

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Юридический институт

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка приложений на языке Си

Кафедра информационного права и информатики юридического института

Образовательная программа бакалавриата
09.03.03 Прикладная информатика

Направленность (профиль) программы
Прикладная информатика в юриспруденции

Форма обучения
очная

Статус дисциплины: дисциплина по выбору

Махачкала, 2022

Рабочая программа дисциплины «Разработка приложений на языке Си» составлена в 2022 году в соответствии с требованиями ФГОС ВО – бакалавриат по направлению подготовки 09.03.03 Прикладная информатика от «19» сентября 2017 г. № 922.

Разработчик(и): кафедра «Информационного права и информатики», Везиров Тельман Тимурович, кандидат педагогических наук, доцент

Рабочая программа дисциплины одобрена:
на заседании кафедры информационного права и информатики
от «25» 02 2022 г., протокол № 7

Зав. кафедрой  Абдусаламов Р.А.
(подпись)

на заседании Методической комиссии юридического института
от «21» 03 2022 г., протокол № 4.

Председатель  Арсланбекова А.З.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением
« 31 » 03 2022 г.

(Начальник УМУ  Гасангаджиева А.Г.
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина «Разработка приложений на языке Си» является дисциплиной по выбору ОПОП бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

Дисциплина реализуется на юридическом институте кафедрой информационного права и информатики.

Содержание дисциплины охватывает круг вопросов, связанных по разработке визуального программного обеспечения на платформе .NET Framework с использованием языка С#, которое должно способствовать формированию у студентов основ научного мышления, в том числе: пониманию основных сведений о принципах построения визуального программного обеспечения, особенностях организации процесса программирования на платформе .NET Framework; умению оценивать степень правильности формирования программного кода и визуального отображения информации.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных – ОПК-8, ПК-2, ПК-6, ПК-11.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме устного опроса, лабораторной работы, тестирования и промежуточный контроль в форме зачета.

Объем дисциплины 3 зачетных единиц, в том числе в академических часах по видам учебных занятий:

Очная форма обучения

Семестр	Учебные занятия							Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)
	в том числе:							
	всего	Контактная работа обучающихся с преподавателем					СРС, в том числе экзамен	
		всего	из них					
		Лекции	Лабораторные занятия	Практические занятия	КСР	консультации		
7	108	48	16	16	16		60	Зачет

1. Цели освоения дисциплины

Целями освоения дисциплины «Разработка приложений на языке Си» являются:

- изучение программирования на языке C#;
- освоение интегрированной среды разработки (ИСР) из Visual Studio .Net для языка Visual C#, работающего с платформой. Net. Framework;
- получение навыков в разработке программ на языке C#.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Разработка приложений на языке Си» является дисциплиной по выбору ОПОП бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

Предшествующими, на которых непосредственно базируется дисциплина «Разработка приложений на языке Си», является дисциплины «Информатика и программирование», «Объектно-ориентированное проектирование и программирование».

Студенты смогут использовать полученные знания при подготовке выпускных квалификационных работ.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения и процедура освоения).

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
ОПК-8.	ОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы.	Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы.	Устный опрос, письменный опрос, тестирование, подготовка реферата, выполнение лабораторных работ.
	ОПК-8.2. Умеет осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы.	Умеет осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы.	

	цикла информационной системы.		
	ОПК-8.3. Владеет навыками составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.	Владеет навыками составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла	
ПК-2.	ИПК- 2.1. Знает принципы организации проектирования и содержание этапов процесса разработки прикладных программ.	Знать: принципы разработки программного обеспечения, концепции и понятия объектно-ориентированного подхода к программированию, механизмы его реализации в языке программирования	Устный опрос, письменный опрос, тестирование, подготовка реферата, выполнение лабораторных работ.
	ИПК- 2.2. Умеет разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования.	Уметь: создавать приложения на различных языках программирования, использовать основные принципы объектно-ориентированного подхода при написании программ; проектировать и реализовывать программы со сложной иерархией классов и объектов	
	ИПК- 2.3. Владеет навыками проектирования и разработки прикладного программного обеспечения с использованием современных технологий программирования	Владеть: навыками анализа поставленных задач, проектирования и разработки приложений, приемами разработки программных комплексов для решения прикладных задач, методами использования современных технологий	

