

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В. Ф. УТКИНА

Кафедра электронных вычислительных машин

«СОГЛАСОВАНО»

Декан факультета ВТ

 Д.А. Перепелкин

«__» _____ 2020 г.

«УТВЕРЖДАЮ»

Проректор по РОПиМД

 А.В. Корячко

«__» _____ 2020 г.



Заведующий кафедрой ЭВМ

 Б.В. Костров

«__» _____ 2020 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

Б1.О.09 «Визуальное программирование»

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

ОПОП - «Математическое обеспечение и администрирование информационных систем»

Квалификация выпускника – бакалавр

Форма обучения – очно-заочная

Рязань 2020 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017г. № 809.

Программу составил
д.т.н., проф. кафедры ЭВМ



Б.В. Костров

Программа рассмотрена и одобрена на заседании кафедры ЭВМ
«11» 06 2020 г., протокол № 10

Заведующий кафедрой
«Электронные вычислительные машины»,
д.т.н., проф. кафедры ЭВМ



Б.В. Костров

1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Рабочая программа по дисциплине «Визуальное программирование» является составной частью основной профессиональной образовательной программы (ОПОП) бакалавриата «Математическое обеспечение и администрирование информационных систем», разработанной в соответствии с ФГОС ВО по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Целью изучения дисциплины «Визуальное программирование» является получение студентами знаний по основам разработки программного обеспечения, элементам визуального проектирования приложений и их использования на практике.

Обучение студентов по дисциплине «Визуальное программирование» направлено на получение представлений об основных понятиях и представлениях объектно-ориентированного подхода к разработке приложений, а также технологии быстрой разработки приложений RAD.

Перечень планируемых результатов обучения по дисциплине

Коды компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-2	Способность применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	<u>Знать:</u> математические основы программирования и языков программирования; <u>Уметь:</u> использовать данный математический аппарат в профессиональной деятельности; <u>Владеть:</u> навыками применения данного математического аппарата для решения конкретных задач.
ОПК-3	Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	<u>Знать:</u> основные положения и концепции в прикладного и системного программирования, технологии создания и эксплуатации программных продуктов и программных комплексов; <u>Уметь:</u> уметь использовать их в профессиональной деятельности; <u>Владеть:</u> практическими навыками разработки программного обеспечения.

2 Место дисциплины в структуре ОПОП

Дисциплина «Визуальное программирование» относится к обязательной части блока Б1 (Б1.О.25) профессионального цикла дисциплин основной образовательной программы подготовки бакалавров направления 02.03.02 «Математическое обеспечение и администрирование

информационных систем».

Для изучения дисциплины обучаемый должен

знать:

- основы структурного программирования;
- базовые понятия программирования на языке высокого уровня;

уметь:

- разрабатывать алгоритмы;
- выполнять декомпозицию предметной области задачи;

владеть:

- навыками обобщения информации.

Данная дисциплина базируется на следующих дисциплинах учебного плана: «Информатика», «Вычислительная математика».

Знания, полученные по освоению дисциплины, необходимы при изучении дисциплин: «Технологии разработки информационных систем» и «Программирование Web-приложений».

3 Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 6 зачетных единицы (ЗЕ), или 216 часа.

Вид учебной работы	Всего часов
Общая трудоемкость дисциплины	216
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	82,35
лекции	48
практические занятия	32
лабораторные работы	16
консультации	2
иная контактная работа (промежуточная аттестация)	0,35
2. Самостоятельная работа обучающихся (всего), в том числе:	98
курсовой проект (работа)	-
иная самостоятельная работа	98
3. Контроль	35,65
Вид промежуточной аттестации обучающегося	экзамен

4 Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Содержание дисциплины, структурированное по темам

Раздел дисциплины	Содержание
Тема 1. Основы работы с платформой .Net Framework	Платформа .NET Framework. Управляемые модули, MSIL код и метаданные. Visual C#. Сборки в .NET. Инструменты, предоставляемые .NET Framework. Создание приложения .NET Framework.

