

Документ подписан простой электронной подписью  
Информация о владельце:

ФИО: Исаев Игорь Магомедович

Должность: Проректор по учебной работе

Дата подписания: 25.08.2023 14:55:17

Уникальный идентификатор документа:

d7a26b9e8ca85e98ec3de2eb454b4659d061f249

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Национальный исследовательский технологический университет «МИСИС»

## Рабочая программа дисциплины (модуля)

# Объектно-ориентированное программирование

Закреплена за подразделением

Кафедра инженерной кибернетики

Направление подготовки

01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА

Профиль

Алгоритмы и методы наукоемкого программного обеспечения

Квалификация **Бакалавр**

Форма обучения **очная**

Общая трудоемкость **4 ЗЕТ**

Часов по учебному плану 144

Формы контроля в семестрах:

в том числе:

экзамен 2

аудиторные занятия 51

самостоятельная работа 51

часов на контроль 42

### Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		Итого	
	Неделя 18			
Вид занятий	УП	РП	УП	РП
Лекции	17	17	17	17
Лабораторные	34	34	34	34
Итого ауд.	51	51	51	51
Контактная работа	51	51	51	51
Сам. работа	51	51	51	51
Часы на контроль	42	42	42	42
Итого	144	144	144	144

Программу составил(и):

*ст.преп., Куренкова Татьяна Васильевна*

Рабочая программа

**Объектно-ориентированное программирование**

Разработана в соответствии с ОС ВО:

Самостоятельно устанавливаемый образовательный стандарт высшего образования - бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС» по направлению подготовки 01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА (приказ от 02.04.2021 г. № 119 о.в.)

Составлена на основании учебного плана:

01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА, 01.03.04-БПМ-23.plx Алгоритмы и методы наукоемкого программного обеспечения, утвержденного Ученым советом НИТУ МИСИС в составе соответствующей ОПОП ВО 22.06.2023, протокол № 5-23

Утверждена в составе ОПОП ВО:

01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА, Алгоритмы и методы наукоемкого программного обеспечения, утвержденной Ученым советом НИТУ МИСИС 22.06.2023, протокол № 5-23

Рабочая программа одобрена на заседании

**Кафедра инженерной кибернетики**

Протокол от 23.06.2021 г., №11

Руководитель подразделения Ефимов Альберт Рувимович

### 1. ЦЕЛИ ОСВОЕНИЯ

1.1	Цель – познакомить студентов с объектно-ориентированной моделью программирования, предоставляемой общезыковой исполняющей средой (CLR) платформы .NET Framework. Научить создавать пользовательские типы и использовать типы библиотеки классов платформы .NET Framework при разработке приложений на языке C# для решения прикладных задач в своей учебной и профессиональной деятельности.
-----	--

### 2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

	Блок ОП:	Б1.О
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>	
2.1.1	Вычислительные машины, сети и системы	
2.1.2	Программирование и алгоритмизация	
<b>2.2</b>	<b>Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>	
2.2.1	Базы данных	
2.2.2	Комбинаторика и теория графов	
2.2.3	Технологии программирования	
2.2.4	Алгоритмы дискретной математики	
2.2.5	Операционные системы и среды	
2.2.6	Разработка клиент-серверных приложений	
2.2.7	Сетевые технологии	
2.2.8	Учебная практика по ознакомлению с технологиями разработки наукоемкого ПО	
2.2.9	Учебная практика по ознакомлению с технологиями разработки робототехнических и киберфизических систем	
2.2.10	Математическое моделирование	
2.2.11	Основы теории информации и автоматов	
2.2.12	Основы электротехники и электроники	
2.2.13	Современные технологии разработки мобильных приложений	
2.2.14	Теория систем автоматического управления	
2.2.15	Теория случайных процессов	
2.2.16	Функциональный анализ	
2.2.17	Численные методы	
2.2.18	Дискретные и нелинейные системы автоматического управления	
2.2.19	Имитационное моделирование	
2.2.20	Машинное обучение II	
2.2.21	Методы и средства обработки изображений	
2.2.22	Методы оптимизации	
2.2.23	Научно-исследовательская работа	
2.2.24	Научно-исследовательская работа	
2.2.25	Прикладной статистический анализ	
2.2.26	Программирование роботов I	
2.2.27	Производственная практика по освоению первичных навыков в области разработки наукоемкого ПО	
2.2.28	Производственная практика по освоению первичных навыков в области разработки робототехнических и киберфизических систем	
2.2.29	Фрактальный анализ	
2.2.30	Введение в разработку приложений дополненной и виртуальной реальностей	
2.2.31	Нейронные сети	
2.2.32	Облачные технологии	
2.2.33	Обработка естественного языка	
2.2.34	Обучение с подкреплением	
2.2.35	Программирование роботов II	
2.2.36	Системный анализ и принятие решений	
2.2.37	Системы автоматизированного проектирования	
2.2.38	Экспертные и рекомендательные системы	
2.2.39	Глубокое обучение	
2.2.40	Искусственный интеллект и мультиагентные системы	