		программирования, тестирования и документирования программных комплексов	
ПК-6.	ИПК- 6.1. Знает основные сведения о методах и способах построения эффективных алгоритмов для решения прикладных задач.	Знать: проблемы и процессы анализа предметной области программных решений современные подходы анализа предметной области программных решений	Устный опрос, письменный опрос, тестирование, подготовка реферата, выполнение лабораторных работ.
	ИПК- 6.2. Умеет создавать программные прототипы решения задач предметной области.	Уметь: разрабатывать программные приложения для предметной области; производить анализ сложности алгоритма и находить пути упрощения полученных алгоритмов	
	ИПК- 6.3. Владеет практическими навыками разработки программных прототипов решения прикладных задач	Владеть: практическими навыками использования языков программирования для создания программные прототипов решения прикладных задач; основные и наиболее популярные программные продукты, позволяющие проектировать и разрабатывать алгоритмы.	
ПК-11.	ИПК- 11.1. Знать: принципы системного подхода и математические методы в формализации решения прикладных задач, в обосновании правильности выбранной модели	Знать: принципы системного подхода и математические методы в формализации решения прикладных задач, в обосновании правильности выбранной модели	Устный опрос, письменный опрос, тестирование, подготовка реферата, выполнение лабораторных работ.

	информационных процессов и систем;	информационных процессов и систем;	
	ИПК- 11.2. Уметь: применять системный подход и математические методы в формализации решения прикладных задач	Уметь: применять системный подход и математические методы в формализации решения прикладных задач	
	ИПК- 11.3. Владеть: методами построения математической модели профессиональных задач и содержательной интерпретации полученных результатов, навыками разработки информационно-логической, функциональной и объектно-ориентированной модели информационной системы, модели данных информационных систем.	Владеть: методами построения математической модели профессиональных задач и содержательной интерпретации полученных результатов, навыками разработки информационно-логической, функциональной и объектно-ориентированной модели информационной системы, модели данных информационных систем.	

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 3 зачетных единиц, 108 академических часов.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/п	Разделы и темы дисциплины по модулям	Семестр	Виды учебной работы, включая самостоятельную работу студентов (в часах)					Формы текущего контроля успеваемости и промежуточной аттестации
			Лекции	Практические занятия	Лабораторные занятия	...	Самостоятельная работа в т.ч. экзамен	
Модуль 1. Основы программирования на языке C#								
1	Основы программирования на языке C#	7	6	6	6		18	Устный опрос, письменный опрос,

								тестирование, выполнение лабораторных работ.
	<i>Итого по модулю 1:</i>		6	6	6		18	
Модуль 2. Объектно-ориентированное программирование на языке C#								
1	Объектно-ориентированное программирование на языке C#	7	4	4	4		24	Устный опрос, письменный опрос, тестирование, выполнение лабораторных работ.
	<i>Итого по модулю 1:</i>		4	4	4		24	
Модуль 3. Дополнительные возможности объектно-ориентированного программирования на языке C#								
1	Обработка исключений	7	2	2	2		6	Устный опрос, письменный опрос, тестирование, выполнение лабораторных работ.
2	Делегаты, события и лямбды	7	2	2	2		6	Устный опрос, письменный опрос, тестирование, выполнение лабораторных работ.
3	Интерфейсы	7	2	2	2		6	Устный опрос, письменный опрос, тестирование, выполнение лабораторных работ.
	<i>Итого по модулю 3:</i>		6	6	6		18	
	ИТОГО:		16	16	16		60	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине.

Модуль 1. Основы программирования на языке C#

Тема 1. Основы программирования на языке C#

Язык C# и платформа .NET. Среда VisualStudio.NET.

Структура программы. Переменные и константы. Литералы. Типы данных. Консольный ввод-вывод. Арифметические операции. Поразрядные операции. Операции присваивания. Преобразования базовых типов данных. Условные выражения. Конструкция if..else и тернарная операция. Циклы. Массивы. Задачи с массивами. Методы. Параметры методов. Возвращение значения и оператор return. Передача параметров по ссылке и значению. Выходные параметры. Массив параметров и ключевое слово params. Рекурсивные функции. Локальные функции. Конструкция switch. Перечисления enum.

Модуль 2. Объектно-ориентированное программирование на языке C#

Тема 2. Объектно-ориентированное программирование на языке C#

Классы и объекты. Конструкторы, инициализаторы и деструкторы. Класс Program и метод Main. Программы верхнего уровня. Структуры. Типы значений и ссылочные типы. Область видимости (контекст) переменных. Пространства имен. Глобальные пространства имен. Подключение пространств имен по умолчанию. Создание библиотеки классов. Модификаторы доступа. Свойства. Перегрузка методов. Статические члены и модификатор static. Поля и структуры для чтения. Перегрузка операторов. Индексаторы. Наследование. Преобразование типов. Перегрузка операций преобразования типов. Виртуальные методы и свойства. Соккрытие методов. Различие переопределения и сокращения методов. Абстрактные классы. Класс System.Object и его методы. Обобщенные типы. Ограничения обобщений. Наследование обобщенных типов.