Тема 2. Переменные и выражения в языке C#	Объявления переменных. Время жизни и область видимости. Локальные и глобальные переменные. Константы. Построение выражений. Операции и их приоритеты.
Тема 3. Операторы языка C#	Операторы языка C#. Оператор присваивания. Составной оператор. Пустой оператор. Операторы выбора. Операторы перехода. Операторы циклов.
Тема 4. Процедуры и функции в языке C#	Процедуры и функции. Понятие методов класса. Атрибуты доступа. Формальные аргументы. Вызов процедур и функций. Фактические аргументы. Перегрузка методов. Рекурсия.
Тема 5. Массивы в языке C#	Одномерные и многомерные массивы. Динамические массивы. Классы-массивы. Массивы объектов.
Тема 6. Символы и строки в языке C#	Строки в C#. Строки постоянной и переменной длины. Классы char, char[]. Класс String. Изменяемые и неизменяемые строковые классы. Классы .Net Framework, расширяющие строковый тип.
Тема 7. Классы в языке C#	Описание класса в языке C#. Поля и методы класса. Конструкторы и деструкторы. Статические поля и методы. Статические конструкторы. Поля только для чтения. Закрытые поля. Стратегии доступа к полям класса. Структуры. Перечисления.
Тема 8. Отношение между классами в языке C#	Отношение наследования. Понятие статического и динамического связывания. Полиморфизм. Абстрактные классы. Множественное наследование. Интерфейсы.
Тема 9. Обработка исключительных ситуаций в языке C#	Корректность и устойчивость программных систем. Исключительные ситуации. Обработка исключительных ситуаций.
Тема 10. Работа с формами в языке C#	Основы работы с Visual Studio. Формы. Solution Explorer. Class View. Properties Window. Toolbox. Режимы дизайна и кода. Свойства проекта. Перемещение по коду, окна Types и Members. XML- документирование кода.
Тема 11. Работа с элементами управления в языке C#	Создание главного меню. Создание MDI-приложений. Контекстное меню. Диалоговые окна. StatusBar. CheckBox. Свойство TabIndex элементов управления. Label. PictureBox. Режимы запуска окон. ToolBar. ImageList. Добавление элементов управления в режиме работы приложения. Элементы управления CheckBox, GroupBox, RadioButton, ComboBox.

4.2 Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
Тема 1. Основы работы с платформой .Net Framework	23	12	4	8	-	-	-	8	3
Тема 2. Переменные и выражения в языке C#	24	12	4	8	-	-	-	9	3
Тема 3. Операторы языка C#	24	12	4	8	-	-	-	9	3
Тема 4. Процедуры и функции в языке C#	24	12	4	8	-	-	-	9	3
Тема 5. Массивы в языке C#	18	6	4	-	2	-	-	9	3
Тема 6. Символы и строки в языке C#	18	6	4	-	2	-	-	9	3
Тема 7. Классы в языке C#	18	6	4	-	2	-	-	9	3
Тема 8. Отношение между классами в языке C#	18,5	6,5	4	-	2	0,5	-	9	3
Тема 9. Обработка исключительных ситуаций в языке C#	18,5	6,5	4	-	2	0,5	-	9	3
Тема 10. Работа с формами в языке C#	20,5	7,5	4	-	3	0,5	-	9	4
Тема 11. Работа с элементами управления в языке C#	25,15	11,5	8	-	3	0,5	-	9	4,65
Промежуточная аттестация	0,35	0,35	-	-	-	-	0,35	-	-
Итого	216	82,35	48	32	16	2	0,35	98	35,65

Виды практических и самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 1. Основы работы с платформой .Net Framework	Практические работы	Знакомство со средой визуального программирования Visual Studio	8
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям	4 4
Тема 2. Переменные и выражения в языке C#	Практическая работа	Работа со списками с помощью элемента управления ListView среды Visual Studio	8
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям	4 5

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 3. Операторы языка C#	Практическая работа	Работа с древовидными структурами с помощью элемента управления TreeView среды Visual Studio	8
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям Консультации	3 4
Тема 4. Процедуры и функции в языке C#	Практическая работа	Использование интерфейсных элементов управления среды Visual Studio	8
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям Консультации	3 4
Тема 5. Массивы в языке C#	Лабораторная работа	Методы и свойства класса GRAPHICS в языке C# в среде Visual Studio	2
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям Консультации	3 4
Тема 6. Символы и строки в языке C#	Лабораторная работа	Инструменты для рисования в языке C# в среде Visual Studio	2
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям Консультации	3 4
Тема 7. Классы в языке C#	Лабораторная работа	Принципы объектно-ориентированного проектирования программ на языке C# в среде Visual Studio	2
	Самостоятельная работа	Изучение конспекта лекций	5
		Изучение методических указаний, подготовка к практическим занятиям Консультации	2 2
Тема 8. Отношение между классами в языке C#	Лабораторная работа	Объектная модель языка C#. Наследование. Абстрактные классы	2
	Самостоятельная работа	Изучение конспекта лекций	5
		Изучение методических указаний, подготовка к практическим занятиям Консультации	2 2
Тема 9. Обработка исключительных ситуаций в языке C#	Лабораторная работа	Обработка строк и исключительных ситуаций на языке C# в среде Visual Studio	2
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям Консультации	3 4
Тема 10. Работа с формами в языке C#	Лабораторная работа	Разработка приложения с однодокументным интерфейсом и модальными формами на языке C# в среде Visual Studio	3