2.2.41	Киберфизические системы
2.2.42	Параллельные вычисления
2.2.43	Подготовка к процедуре защиты и защита выпускной квалификационной работы
2.2.44	Подготовка к процедуре защиты и защита выпускной квалификационной работы
2.2.45	Преддипломная практика для апробации темы выпускной квалификационной работы
2.2.46	Преддипломная практика для апробации темы выпускной квалификационной работы
2.2.47	Системы обеспечения информационной безопасности и блокчейн
2.2.48	Современные инструменты DevOps
2.2.49	Специальные главы баз данных
2.2.50	Специальные главы математики для Computer Science

### 3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ФОРМИРУЕМЫМИ КОМПЕТЕНЦИЯМИ

**ПК-3: Способен осуществлять подготовку элементов документации, проектов планов и программ проведения отдельных этапов работ**

**Знать:**

ПК-3-31 правила подготовки элементов документации и программ проведения отдельных этапов разработки программного обеспечения.

Знать правила создания документации для кода путем включения XML-элементов в специальные поля комментариев.

Знать современные технологии командной разработки программного обеспечения.

**ОПК-4: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности, разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения, выбирать и применять методики проектирования и актуальные инструментальные средства разработки**

**Знать:**

ОПК-4-31 Знать архитектуру и основные функции платформы .NET Framework.

Знать основные функциональные возможности среды выполнения CLR.

Знать правила предоставляемые общей системой типов (CTS) для определения следующих типов: класс, структура, интерфейс, делегат, перечисление, массив, и следующих членов типа: константа, поле, метод, конструктор, свойство, событие, индексатор и другие.

Понимать различие между типом значения и ссылочным типом.

Знать правила работы с библиотекой классов .NET Framework.

Знать функциональные возможности инструментальных средств среды разработки Microsoft Visual Studio

**ОПК-1: Способен применять знание фундаментальной математики и естественно-научных дисциплин при решении задач в области естественных наук и инженерной практике**

**Знать:**

ОПК-1-31 Знать базовые концепции и терминологию объектно-ориентированного программирования: абстрагирование, класс, объект, инкапсуляция, наследование, полиморфизм.

Понимать, как использовать объектно-ориентированную модель программирования при разработке и проектировании пользовательских типов.

**ПК-3: Способен осуществлять подготовку элементов документации, проектов планов и программ проведения отдельных этапов работ**

**Уметь:**

ПК-3-У1 Использовать справочную систему <https://docs.microsoft.com/ru-ru/dotnet/>, спецификацию языка.

Находить и использовать литературу и информацию, предоставленную в открытом доступе: интернет форумы, статьи специалистов.

Использовать XML-документацию, для организации документооборота между группами разработчиков и специалистов по тестированию.

**ОПК-4: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности, разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения, выбирать и применять методики проектирования и актуальные инструментальные средства разработки**

**Уметь:**

ОПК-4-У1 Определять классы. Создавать объекты в программе на языке C#. Использовать статические конструкторы и конструкторы экземпляров.

Определять структуры, перечисления, делегаты, события.

Определять свойства, индексаторы.

Определять и вызывать методы. Определять методы операторов преобразования. Определять методы перегруженных операторов..

Определять интерфейсы. Выполнять реализацию для всех членов, определяемых интерфейсом в производных классах

(структурах). Определять абстрактные и запечатанные классы и члены классов  
Создавать пользовательские универсальные типы, итераторы. Использовать LINQ.  
Использовать типы предоставляемые библиотекой классов .NET Framework для работы со встроенными типами, коллекциями и универсальными шаблонами, файлами, строками и другие при разработке приложений для решения прикладных задач.  
Работать в среде разработки Microsoft Visual Studio. Создавать, отлаживать, компилировать и выполнять программы на языке C#.

**ОПК-1: Способен применять знание фундаментальной математики и естественно-научных дисциплин при решении задач в области естественных наук и инженерной практике**

**Уметь:**

ОПК-1-У1 Применять типовые алгоритмы и структуры данных для решения прикладных задач.  
Исследовать и проводить анализ задачи, выделять сущности (классы) в данной предметной области.  
Применять знание фундаментальной математики и естественно-научных дисциплин при решении задач в области естественных наук и инженерной практике.