Модуль 3. Дополнительные возможности объектно-ориентированного программирования на языке C#

Тема 3. Обработка исключений

Конструкция try..catch..finally. Блок catch и фильтры исключений. Типы исключений. Класс Exception. Создание классов исключений. Поиск блока catch при обработке исключений. Генерация исключения и оператор throw.

Тема 4. Делегаты, события и лямбды

Делегаты. Применение делегатов. Анонимные методы. Лямбды. События. Ковариантность и контравариантность делегатов. Делегаты Action, Predicate и Func. Замыкания.

Тема 5. Интерфейсы

Определение интерфейсов. Применение интерфейсов. Явная реализация интерфейсов. Реализация интерфейсов в базовых и производных классах. Наследование интерфейсов. Интерфейсы в обобщениях. Копирование объектов. Интерфейс `ICloneable`. Сортировка объектов. Интерфейс `IComparable`. Ковариантность и контравариантность обобщенных интерфейсов.

4.3.2. Содержание практических занятий по дисциплине.

Модуль 1. Основы программирования на языке C#

Тема 1. Основы программирования на языке C#

1. Язык C# и платформа .NET. Среда VisualStudio.NET.
2. Структура программы.
3. Переменные и константы.
4. Консольный ввод-вывод.
5. Арифметические операции.
6. Поразрядные операции.
7. Преобразования базовых типов данных.
8. Условные выражения.
9. Массивы. Задачи с массивами.
10. Методы. Параметры методов.

Модуль 2. Объектно-ориентированное программирование на языке C#

Тема 2. Объектно-ориентированное программирование на языке C#

1. Классы и объекты.
2. Конструкторы, инициализаторы и деструкторы.
3. Область видимости (контекст) переменных.
4. Пространства имен. Глобальные пространства имен.
5. Создание библиотеки классов. Модификаторы доступа.
6. Перегрузка методов. Статические члены и модификатор `static`.
7. Поля и структуры для чтения. Перегрузка операторов.
8. Индексаторы. Наследование. Преобразование типов.
9. Перегрузка операций преобразования типов.
10. Виртуальные методы и свойства. Соккрытие методов.
11. Различия переопределения и сокращения методов.
12. Абстрактные классы. Класс `System.Object` и его методы.
13. Обобщенные типы. Ограничения обобщений.

14. Наследование обобщенных типов.

Модуль 3. Дополнительные возможности объектно-ориентированного программирования на языке C#

Тема 3. Обработка исключений

1. Конструкция try..catch..finally.
2. Блок catch и фильтры исключений.
3. Типы исключений. Класс Exception.
4. Создание классов исключений.
5. Поиск блока catch при обработке исключений.
6. Генерация исключения и оператор throw.

Тема 4. Делегаты, события и лямбды

1. Делегаты. Применение делегатов.
2. Анонимные методы.
3. Лямбды. События.
4. Ковариантность и контравариантность делегатов.
5. Делегаты Action, Predicate и Func. Замыкания.

Тема 5. Интерфейсы

1. Определение интерфейсов. Применение интерфейсов.
2. Явная реализация интерфейсов.
3. Реализация интерфейсов в базовых и производных классах.
4. Наследование интерфейсов. Интерфейсы в обобщениях.
5. Копирование объектов. Интерфейс ICloneable.
6. Сортировка объектов. Интерфейс IComparable.
7. Ковариантность и контравариантность обобщенных интерфейсов.

4.3.3. Содержание лабораторных занятий по дисциплине.

Модуль 1. Основы программирования на языке C#

Тема 1. Основы программирования на языке C#

1. Язык C# и платформа .NET. Среда VisualStudio.NET.
2. Структура программы.
3. Переменные и константы.
4. Консольный ввод-вывод.
5. Арифметические операции.
6. Поразрядные операции.
7. Преобразования базовых типов данных.
8. Условные выражения.

9. Массивы. Задачи с массивами.
10. Методы. Параметры методов.