Тема	Вид работы	Наименование и содержание работы	Трудо-емкость, часов
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям	3
		Консультации	4
Тема 11. Работа с элементами управления в языке C#	Лабораторная работа	Создание MDI-приложений	3
	Самостоятельная работа	Изучение конспекта лекций	2
		Изучение методических указаний, подготовка к практическим занятиям	3
		Консультации	4

5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

1. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 8. № 4867. РГРТУ, Рязань, 2014.
2. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 9. № 4905. РГРТУ, Рязань, 2015.
3. Гринченко Н.Н., Конкин Ю.В. Разработка моделей информационных систем на языке UML: учеб. пособие / РГРТУ. - Рязань, 2015. - 48с.

6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств приведен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Объектно-ориентированное программирование»).

7 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная учебная литература

1. Вендров А.М. Современные технологии создания программного обеспечения [Электронный ресурс] / «CitForum». URL: <http://citforum.ru/programming/application/program/> (дата обращения: 01.12.2014).
2. Липаев В.В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс]: учебное пособие/ Липаев В.В.- Электрон. текстовые данные.- М.: МАКС Пресс, 2014.- 309 с.- Режим доступа: <http://www.iprbookshop.ru/27297>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).
3. Алексеев В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений [Электронный ресурс]/ Алексеев В.Е., Таланов В.А.- Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.- 153 с.- Режим доступа: <http://www.iprbookshop.ru/52186>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).
4. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual C# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.- Электрон. текстовые

данные.- М.: Московский государственный строительный университет, ЭБС АСВ, 2011.- 180 с.- Режим доступа: <http://www.iprbookshop.ru/19258>.- ЭБС "IPRbooks" (дата обращения: 21.06.2016).

5. Гринченко Н.Н., Конкин Ю.В. Разработка моделей информационных систем на языке UML: учеб. пособие / РГРТУ. - Рязань, 2015. - 48с.
6. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 8. № 4867. РГРТУ, Рязань, 2014.
7. Конкин Ю.В. . Объектно-ориентированное программирование. Методические указания к лабораторной работе. Часть 9. № 4905. РГРТУ, Рязань, 2015.

Дополнительная учебная литература

1. Э. Фримен, К. Сиерра, Б. Бейтс Паттерны проектирования. – Спб.: «Питер», 2016. - 656с.
2. Аникеев С.В. Разработка приложений баз данных в Delphi : самоучитель. - М. : Диалог-МИФИ, 2013. - 160с.
3. Б. Маклафлин, Г. Поллайс, Д. Уэст Объектно-ориентированный анализ и проектирование. – Спб.: «Питер», 2014. - 285с.
4. М. Вайсфельд Объектно-ориентированное мышление. – Спб.: «Питер», 2014. - 340с.

8 ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

9 Методические указания для обучающихся по освоению дисциплины

Рекомендации по планированию и организации времени, необходимого для изучения дисциплины

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Изучение конспекта лекции в тот же день, после лекции (10 – 15 минут).

Изучение конспекта лекции за день перед следующей лекцией (10 – 15 минут).

Изучение теоретического материала по учебнику и конспекту (1 час в неделю).

Описание последовательности действий студента («сценарий изучения дисциплины»)

При изучении дисциплины рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном занятии. Тогда лекция будет гораздо понятнее. Но легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного его усвоения рекомендуется следующая последовательность действий.

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сего-

дня (10 – 15 минут).

2. При подготовке к лекции следующего дня нужно просмотреть текст предыдущей лекции, подумать о том, какой может быть тема следующей лекции (10 – 15 минут).

В течение недели выбрать время (1 час) для работы с литературой.

Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по теории принятия решений. Литературу по дисциплине необходимо читать только в бумажном (не в электронном) виде. Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Кроме того, полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?».

10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

При проведении занятий по дисциплине используются следующие информационные технологии:

– удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;

– доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам.

Перечень лицензионного программного обеспечения:

- 1) Операционная система Windows XP Professional (лицензия Microsoft DreamSpark Membership ID 700102019);
- 2) Open Office (лицензия Apache License, Version 2.0);
- 3) Система программирования Visual Studio.