**ПК-3: Способен осуществлять подготовку элементов документации, проектов планов и программ проведения отдельных этапов работ**

**Владеть:**

ПК-3-В1 Владеть навыками индивидуальной и командной работы. Демонстрировать личную ответственность, готовность к самосовершенствованию и повышению своей квалификации.

**ОПК-4: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности, разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения, выбирать и применять методики проектирования и актуальные инструментальные средства разработки**

**Владеть:**

ОПК-4-В1 Владеть навыками разработки программного обеспечения с использованием платформы разработки .NET Framework на языке C#.  
Владеть навыками построения, отладки, тестирования и документирования программного обеспечения.  
Владеть навыками разработки алгоритмов для решения прикладных задач. Рассчитывать оценку их трудоемкости и эффективности использования.

**ОПК-1: Способен применять знание фундаментальной математики и естественно-научных дисциплин при решении задач в области естественных наук и инженерной практике**

**Владеть:**

ОПК-1-В1 Владеть навыками разработки программ с использованием объектно-ориентированного подхода.  
Владеть навыками использования методов математики и естественно-научных дисциплин при разработке алгоритмов для решения прикладных задач.

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Формируемые индикаторы компетенций	Литература и эл. ресурсы	Примечание	КМ	Выполняемые работы
	Раздел 1. Программирование на платформе .NET Framework. Проектирование и разработка типов на языке C# в соответствии со стандартами принятыми в спецификации CTS (общая система типов).							

1.1	<p>Разработка пользовательских классов и структур.</p> <p>Объявление классов и структур. Члены классов и структур: поля, методы, свойства, статические и экземплярные.</p> <p>Использование модификаторов public или private при определении полей, методов и свойств.</p> <p>Инициализация открытых полей в классах и в структурах. Конструкторы экземпляров в классах и в структурах. Закрытые поля в классах и в структурах переопределение метода ToString(). Перегрузка операторов. Методы операторов преобразования /Лаб/</p>	2	4	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1</p>	Л1.1Л2.1 Э1			P1
1.2	<p>Разработка пользовательских классов и структур.</p> <p>Объявление классов и структур. Определение в классе (структуре) поля, которое является массивом и имеет модификатор доступа private.</p> <p>Индексаторы обращение к экземпляру класса по индексу. Определение в классе (структуре) поля, которое является массивом и имеет модификатор доступа private.</p> <p>Использование итераторов. /Лаб/</p>	2	4	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1</p>	Л1.1Л2.1 Э1			P2
1.3	<p>Наследование и полиморфизм.</p> <p>Базовые и производные классы. Видимость типов. Конструкторы производных классов. Использование base. Доступ производного класса к членам базового класса. Виртуальные методы. Переопределение методов базового класса. Переопределение не виртуальных методов (сокрытие). /Лаб/</p>	2	2	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1</p>	Л1.1Л2.1 Э1			P3

1.4	<p>Наследование и полиморфизм.</p> <p>Интерфейсы. Определение интерфейса. Модификаторы интерфейса. Члены интерфейса. Наследование от интерфейсов. Реализация для всех членов, определяемых интерфейсом в производных классах (структурах). Явная реализация членов интерфейса. Абстрактные классы. Запечатанные классы. /Лаб/</p>	2	6	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1</p>	Л1.1Л2.1 Э1		КМ1	Р4
1.5	<p>Разработка программ с использованием делегатов.</p> <p>1. Объявление делегата. Создание нескольких экземпляров делегата. Объединение экземпляров делегата. Использование делегата для передачи метода в качестве аргумента в другой метод. Разработка событий.</p> <p>2. Анонимные функции. Примеры создания анонимных методов и лямбда выражений (материал в лекциях). Оператор =&gt;. Создание типа дерева выражений. Выполнение дерева выражений. LINQ (Language-Integrated Query) . Основные операции запросов LINQ. Использование лямбда-выражений в запросах. Методы расширения стандартных операторов запросов. /Лаб/</p>	2	4	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1</p>	Л1.1Л2.1 Э1			Р5