Модуль 2. Объектно-ориентированное программирование на языке C#

Тема 2. Объектно-ориентированное программирование на языке C#

1. Классы и объекты.
2. Конструкторы, инициализаторы и деструкторы.
3. Область видимости (контекст) переменных.
4. Пространства имен. Глобальные пространства имен.
5. Создание библиотеки классов. Модификаторы доступа.
6. Перегрузка методов. Статические члены и модификатор static.
7. Поля и структуры для чтения. Перегрузка операторов.
8. Индексаторы. Наследование. Преобразование типов.
9. Перегрузка операций преобразования типов.
10. Виртуальные методы и свойства. Скрытие методов.
11. Различия переопределения и сокрытия методов.
12. Абстрактные классы. Класс System.Object и его методы.
13. Обобщенные типы. Ограничения обобщений.
14. Наследование обобщенных типов.

Модуль 3. Дополнительные возможности объектно-ориентированного программирования на языке C#

Тема 3. Обработка исключений

1. Конструкция try..catch..finally.
2. Блок catch и фильтры исключений.
3. Типы исключений. Класс Exception.
4. Создание классов исключений.
5. Поиск блока catch при обработке исключений.
6. Генерация исключения и оператор throw.

Тема 4. Делегаты, события и лямбды

1. Делегаты. Применение делегатов.
2. Анонимные методы.
3. Лямбды. События.
4. Ковариантность и контравариантность делегатов.
5. Делегаты Action, Predicate и Func. Замыкания.

Тема 5. Интерфейсы

1. Определение интерфейсов. Применение интерфейсов.

2. Явная реализация интерфейсов.
3. Реализация интерфейсов в базовых и производных классах.
4. Наследование интерфейсов. Интерфейсы в обобщениях.
5. Копирование объектов. Интерфейс ICloneable.
6. Сортировка объектов. Интерфейс IComparable.
7. Ковариантность и контравариантность обобщенных интерфейсов.

5. Образовательные технологии

В соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика в рамках изучения данной дисциплины для реализации компетентностного подхода предусмотрено все проводимые занятия, в том числе самостоятельная работа студентов, сочетать передовые методические приемы с новыми образовательными информационными технологиями и достижениями науки и техники.

№ п/п	Вид учебной работы	Образовательные технологии
1.	Лекции	<ul style="list-style-type: none"> • Вводная лекция • Лекция-информация с визуализацией • Проблемная лекция
2.	Практические занятия	<ul style="list-style-type: none"> • Семинар-дискуссия, семинар-дебаты по актуальным проблемам искусственного интеллекта • Выполнение практических работ • Решение проблемных ситуаций из области искусственного интеллекта • Мини-конференция по студенческим докладам • Проектные технологии
3.	Самостоятельная работа	<ul style="list-style-type: none"> • Поиск и анализ информации в сети Интернет • Освоение теоретического материала • Подготовка к практическим занятиям • Выполнение домашних заданий • Выполнение творческой работы • Работа с электронным учебно-методическим комплексом • Подготовка к текущему и итоговому контролю
4.	Контроль	<ul style="list-style-type: none"> • Практические задания • Контрольная работа

	<ul style="list-style-type: none"> • Коллоквиум • Защита рефератов • Устный опрос • Письменный опрос
--	--

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Нормативные акты

1. "Конституция Российской Федерации" (принята всенародным голосованием 12.12.1993) (с учетом поправок, внесенных Законами РФ о поправках к Конституции РФ от 30.12.2008 N 6-ФКЗ, от 30.12.2008 N 7-ФКЗ, от 05.02.2014 N 2-ФКЗ, от 21.07.2014 N 11-ФКЗ) // Официальный текст Конституции РФ с внесенными поправками от 21.07.2014 опубликован на Официальном интернет-портале правовой информации <http://www.pravo.gov.ru>, 01.08.2014.

2. Гражданский кодекс Российской Федерации (часть четвертая)" от 18.12.2006 N 230-ФЗ (ред. от 23.05.2018) // "Российская газета", N 289, 22.12.2006.

3. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 19.07.2018) "Об информации, информационных технологиях и о защите информации" // "Российская газета", N 165, 29.07.2006.

Виды и порядок выполнения самостоятельной работы:

№ п/п	Вид самостоятельной работы	Вид контроля
1.	Подготовка реферата, презентации и доклада	Прием реферата, презентации, доклада и оценка качества их исполнения на мини-конференции
2.	Освоение теоретического материала	Устный опрос
3.	Подготовка к практическим занятиям	Практические задания
4.	Подготовка к текущему контролю	Контрольная работа. Коллоквиум. Тестирование

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Типовые контрольные задания

Примерная тематика рефератов

1. История развития языков ООП.
2. Виртуальная машина Java. Версии Java машины и их эволюция.
3. История и предпосылки появления Java.
4. Сферы применения Java в современном информационном мире.
5. Среда разработки NetBeans.
6. Среда разработки Eclipse.
7. Встроенные типы C#.
8. Среда от JetBrains для работы с платформой .NET.
9. Свободная среда разработки от Xamarin для создания приложений на C#.
10. История развития языка C#.
11. Сравнение языков ООП: Java, C#, Object Pascal.
12. Первые языки объектно-ориентированного программирования.
13. Особенности объектно-ориентированных языков.
14. Языки WEB-программирования.
15. История развития Visual Studio.