10 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

- 1) Интерактивная доска.
- 2) Мультимедиа-проектор.

Программа составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем» (квалификация выпускника – бакалавр, форма обучения – очная, срок обучения – 4 года).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.О.25 «Визуальное программирование»

Направление подготовки

02.03.03 «Математическое обеспечение и администрирование информационных систем»

ОПОП академического бакалавриата

«Математическое обеспечение и администрирование информационных систем»

Квалификация выпускника – бакалавр

Форма обучения – 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 общекультурных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена. Форма проведения экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

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

Описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	Задача решена верно
2 балла (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На промежуточную аттестацию (экзамен, зачет) выносятся тест, два теоретических вопроса и 2 задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Оценка «отлично» выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

Промежуточная аттестация в форме зачета переводится в форму «зачтено», «незачтено»

Оценка «зачтено» выставляется студенту, который набрал в сумме более 5 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «незачтено» выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Тема 1. Основы работы с платформой .Net Framework	ОПК-2, ОПК-3	Зачет
Тема 2. Переменные и выражения в языке C#	ОПК-2, ОПК-3	Зачет
Тема 3. Операторы языка C#	ОПК-2, ОПК-3	Зачет
Тема 4. Процедуры и функции в языке C#	ОПК-2, ОПК-3	Зачет
Тема 5. Массивы в языке C#	ОПК-2, ОПК-3	Зачет
Тема 6. Символы и строки в языке C#	ОПК-2, ОПК-3	Зачет
Тема 7. Классы в языке C#	ОПК-2, ОПК-3	Зачет
Тема 8. Отношение между классами в языке C#	ОПК-2, ОПК-3	Зачет
Тема 9. Обработка исключительных ситуаций в языке C#	ОПК-2, ОПК-3	Зачет
Тема 10. Работа с формами в языке C#	ОПК-2, ОПК-3	Зачет

Тема 11. Работа с элементами управления в языке С#	ОПК-2, ОПК-3	Зачет
--	--------------	-------

ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме зачета

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

Типовые тестовые вопросы:

1. Основные принципы объектно-ориентированного программирования включают в себя:

+ инкапсуляцию, наследование, полиморфизм;
 виртуализацию, полиморфизм, наследование;
 абстрагирование, виртуализацию, наследование;
 наследование, виртуализацию, абстрагирование.

2. Свойство системы, позволяющее объединить данные и методы, работающие с ними, в одном классе, а также организовывать к ним отдельный доступ для различных частей программы называется:

виртуализацией;
 полиморфизмом;
 +инкапсуляцией;
 наследованием.

3. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью называется:

инкапсуляцией;
 виртуализацией;
 полиморфизмом;
 +наследованием.

4. Свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта называется:

инкапсуляцией;
 +полиморфизмом;
 виртуализацией;
 наследованием.

5. Класс - это:

+тип данных, включающий описание свойств и функций объектов;
 множество объектов с общим состоянием и поведением;
 тип данных, включающий описание функций с возможностью их повторного использования;
 множество объектов, имеющих общие свойства и функции.

6. Объект - это:

абстрактный тип данных, включающий состояние и поведение;

набор данных и функций работы с ними;
+экземпляр класса;
тип данных, реализующий функции класса и хранящий его состояние.

7. Скрытая часть класса описывается:
по умолчанию до первого спецификатора;
после спецификатора public;
после спецификатора protected;
+после спецификатора private.

8. Защищенные элементы класса описываются:
после спецификатора public;
+ после спецификатора protected;
после спецификатора private;
по умолчанию до первого спецификатора.

9. Общедоступные элементы класса описываются:
после спецификатора protected;
после спецификатора private;
+после спецификатора public;
по умолчанию до первого спецификатора.

10. Сообщения от драйвера мыши всегда передаются:
первому экранному элементу;
+функции окна, над которым находится курсор мыши;
экранному элементу, имеющему фокус ввода;
экранному элементу, способному получать сообщения от клавиатуры.

Типовые практические задания:

Задание 1

Разработать программу на языке C# в системе программирования Visual Studio, в которой выполняется открытие и чтение информации из текстового файла. Доступ к файлу должен контролироваться обработчиками исключительных ситуаций.

Критерии выполнения задания 1

Задание считается выполненным, если обучающийся использовал в программе операторы обрабатывающей и завершающей конструкций обработки исключений.

