C# значимые типы напрямую наследуются от класса ... (**ValueType**).
6. Для доступа к члену базового класса из производного класса используется ключевое слово ... (**base**).
7. Основной единицей инкапсуляции в C# является ... (**класс**).
8. Способность языка позволять строить новые определения классов на основе определений существующих классов – это ... (**наследование**).
9. Для описания сущностей, не имеющих конкретного воплощения, используется ... (**абстрактный**) класс.
10. При раннем связывании формирование кода происходит на этапе ... (**компиляции**).

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-3	Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

ОПК-3.1 Применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

Типовые тестовые вопросы

1. Каким ключевым словом обозначается пространство имен?
static
namespace
using
readonly
2. Какой интерфейс должен реализовать коллекция, используемая в цикле foreach?
IEnumerable
IEnumerator
IList
ICollection
3. Как расшифровывается аббревиатура JIT?
Just-in-Time
Journal-in-Transaction
Just-into-Timespan
Java-internal-Translation
4. Какой класс используется для снижения затрат памяти при работе со строками?
String
Regex
StringBuilder
StreamReader
5. Какой механизм .NET позволяет избежать проблем упаковки-распаковки и дублирования кода?
делегаты
обобщения
события
сборка мусора
6. В каком пространстве имён содержится класс Regex?
System.Collections.Generic
System.IO
System.Text.RegularExpressions
System.Text
7. Как можно инициализировать делегат?
имя метода
лямбда-функция
анонимный метод
всё вышеперечисленное
8. В каком свойстве исключения хранится информация о стеке вызовов?
Data
StackTrace
Message
InnerException

9. Какой интерфейс используется при работе с неуправляемыми ресурсами?

ICollection

IFile

IDisposable

IComparable

10. С какой областью памяти работает сборщик мусора?

стек

управляемая куча

стек и управляемая куча

ничего из вышеперечисленного

Типовые вопросы открытого типа:

1. Для хранения ссылки на метод используется ... (**делегат**).

2. При работе с событиями используется механизм ... (**публикации-подписки**).

3. Объекты значимых типов хранятся в ... (**стеке**).

4. Для хранения очень больших объектов используется специальная куча ... (**Large object heap**).

5. Во время работы программы при нехватке памяти для создания новых объектов начинает работать... (**сборщик мусора**).

6. Стандартный делегат, не имеющий возвращаемого значения – это ... (**Action**).

7. Для работы с исключениями используется стандартный класс ... (**Exception**).

8. При создании обобщенного класса код where T : struct означает, что T должен быть ... (**структурой**).

9. Необобщенные коллекции содержат объекты типа ... (**object**).

10. Класс StreamReader используется для ... (**чтения из файла**).

Типовые теоретические вопросы для экзамена по дисциплине

1. NET. Основные понятия. Диспетчер пакетов NuGet.
2. .NET. Common Language Runtime.
3. .NET. Just-in-Time компиляция.
4. Структура Solution. Структура проекта.
5. Язык C#. Типы данных. Пространство имен.
6. Параметры методов. Ключевые слова in, out и params.
7. Принципы ООП. Класс. Понятие, описание. Использование ключевого слова partial.
8. Класс. Данные-члены. Примеры.
9. Класс. Функции-члены. Примеры.
10. Статический класс. Статические члены класса.
11. Инкапсуляция. Понятие, реализация в C#.
12. Наследование. Основные понятия. Ключевое слово base. Запечатанный класс.
13. Абстрактные классы. Абстрактные и виртуальные методы.
14. Интерфейс. Основные понятия. Ковариантность и контравариантность интерфейсов.
15. Полиморфизм. Основные понятия. Раннее и позднее связывание.
16. Полиморфизм. Переопределение, перекрытие и перегрузка методов.
17. Строки. Тип string и класс String.
18. Класс StringBuilder. Основные методы. В каких случаях используется.
19. Регулярные выражения. Для чего используются. Примеры регулярных выражений. Класс Regex.
20. Типобезопасность. Упаковка и распаковка. Примеры.
21. Обобщения. Основные понятия, достоинства. Ограничения дженериков.
22. Обобщения. Наследования дженериков. Ковариантность и контравариантность.
23. Коллекции. Основные понятия. Типы коллекций. Примеры.
24. Перечислитель. Интерфейсы IEnumerable и IEnumerator и их обобщенные версии. Итератор.
25. Делегаты. Определение. Свойства делегатов. Синтаксис объявления, инициализации и вызова.
26. Делегаты. Ковариантность и контравариантность делегатов. Примеры.
27. Анонимные функции. Использование лямбда-выражений и анонимных методов при инициализации делегатов.
28. Мультикаст-делегаты. Основные понятия. Пример.
29. Стандартные делегаты. Примеры и особенности использования.
30. Исключения. Обработка исключений. Класс Exception.
31. Исключения. Создание специальных исключений. Пример.
32. События. Основные понятия. Механизм публикации/подписки. Обработчики событий.
33. События. Порядок работы с событиями. Статические, виртуальные и запечатанные события. Примеры.
34. События. Делегаты и события. EventHandler и EventArgs.
35. LINQ. Основные понятия.
36. LINQ. Fluent синтаксис. Использование лямбда-выражений.
37. LINQ. Выражения запросов. Анонимные типы.
38. LINQ. Подзапросы. Создание пользовательских LINQ.
39. LINQ. Отложенное выполнение.
40. Файловая система. Работа с дисками и каталогами.
41. Файловая система. Работа с файлами. Чтение и запись.
42. Сборка мусора. Основные функции сборщика мусора. Преимущества и основные этапы сборки мусора.
43. Сборка мусора. Корневые элементы. Large object heap.
44. Сборка мусора. Основные этапы сборки мусора. Поколения.
45. Финализация. Основные понятия и этапы.

46. Неуправляемые ресурсы. Шаблон dispose.
47. Windows Forms. Общие понятия. Структура проекта. Форма.
48. Windows Forms. Элементы управления. Описание пользовательского элемента управления.
49. Windows Forms. Свойства форм и элементов управления. События и их обработчики.
50. Xamarin. Основные понятия. Достоинства.
51. Xamarin. Архитектура Xamarin приложений.
52. Xamarin. Конструктор Android Designer. Библиотеки Xamarin.
53. Xamarin. Структура проекта Xamarin. XAML.
54. Архитектурный шаблон MVC. Основные понятия. Преимущества. Реализация MVC в ASP.NET.
55. ASP.NET. Структура проекта ASP.NET.
56. Маршрутизация URL. Маршрутизация в проектах MVC и Razor Pages.