Примерные тестовые задания

Создателем языка C# является:

- (1) Дональд Кнут
- (2) Никлас Вирт
- (3) Андреас (Андерс) Хейлсберг
- (4) Бертран Мейер

Свойство открытости среды разработки Visual Studio 2008 означает:

- (1) возможность открытия проектов
- (2) возможность открытия файлов
- (3) возможность создания новых проектов;
- (4) возможность включения в среду разработки новых языков программирования и их компиляторов

Framework Class Library (FCL) - это библиотека классов, которая может быть использована:

- (1) только компиляторами среды разработки
- (2) только при разработке проектов на языке C#

(3) как компилятором языка C#, так и при разработке проектов на языке C#. Но она не может быть использована при работе с другими языками программирования, включенными в среду разработки

(4) как компиляторами, так и при разработке проектов на всех языках, включенных в среду разработки

(5) только классами этой же библиотеки

Среда разработки Visual Studio 2008 позволяет:

(1) открывать существующие проекты и работать с ними

(2) открывать существующие файлы и редактировать их

(3) создавать новые проекты;

(4) включать новые языки программирования и их компиляторы

Язык C#:

(1) не имеет ничего общего с языком C++

(2) полностью совпадает с языком C++

(3) унаследовал основные черты C++

(4) проще и надежнее языка C++

Язык C#:

(1) является полностью объектно-ориентированным языком

(2) учитывает все возможности Framework .Net

(3) допускает наследование

(4) допускает универсализацию

Язык C# в версии 3.0:

(1) предлагает новые типы проектов

(2) обладает чертами функционального языка программирования, допуская построение лямбда-выражений и анонимных типов и функций

(3) позволяет использовать инструмент LINQ (Language Integrated Query) для построения запросов к базам данных

Компонентами каркаса Framework .Net являются:

(1) библиотека классов FCL (Framework Class Library)

(2) библиотека классов MFC (Microsoft Foundation Classes)

(3) общезыковая исполнительная среда - CLR (Common Language Runtime)

(4) классы объектов ADO .Net

В Visual Studio 2008 число языков программирования, поставляемых корпорацией Microsoft:

- (1) увеличилось
- (2) осталось прежним
- (3) уменьшилось

На каких языках программирования можно работать в Visual Studio 2008:

- (1) только на языке C#
- (2) на трех языках - Visual C++, C#, Visual Basic
- (3) число языков не ограничено
- (4) на всех языках программирования, которые фирма Microsoft включит в состав Visual Studio

Каркас Framework .Net:

- (1) используется только при создании проектов на языке Net
- (2) является единым для всех языков программирования среды разработки
- (3) используется только при создании проектов на языке C#
- (4) используется только при создании проектов на языке C++

Каркас Framework .Net это:

- (1) надстройка над операционной системой
- (2) транслятор JIT, выполняющий трансляцию модулей, написанных на языке IL
- (3) библиотека классов, используемая компилятором
- (4) набор классов, используемых при построении самой Visual Studio

JIT - это:

- (1) компилятор одного из языков, включаемых в состав Visual Studio
- (2) компонент исполнительной среды CLR
- (3) компонент компилятора C#;
- (4) компилятор, используемый сборщиком мусора

Отметьте истинные высказывания:

- (1) важной частью библиотеки FCL являются классы, задающие примитивные типы, те типы, которые считаются встроенными в язык программирования
- (2) типы данных, создаваемые программистом, включаются в состав библиотеки FCL

(3) типы различных языков программирования могут проецироваться на один и тот же тип FCL

(4) частью библиотеки FCL являются структурные типы - строки, массивы, стеки, очереди, списки, деревья

Сборщик мусора Garbage Collector это:

- (1) компонент компилятора
- (2) компонент Visual Studio
- (3) компонент Framework .Net

Отметьте истинные высказывания:

(1) пространство имен - это модуль, объединяющий группу классов, как правило, близких по функциональности и имеющих уникальные имена внутри пространства имен