1.6	<p>Создание пользовательских универсальных типов</p> <p>Обобщенные решения и шаблоны разработки.</p> <p>Создание пользовательских универсальных типов</p> <p>Объявление универсального типа.</p> <p>Параметры типа и аргументы типа.</p> <p>Ограничения параметров типа. Равенство ссылок. Равенство значений. Метод <code>Object.ReferenceEquals</code></p> <p>Переопределение метода <code>Equals</code></p> <p>Универсальные методы.</p> <p>Универсальные интерфейсы.</p> <p>Ковариантность и контрвариантность в универсальных шаблонах.</p> <p>Универсальные интерфейсы с ковариантными параметрами типа.</p> <p>Универсальные интерфейсы с контрвариантными параметрами типа.</p> <p>Универсальные делегаты с параметрами вариантного типа. Универсальные типы и отражение. Пространство имен <code>System.Reflection</code>. /Лаб/</p>	2	4	<p>ОПК-1-31</p> <p>ОПК-1-У1</p> <p>ОПК-1-В1</p> <p>ОПК-4-31</p> <p>ОПК-4-У1</p> <p>ОПК-4-В1</p>	Л1.1Л2.1 Э1		КМ2	Р6
1.7	<p>Типы перечислений.</p> <p>Объявление перечисления.</p> <p>Модификаторы перечисления. Тип <code>System.Enum</code>. Создание экземпляра перечислимого типа. Примеры работы с перечислениями.</p> <p>Использование интерфейса <code>IFormattable</code>.</p> <p>/Лаб/</p>	2	2	<p>ОПК-1-31</p> <p>ОПК-1-У1</p> <p>ОПК-1-В1</p> <p>ОПК-4-31</p> <p>ОПК-4-У1</p> <p>ОПК-4-В1</p>	Л1.1Л2.1 Э1			Р7



1.8	<p>Платформа .NET Framework: общезыковая среда выполнения CLR, библиотека классов .NET Framework. Общие сведения о системе общих типов (CTS). Типы значений и ссылочные типы. Встроенные типы. Литералы. Операторы языка. Области объявления. Области видимости. Переменные. Локальные переменные. Приведение и преобразование типов. Неявные преобразования. Преобразования упаковки. Явные преобразования. Операторы checked и unchecked. Преобразования распаковки. Выражения. Основные выражения. Простые имена. Выражения со скобками. Доступ к члену. Доступ this. Оператор new. Выражения создания объектов. Оператор typeof. Унарные операторы. Бинарные операторы. Ненулевые конечные значения, нули, бесконечные значения и ошибки NaN. Массивы. Массив массивов. Тип System.Array</p> <p>Объявление пользовательского типа. Атрибуты, модификаторы, ключевое слово, определяющее тип, идентификатор, список параметров типа, базовый тип данного типа, ограничения параметров типа, тело типа. Члены типа. Видимость типа и модификаторы доступа к члену типа. Ключевые слова, которые влияют на определение типа или члена типа: abstract, virtual, override, sealed, new /Лек/</p>	2	2	ОПК-1-31 ОПК-4-31	Л1.Л2.1 Э1			
-----	---	---	---	----------------------	---------------	--	--	--

1.9	<p>Инкапсуляция. Классы. Объявление класса. Модификаторы класса. Абстрактные классы. Запечатанные классы. Статические классы. Тело класса. Модификаторы доступа для класса и членов класса. Объявление членов класса. Создание объектов. Оператор new. Конструкторы экземпляров. Инициализаторы конструкторов. Использование ключевого слова this, base. Выполнение конструктора. Конструкторы по умолчанию. Закрытые конструкторы. Статические конструкторы. Создание конструктора копии. Константы. Поля. Объявление поля. Инициализация поля. Модификаторы полей. Ключевые слова readonly, volatile. Статические и экземплярные поля. Использование модификаторов static readonly Доступ к полю. Методы. Объявление метода. Модификаторы метода. Возвращаемое значение. Имя метода. Параметры метода. Массив параметров Статические и экземплярные методы. Виртуальные методы. Переопределяющие методы. Запечатанные методы Абстрактные методы. Тело метода. Пользовательские преобразования. Методы перегруженных операторов. /Лек/</p>	2	2	ОПК-1-31 ОПК-4-31	Л1.Л2.1 Э1			
-----	---	---	---	----------------------	---------------	--	--	--

1.10	<p>Свойства: Объявление свойства. Модификаторы свойства. Методы доступа свойства get и set. Ключевые слова value и return. Статические и экземплярные свойства. Доступ к свойству. Автоматически реализуемые свойства. Индексаторы: Объявление индексатора. Модификаторы индексатора. Методы доступа get и set. Ключевые слова value и return, this.</p> <p>Наследование и полиморфизм. Базовые и производные классы. Видимость типов. Конструкторы производных классов. Использование base. Доступ производного класса к членам базового класса. Виртуальные методы. Переопределение методов базового класса. Переопределение не виртуальных методов (сокрытие). /Лек/</p>	2	2	ОПК-1-31 ОПК-4-31	Л1.1Л2.1 Э1			
1.11	<p>Перечисления. Интерфейсы. Определение интерфейса. Модификаторы интерфейса. Члены интерфейса. Наследование от интерфейсов. Реализация для всех членов, определяемых интерфейсом в производных классах (структурах). Явная реализация членов интерфейса.</p> <p>Делегаты. Объявление делегата. Модификаторы делегата. Создание экземпляра делегата События. Подписка и отмена подписки на событие. Анонимные функции: лямбда выражения и выражения анонимных методов. /Лек/</p>	2	4	ОПК-1-31 ОПК-4-31	Л1.1Л2.1 Э1			