Задание 2

Разработать программу на языке C# в системе программирования Visual Studio, в которой выполняется создание и вывод на экран списка студентов. Абстракция студент должна быть представлена классом.

Критерии выполнения задания 2

Задание считается выполненным, если обучающийся организовал обработку списка на основе процедур, представленных методами соответствующего класса.

Типовые теоретические вопросы:

1. Основные этапы развития программирования.
2. Структурная декомпозиция
3. Модульное программирование.

4. Отличительные особенности объектно-ориентированного подхода в программировании.
5. Архитектура объектно-ориентированных программ.
6. Основные принципы объектно-ориентированного программирования.
7. Этапы разработки программного обеспечения при ООП.
8. Понятие объектной декомпозиции. Пример.
9. Понятие объекта. Объекты и сообщения. Состояние и поведение объектов.
10. Понятие объекта. Операции над объектами.

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

Типовые тестовые вопросы:

1. К общедоступным свойствам класса (public) разрешен доступ:
 - только из методов данного класса и классов потомков;
 - только из методов данного класса;
 - только из методов данного класса и классов предков;
 - +из всех методов программы.

2. К защищенным свойствам класса (protected) разрешен доступ:
 - из всех методов программы;
 - только из методов данного класса;
 - только из методов данного класса и класса предка;
 - +только из методов данного класса и классов производных от данного.

3. К закрытым свойствам класса (private) разрешен доступ:
 - только из методов данного класса, классов потомков и классов предков;
 - +только из методов данного класса
 - из всех методов программы
 - только из методов данного класса и классов потомков;

4. Методам базового класса разрешен доступ:
 - +ко всем свойствам и методам базового класса;
 - только к защищенным свойствам и методам базового класса;
 - только к закрытым свойствам и методам базового класса;
 - только к защищенным и общим свойствам и методам базового класса.

5. Методам класса потомка разрешен доступ:
 - ко всем свойствам и методам базового класса;
 - +к защищенным и общим свойствам и методам базового класса;
 - только к защищенным свойствам и методам базового класса;
 - к закрытым и общим свойствам и методам базового класса.

7. Конструктор класса используется для:
 - +инициализации объектов класса;
 - инициализации класса для наследования;
 - инициализации общих свойств класса;
 - начального присвоения значений переменным класса.

8. Деструктор класса используется для:
удаления объектов базового класса;
+освобождение памяти, выделяемой при инициализации объектов класса;
освобождение памяти, выделяемой для динамических переменных класса;
удаление динамических переменных и указателей класса.

9. Абстрактный класс предназначен для:
создания объектов базового класса;
создания одного экземпляра базового класса;
+определения реализации части функций в классах потомках;
определения общих свойств для всех классов проекта.

10. Абстрактный класс не может быть использован для:
определения реализации части функций в классах потомках;
определения данного класса как базового;
инициализации общих свойств базового класса;
+создания объектов.

Типовые практические задания:

Задание 1

Разработать приложение, отображающее последовательность заданных изображений в формате BMP. Порядок выбора и скорость смены изображений должны задаваться.

Критерии выполнения задания 1

Задание считается выполненным, если обучающийся разработал структуру класса для работы с изображениями и использовал управляющие экранные элементы для работы с изображениями.

Задание 2

Подготовить и выполнить программу на языке C# в системе программирования Visual Studio построения графика функции $Y=F(X)$, которая задается следующим образом:

$$\begin{aligned} 0 <= X < 50 & \quad Y = 0, \\ 50 <= X < 100 & \quad Y = X - 50, \\ 100 <= X < 150 & \quad Y = 50 \\ 150 <= X < 200 & \quad Y = -(X - 200), \\ 200 <= X < 250 & \quad Y = 0. \end{aligned}$$

Критерии выполнения задания 2

Задание считается выполненным, если обучающийся разработал алгоритм табулирования функции и вывода графика на экран.

Типовые теоретические вопросы:

1. Понятие класса. Структура и организация определения класса.
2. Определение классов и объектов на языке C#.
3. Конструирование и уничтожение объектов в языке C#.
4. Особенности работы с объектами в языке C#.
5. Средства разработки классов. Наследование. Инициализация объектов производных классов. Управление доступом к элементам класса в связи с наследованием.
6. Полиморфизм как средство разработки классов.
7. Понятие о статическом и динамическом связывании.
8. Полиморфизм. Пример.
9. Виртуальные методы как средство реализации сложного полиморфизма.

10. Обработка исключительных ситуаций в программах на языке ООП.