(2) пространство имен содержит классы, но не может содержать вложенные в него пространства имен

(3) пространство имен содержит не только классы, но может содержать вложенные в него пространства имен

Отметьте истинные высказывания:

(1) компилятор языка C# создает код на промежуточном языке IL (Intermediate Language)

(2) компилятор JIT (Just In Time), входящий в состав CLR, транслирует IL код в код текущей платформы с учетом ее особенностей

(3) двухэтапная компиляция позволяет улучшить эффективность и надежность создаваемого программного кода

(4) двухэтапная компиляция требует от программиста дополнительных усилий по улучшению кода на промежуточном языке IL

Отметьте истинные высказывания:

(1) для Visual Studio 2008 разработано много компиляторов разных языков программирования, и каждый программист при наличии компилятора может добавить его в состав используемой им студии

(2) компилятор C# транслирует исходный текст в код на языке IL

(3) класс, удовлетворяющий спецификациям CLS, называется CLS-совместимым

Спецификации CLS задают:

(1) спецификации класса, предназначенного для использования в проектах на разных языках программирования

(2) спецификации, которые безусловно должны выполняться программистами, работающими в среде Visual Studio 2008

(3) спецификации для программистов, работающих на языке C#

Отметьте истинные высказывания:

(1) для языков программирования, входящих в Visual Studio .Net, каждый компилятор использует собственную библиотеку классов FCL

(2) для языков программирования, входящих в Visual Studio .Net, компиляторы используют одну и ту же библиотеку классов FCL

(3) библиотека классов FCL является частью Visual Studio 2008

(4) библиотека классов FCL является частью каркаса Framework .Net, отделенного от Visual Studio 2008

Примерные контрольные задания

1. Разработки системы учета кадров в юридической организации.
2. Разработка информационной системы «КУП» (Книга учета преступлений).
3. Разработка информационной системы по хранению и учету документов юридического отдела предприятия.
4. Разработка информационной системы прокурорского надзора за прохождением судебных дел.
5. Разработка информационной системы мгновенной регистрации ДТП.
6. Разработка информационной системы расчет с клиентами нотариальной канторы.
7. Разработка информационной системы расчет с клиентами юридической консультации.
8. Разработка информационной системы регистрации дорожно-транспортных происшествий.
9. Разработка информационной системы регистрация разыскиваемых транспортных средств.
10. Разработка информационной системы регистрация уличных правонарушений.
11. Разработка информационной системы следственного делопроизводства.
12. Разработка информационной системы учета малолетних правонарушений.

13. Разработка информационной системы учета преступных организованных группировок.

14. Разработка модуля проверки наличия неуплаченных штрафов за дорожно-транспортные нарушения.

15. Разработка информационной системы учета разыскиваемых граждан.

16. Разработка информационной системы учета транспорта на штрафстоянке.

17. Разработка информационной системы электронного документооборота юридического отдела какого-либо предприятия (организации).

18-25. Создать информационно-поисковую систему. Информационно-поисковая система должна являться клиент-сервер приложением БД (созданным по архитектуре клиент-сервер либо файл-сервер). Она должна обеспечивать возможность ввода, редактирования, удаления, поиска, фильтрации данных, а так же вывода отчетов по заданным критериям (не менее 3х отчетов):

- Разработка информационной поисковой системы «Библиотека».
- Разработка информационной поисковой системы «Расписание занятий».
- Разработка информационной поисковой системы «Успеваемость».
- Разработка информационной поисковой системы «Склад оптовой базы».
- Разработка информационной поисковой системы «Поликлиника».
- Разработка информационной поисковой системы «Продажа видео и аудио продукции».
- Разработка информационной поисковой системы «ПО и состав ПК организации».
- Разработка информационной поисковой системы «Каталог музыкальных треков».

Контрольные вопросы к зачету

1. История возникновения ООП.
2. Тенденции середины 80-х связанные с популярностью ООП.
3. Набор основных принципов ООП по Алану Кею.
4. Понятие ООП. Классы и объекты.
5. Основные принципы ООП.
6. Достоинства и недостатки ООП.