1.12	<p>Обобщенные решения и шаблоны разработки. Создание пользовательских универсальных типов (универсальные шаблоны). Объявление универсального типа. Параметры типа и аргументы типа. Ограничения параметров типа. Универсальные методы. Универсальные интерфейсы. Равенство ссылок. Равенство значений. Ковариантность и контрвариантность в универсальных шаблонах. Универсальные интерфейсы с ковариантными параметрами типа. Универсальные интерфейсы с контрвариантными параметрами типа. Универсальные делегаты с параметрами вариантного типа. Универсальные типы и отражение. /Лек/</p>	2	3	<p>ОПК-1-31 ОПК-4-31</p>	<p>Л1.1Л2.1 Э1</p>			
1.13	<p>Проработка лекционного материала в электронной системе обучения Canvas Самостоятельное изучение литературы. Подготовка к ЛР Программная реализация задач для самостоятельной работы. /Ср/</p>	2	33	<p>ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1 ПК-3-У1</p>	<p>Л1.1Л2.1 Э1</p>			
	<p><b>Раздел 2. Использование типов, предоставляемых библиотекой классов .NET Framework для работы со встроенными типами, строками, коллекциями, файлами и другими</b></p>							

2.1	<p>Абстрактные типы данных и структуры данных. Типы, предоставляемые библиотекой классов .NET Framework для работы с коллекциями и универсальными шаблонами. Пространства имен System.Collections. System.Collections.Generic. Классы ArrayList, Queue, Stack, Hashtable. Интерфейсы IComparer, IEnumerator, IEnumerable, IList.</p> <p>Итераторы. Универсальные типы и отражение. Пространство имен System.Reflection, класс System.Type.</p> <p>Типы для работы с символами, строками. Кодирование и декодирование потоков символов, строк. Класс Encoding пространства имен System.Text. Форматы и региональные стандарты. Форматирование. Интерфейсы IFormattable, IFormatProvider. Разбор строк. Получение строкового представления объекта</p> <p>Обработка исключений. Классы исключений, определенные в библиотеке классов .NET Framework. Пространство имен System класс Exception</p> <p>Создание документации для кода путем включения XML-элементов в специальные поля комментариев.</p> <p>Современные технологии командной разработки программного обеспечения. /Лек/</p>	2	4	ОПК-1-31 ОПК-4-31 ПК-3-31	Л1.1Л2.1 Э1			
2.2	<p>Коллекции, универсальные коллекции. Итераторы. Интерфейсы IComparer, IEnumerator, IEnumerable, IList</p> <p>Пространства имен System.Collections. System.Collections.Generic. Классы ArrayList, Queue, Stack, Hashtable.</p> <p>Работа с файлами и строками. Интерфейсы IFormattable, IFormatProvider.</p> <p>Обработка исключений /Лаб/</p>	2	8	ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1 ПК-3-31 ПК-3-У1 ПК-3-В1	Л1.1Л2.1 Э1		КМЗ	Р8

2.3	Проработка лекционного материала в электронной системе обучения Canvas Самостоятельное изучение литературы. Подготовка к ЛР Программная реализация задач для самостоятельной работы. /Ср/	2	18	ОПК-1-31 ОПК-1-У1 ОПК-1-В1 ОПК-4-31 ОПК-4-У1 ОПК-4-В1 ПК-3-31 ПК-3-У1 ПК-3-В1	Л1.1Л2.1 Э1			
-----	---	---	----	---	----------------	--	--	--

### 5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

#### 5.1. Контрольные мероприятия (контрольная работа, тест, коллоквиум, экзамен и т.п), вопросы для самостоятельной подготовки

Код КМ	Контрольное мероприятие	Проверяемые индикаторы компетенций	Вопросы для подготовки
КМ1	Контрольная работа 1 Разработка пользовательских классов и структур	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	<p>Отличие переменных типа значений от переменных ссылочного типа. Перечислите ссылочные и значимые типы, которые может определить пользователь.</p> <p>Какие сведения может включать объявление типа</p> <p>Видимость типа и модификаторы доступа к члену типа</p> <p>Какой тип является базовым для всех типов</p> <p>Создание объекта. Привести пример.</p> <p>Статические члены и члены экземпляра.</p> <p>Ключевые слова, которые влияют на определение типа или члена типа <code>abstract</code>, <code>virtual</code>, <code>override</code>, <code>sealed</code>, <code>new</code></p> <p>Инкапсуляция. Что такое класс? Чем класс отличается от структуры? Чем класс отличается от объекта?</p> <p>Объявление классов и структур. Привести пример использования.</p> <p>Модификаторы классов и структур.</p> <p>Абстрактные классы.</p> <p>Запечатанные классы.</p> <p>Статические классы</p> <p>Члены классов и структур: поля, методы, свойства. Дайте определение, приведите пример создания. Доступ к полю, к свойству, вызов метода. Модификаторы поля, метода, свойства.</p> <p>Инициализация открытых полей в классах и в структурах.</p> <p>Конструкторы экземпляров в классах и в структурах.</p> <p>Конструкторы экземпляров. Инициализаторы конструкторов.Использование ключевого слова <code>this</code>, <code>base</code>.</p> <p>Выполнение конструктора. Конструкторы по умолчанию. Закрытые конструкторы.</p> <p>Статические конструкторы. Создание конструктора копии.</p> <p>Индексаторы: Объявление индексатора. Модификаторы индексатора. Методы доступа <code>get</code> и <code>set</code>. Привести пример использования. Ключевые слова <code>value</code> и <code>return</code>, <code>this</code>.</p>

КМ2	Контрольная работа 2 Наследование и полиморфизм	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	<p>Базовые и производные классы. Видимость типов. Конструкторы производных классов. Привести пример использования</p> <p>Использование base. Доступ производного класса к членам базового класса. Привести пример использования.</p> <p>Виртуальные методы. Переопределяющие методы. Запечатанные методы. Абстрактные методы.</p> <p>Для чего используются виртуальные и абстрактные методы? Привести пример их использования на примере двух классов базового и производного.</p> <p>Переопределение методов базового класса. Переопределение не виртуальных методов (сокрытие). Привести пример использования.</p> <p>Интерфейсы. Определение интерфейса.</p> <p>Какие модификаторы может содержать интерфейс?</p> <p>Какие члены может содержать интерфейс?</p> <p>Какие модификаторы могут содержать члены интерфейса?</p> <p>Можно ли создать экземпляр интерфейса?</p> <p>Наследование от интерфейсов.</p> <p>Реализация для всех членов, определяемых интерфейсом в производных классах (структурах).</p> <p>Явная реализация членов интерфейса.</p>
КМ3	Контрольная работа 1 Разработка программ с использованием делегатов	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	<p>Объявление делегата.</p> <p>Модификаторы делегата.</p> <p>Создание экземпляра делегата</p> <p>События. Подписка и отмена подписки на событие.</p> <p>Анонимные функции: лямбда выражения и выражения анонимных методов</p> <p>Использование лямбда выражения для создания делегатов и создания типов дерева выражений.</p> <p>Делегаты Action, Func.</p> <p>Создание типа дерева выражений с использованием класса Expression.</p> <p>Использование лямбда-выражений в запросах. LINQ.</p> <p>Методы расширения стандартных операторов запросов</p>