7. Объявление классов в C#.
8. Создание объектов (экземпляров класса) в C#.
9. Члены класса: поля и константы.
10. Методы класса: создание простого метода.
11. Методы класса: создание статического метода.
12. Методы класса: создание метода возвращающего значения.
13. Создание конструкторов в C#.
14. Ключевое слово `this` и его применение в конструкторе.
15. Создание нескольких конструкторов класса.
16. Свойства класса.
17. Создание автоматических свойств класса.
18. Аксессоры `get` и `set`.
19. Наследование классов.
20. Конструкторы базового класса.
21. Доступ к членам базового класса из класса-наследника.
22. Среда разработки VisualStudio 2017.
23. Создание приложений в VisualStudio 2017.
24. Создание проекта web-приложения в среде VisualStudio 2017.
25. Размещение элементов управления и написание кода.
26. События `Application` и `Session`.
27. События серверных элементов управления.
28. Знакомство с пространством имен.
29. Обзор пространства имен `Web`.
30. Сохранение сведений о состоянии: использование файлов `cookie`.
31. Сохранение сведений о состоянии: использование переменных состояния `Application` и `Session`.
32. Использование элементов управления.
33. Проверка вводимых данных.
34. Переход между web-формами.
35. Доступ к данным с помощью `ADO.NET`.
36. Подключение к БД.
37. Создание набора данных.
38. Отображение набора данных.
39. Создание соединения с базой данных во время выполнения.
40. Использование наборов данных в web-формах.
41. Отображение набора данных с помощью `DataGrid`.
42. Отображение набора данных с помощью `DataList`.
43. Аутентификация и авторизация пользователей.

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля – 60 % и промежуточного контроля – 40 %.

Текущий контроль по дисциплине включает:

- посещение занятий – 5 баллов,
- участие на практических занятиях – 15 баллов,
- выполнение лабораторной работы – 25 баллов,
- написание реферата – 15 баллов.

Промежуточный контроль по дисциплине включает:

- контрольная работа – 20 баллов;
- компьютерное тестирование – 20 баллов.

8. Учебно-методическое обеспечение дисциплины.

а) адрес сайта курса

<http://edu.dgu.ru/course/view.php?id=3376>

б) основная литература:

1. Подбельский, В. В. Программирование. Базовый курс C#: учебник для вузов / В. В. Подбельский. — Москва: Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868> (дата обращения: 20.03.2021).

2. Туральчук, К. А. Параллельное программирование с помощью языка C# / К. А. Туральчук. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 189 с. — ISBN 978-5-4486-0506-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/79714.html> (дата обращения: 20.03.2021). — Режим доступа: для авторизир. Пользователей.

в) дополнительная литература:

1. Албахари Дж., Албахари Б. C# 7.0. Карманный справочник. Скорая помощь для программистов на C# 7.0. – М.: Изд-во: Альфа-Книга, 2017. - 224 с.: ISBN 978-5-9909-4461-9.

2. Вайсфельд М. Объектно-ориентированное мышление. — СПб.: Питер,

2017. — 304 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-496-00793-1.

3. Васильев А. Н. Java. Объектно-ориентированное программирование: Учебное пособие. — СПб.: Питер, 2017. — 400 с. ISBN 978-5-49807-948-6.

4. Майк МакГрат. Программирование на Java для начинающих. — М.: «Эксмо», 2016. — 193 с. — (Серия «Программирование для начинающих»). ISBN 978-5-699-85743-2.

5. Мейер Б. Почувствуй класс / Мейер Б.; пер. с англ. под ред. В.А. Биллига. - М.: Национальный Открытый Университет «ИНТУИТ»: БИНОМ. Лаборатория знаний, 2018. — 775 с.: ил., табл. ISBN 978-5-9963-0573-5.

6. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов. — СПб.: Питер, 2019. — 432 с.: ISBN 978-5-91180-174-8.

7. Стилмен Э., Грин Дж. Изучаем C#. 3-е издание. - СПб.: Питер, 2018. — 816 с.: ил. - (Серия «HeadFirstO'Reilly»). ISBN 978-5-4960-0867-9.

8. ШарпДж. MicrosoftVisualC#. Подробное руководство. — СПб.: Питер, 2019. — 848 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-4960-2372-6.

9. Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - СПб.: Питер, 2020. — 366 с.: ил. - (Серия «Библиотека программиста»). ISBN 978-5-459-01720-5, 978-5-496-00389-6.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах литературы, поступающих в фонд НБ ДГУ / Дагестанский государственный университет. — Махачкала, 2010 — Режим доступа: <http://elib.dgu.ru>, свободный (дата обращения: 21.03.2021).

2. Moodle [Электронный ресурс]: система виртуального обучения: [база данных] / Даг. Гос. ун-т. — Махачкала, г. — Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. — URL: <http://moodle.dgu.ru/> (дата обращения: 22.03.2021).

3. Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах литературы, поступающих в фонд НБ ДГУ / Дагестанский гос. ун-т. — Махачкала, 2010 — режим доступа: <http://elib.dgu.ru>, свободный (дата обращения: 21.03.2021).

4. Сайт о программировании METANIT.COM — режим доступа: <https://metanit.com/sharp/>, свободный (дата обращения: 21.03.2021).

5. Сайт о программировании ProfessorWeb.ru – режим доступа: https://professorweb.ru/my/csharp/charp_theory/level1/index1.php, свободный (дата обращения: 21.03.2021).

10. Методические указания для обучающихся по освоению дисциплины.

Для эффективного усвоения программного материала по дисциплине «Разработка приложений на языке Си», как и по любой другой дисциплине, предусмотрены разнообразные формы аудиторной и внеаудиторной самостоятельной работы студентов, в том числе:

- прослушивание лекционного курса в аудитории с написанием конспекта;
- выполнение самостоятельных работ с использованием рекомендованной литературы и Интернет-ресурсов;
- подготовка рефератов, участие на студенческих научно-практических конференциях с докладами по тематике дисциплины;
- выполнение домашних контрольных работ.

В процессе подготовки к семинару студенты могут воспользоваться консультациями преподавателя. Примерные темы докладов, сообщений, вопросов для обсуждения приведены в настоящих рекомендациях. Кроме указанных в настоящих учебно-методических материалах тем, студенты могут по согласованию с преподавателем избирать и другие темы.

Самостоятельная работа необходима студентам для подготовки к семинарским занятиям и подготовки рефератов на выбранную тему с использованием материалов преподаваемого курса, лекций и рекомендованной литературы.

Самостоятельная работа включает глубокое изучение монографий, научных статей и работ, учебных пособий по данной дисциплине.

Самостоятельную работу над дисциплиной следует начинать с изучения программы, которая содержит основные требования к знаниям, умениям, навыкам обучаемых. Обязательно следует выполнять рекомендации преподавателя, данные в ходе установочных занятий. Затем приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью учебника. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела, включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно. Кроме того, формой самостоятельной работы студента является подготовка реферата.

Примерная тематика рефератов приведена в настоящем пособии. Студент может выбрать и другую тему, согласовав ее с преподавателем.

Реферат - это научно-исследовательская работа студента, и которой он делает анализ источников права и изученной литературы по выбранной теме.

Реферат является отражением знания студента выбранной темы. Работа проводится под руководством преподавателя: согласовывается план работы, определяются источники и литература, обсуждаются возможные методы исследования вопросов выбранной темы. Объем реферата 15-20 страниц печатного текста (размер шрифта (кегель) - 14, междустрочный интервал полусторонний).

На титульном листе указываются: принадлежность к ФГБОУ ДГУ; тема реферата; фамилия, имя, отчество автора, курс, учебная группа, год написания. На втором листе помещается план реферата, включающий введение, основные вопросы, заключение.

Изложение материала в письменной работе (реферат) делится на три логические части: введение, основная часть, заключение. Введение содержит обоснование темы. Основная часть также делится на логически завершенные части исследуемой темы (разделы, главы, параграфы). Заключение должно содержать выводы, к которым пришел автор. В работе должны быть сноски на цитируемую литературу, малоизвестные факты, статистические данные. В конце работы приводится список использованной литературы в алфавитном порядке (фамилия, инициалы автора; заголовок использованного учебника, монографии, статьи и т.п.; издательство, год издания, страницы).

Оценка учебной деятельности студентов проводится по модульно-рейтинговой системе, которая включает в себя следующие формы контроля: текущий, промежуточный и итоговый. Результаты всех видов учебной деятельности оцениваются рейтинговыми баллами. Максимальное количество баллов по результатам текущей работы и промежуточного контроля по дисциплинарному модулю составляет 100 баллов.

Промежуточный контроль включает в себя контрольную работу в традиционной письменной форме и компьютерное тестирование (40 баллов).

Итоговый контроль – это проведение итогов текущей работы и промежуточных контролей по дисциплинарным модулям, которая оценивается 100 баллов. Формой итогового контроля может быть письменная контрольная работа или компьютерное тестирование.

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

систем.

1. Операционная система Windows7.
2. Пакет офисных программ Microsoft Office 2013.
3. Среда программирования Microsoft Visual Studio.NET.

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

Лекционные занятия по данной дисциплине проводятся в мультимедийном зале, где установлен проектор и экран.

Практические занятия проводятся в двух компьютерных классах где установлены по 15 компьютеров, все они подключены локальной сети университета т.е. имеют доступ к локальным ресурсам ДГУ и глобальной сети Интернет.