КМ4	Экзамен	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	<p>Отличие переменных типа значений от переменных ссылочного типа. Перечислите ссылочные и значимые типы, которые может определить пользователь.</p> <p>Какие сведения может включать объявление типа.</p> <p>Видимость типа и модификаторы доступа к члену типа</p> <p>Какой тип является базовым для всех типов</p> <p>Создание объекта. Привести пример.</p> <p>Статические члены и члены экземпляра. Привести пример использования.</p> <p>Ключевые слова, которые влияют на определение типа или члена типа <code>abstract</code>, <code>virtual</code>, <code>override</code>, <code>sealed</code>, <code>new</code></p> <p>Инкапсуляция. Что такое класс? Чем класс отличается от структуры? Чем класс отличается от объекта?</p> <p>Объявление классов и структур. Привести пример использования.</p> <p>Модификаторы классов и структур.</p> <p>Абстрактные классы. Привести пример использования</p> <p>Запечатанные классы. Привести пример использования</p> <p>Статические классы. Привести пример использования</p> <p>Члены классов и структур: поля, методы, свойства. Дайте определение, приведите пример создания. Доступ к полю, к свойству, вызов метода. Модификаторы поля, метода, свойства. Привести пример использования.</p> <p>Инициализация открытых полей в классах и в структурах. Привести пример использования</p> <p>Конструкторы экземпляров в классах и в структурах. Привести пример</p> <p>Конструкторы экземпляров. Инициализаторы конструкторов.</p> <p>Использование ключевого слова <code>this</code>, <code>base</code>. Привести пример</p> <p>Выполнение конструктора. Конструкторы по умолчанию. Закрытые конструкторы. Привести пример</p> <p>Статические конструкторы. Создание конструктора копии.</p> <p>Индексаторы: Объявление индексатора. Модификаторы индексатора. Методы доступа <code>get</code> и <code>set</code>. Привести пример использования. Ключевые слова <code>return</code>.</p> <p>Какие операторы нельзя перегрузить?</p> <p>Можно ли перегружать стандартные операторы?</p> <p>Методы перегруженных операторов. Дайте определение, приведите пример создания.</p> <p>Что должен определять тип чтобы выполнить пользовательское преобразование экземпляра типа, в котором определен метод оператора преобразования, в экземпляр другого типа?</p> <p>Базовые и производные классы. Видимость типов. Конструкторы производных классов. Привести пример использования</p> <p>Использование <code>base</code>. Доступ производного класса к членам базового класса. Привести пример использования.</p> <p>Виртуальные методы. Переопределяющие методы. Запечатанные методы. Абстрактные методы.</p> <p>Для чего используются виртуальные и абстрактные методы?</p> <p>Привести пример их использования на примере двух классов базового и производного.</p> <p>Переопределение методов базового класса. Переопределение не виртуальных методов (сокрытие). Привести пример использования.</p> <p>Как создать экземпляр перечислимого типа?</p> <p>Как получить доступ к статическим членам перечисления?</p> <p>Интерфейсы. Определение интерфейса.</p> <p>Какие модификаторы может содержать интерфейс?</p> <p>Какие члены может содержать интерфейс?</p> <p>Какие модификаторы могут содержать члены интерфейса?</p> <p>Можно ли создать экземпляр интерфейса?</p> <p>Наследование от интерфейсов.</p> <p>Реализация для всех членов, определяемых интерфейсом в производных классах (структурах).</p> <p>Явная реализация членов интерфейса.</p> <p>Объявление делегата.</p> <p>Модификаторы делегата.</p> <p>Создание экземпляра делегата</p> <p>События. Подписка и отмена подписки на событие.</p> <p>Анонимные функции: лямбда выражения и выражения анонимных</p>
-----	---------	---	---



			<p>методов</p> <p>Использование лямбда выражения для создания делегатов и создания типов дерева выражений.</p> <p>Типы, предоставляемые библиотекой классов .NET Framework для работы с коллекциями и универсальными шаблонами.</p> <p>Пространства имен System.Collections. System.Collections.Generic. Класс ArrayList.</p> <p>Интерфейсы IComparer, IEnumerator, IEnumerable, IList. Итераторы.</p> <p>Универсальные типы и отражение. Пространство имен System.Reflection, класс System.Type.</p> <p>Типы для работы с символами, строками. Кодирование и декодирование потоков символов, строк. Класс Encoding пространства имен System.Text. Форматы и региональные стандарты. Форматирование. Интерфейсы IFormattable, IFormatProvider. Разбор строк. Получение строкового представления объекта</p> <p>Обработка исключений. Классы исключений, определенные в библиотеке классов .NET Framework. Пространство имен System класс Exception</p> <p>Создание документации для кода путем включения XML-элементов в специальные поля комментариев.</p> <p>Современные технологии командной разработки программного обеспечения.</p>
--	--	--	--

**5.2. Перечень работ, выполняемых по дисциплине (Курсовая работа, Курсовой проект, РГР, Реферат, ЛР, ПР и т.п.)**

Код работы	Название работы	Проверяемые индикаторы компетенций	Содержание работы
P1	ЛР Разработка пользовательских классов и структур	ОПК-4-У1;ОПК-4-В1;ОПК-1-У1;ОПК-1-В1;ОПК-4-31;ОПК-1-31	Объявление классов и структур. Члены классов и структур: поля, методы, свойства, статические и экземплярные. Использование модификаторов public или private при определении полей, методов и свойств. Инициализация открытых полей в классах и в структурах. Конструкторы экземпляров в классах и в структурах. Закрытые поля в классах и в структурах переопределение метода ToString().
P2	ЛР Разработка пользовательских классов и структур	ОПК-4-У1;ОПК-4-В1;ОПК-1-У1;ОПК-1-В1;ОПК-4-31;ОПК-1-31	Объявление классов и структур. Определение в классе (структуре) поля, которое является массивом и имеет модификатор доступа private. Индексаторы обращение к экземпляру класса по индексу. Определение в классе (структуре) поля, которое является массивом и имеет модификатор доступа private. Использование итераторов.
P3	ЛР Наследование и полиморфизм.	ОПК-4-У1;ОПК-4-В1;ОПК-1-У1;ОПК-1-В1;ОПК-4-31;ОПК-1-31	Базовые и производные классы. Видимость типов. Конструкторы производных классов. Использование base. Доступ производного класса к членам базового класса. Виртуальные методы. Переопределение методов базового класса. Переопределение не виртуальных методов (сокрытие).
P4	ЛР Наследование и полиморфизм.	ОПК-4-У1;ОПК-4-В1;ОПК-1-У1;ОПК-1-В1;ОПК-4-31;ОПК-1-31	Интерфейсы. Определение интерфейса. Модификаторы интерфейса. Члены интерфейса. Наследование от интерфейсов. Реализация для всех членов, определяемых интерфейсом в производных классах (структурах). Явная реализация членов интерфейса. Абстрактные классы. Запечатанные классы.
P5	ЛР Разработка программ с использованием делегатов.	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	<p>1. Объявление делегата. Создание нескольких экземпляров делегата. Объединение экземпляров делегата. Использование делегата для передачи метода в качестве аргумента в другой метод. Разработка событий.</p> <p>2. Анонимные функции. Примеры создания анонимных методов и лямбда выражений (материал в лекциях). Оператор =&gt;. Создание типа дерева выражений. Выполнение дерева выражений. LINQ (Language-Integrated Query) . Основные операции запросов LINQ. Использование лямбда-выражений в запросах. Методы расширения стандартных операторов запросов.</p>

P6	ЛР Создание пользовательских универсальных типов	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	Обобщенные решения и шаблоны разработки. Создание пользовательских универсальных типов Объявление универсального типа. Параметры типа и аргументы типа. Ограничения параметров типа. Равенство ссылок. Равенство значений. Метод Object.ReferenceEquals Переопределение метода Equals  Универсальные методы. Универсальные интерфейсы. Ковариантность и контрвариантность в универсальных шаблонах. Универсальные интерфейсы с ковариантными параметрами типа. Универсальные интерфейсы с контрвариантными параметрами типа. Универсальные делегаты с параметрами вариантного типа. Универсальные типы и отражение. Пространство имен System.Reflection.
P7	ЛР Типы перечислений.	ОПК-4-31;ОПК-4-У1;ОПК-4-В1;ОПК-1-31;ОПК-1-У1;ОПК-1-В1	Объявление перечисления. Модификаторы перечисления. Тип System.Enum. Создание экземпляра перечислимого типа. Примеры работы с перечислениями. Использование интерфейса IFormattable.
P8	ЛР Использование типов, предоставляемых библиотекой классов .NET Framework	ОПК-4-В1;ОПК-1-У1;ОПК-1-В1	Коллекции, универсальные коллекции. Итераторы. Интерфейсы IComparer, IEnumerator, IEnumerable, IList Пространства имен System.Collections. System.Collections.Generic. Классы ArrayList, Queue, Stack, Hashtable. Работа с файлами и строками. Интерфейсы IFormattable, IFormatProvider. Обработка исключений. Создание документации для кода путем включения XML-элементов в специальные поля комментариев.

### 5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)

Экзамен состоит из:

экзаменационная письменная работа (разработка программы на компьютере);  
экзаменационный тест (тест проводится с использованием системы LMS Canvas).

На лабораторных занятиях проводится устный опрос и самостоятельная работа за компьютером (разработка программы на компьютере).

Контрольная письменная работа предусматривает разработку программы на компьютере, выполняется на лабораторных занятиях.

«Национальный исследовательский технологический университет «МИСиС»

Кафедра инженерной кибернетики

Дисциплина: «Объектно-ориентированное программирование»

Форма проведения экзамена: письменная и тестирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

I. Тест по теоретическому материалу дисциплины в системе Canvas.

II. Письменная работа. Выполнение задания на компьютере.

Разработать класс `BookList<T>` представляющий строго типизированный список объектов, доступных по индексу (реализация динамического массива). T: Тип элементов в списке.

Пояснение: методы класса `List<T>` библиотеки классов использовать нельзя. Делегаты и интерфейсы библиотеки классов использовать можно.

Класс содержит:

Закрытое поле - массив. Тип элемента массива – T.

Закрытое поле count. Тип поля int. Содержит индекс элемента, по которому добавляется элемент.

Конструктор экземпляра с параметром. Параметр - количество элементов, которое может вместить массив.

Методы:

`public void Add(T item)` Добавляет объект в конец списка (массива).

`public BookList<T> FindAll(System.Predicate<T> match)` Извлекает все элементы, удовлетворяющие условиям указанного предиката. Параметр - делегат `Predicate<T>`, определяющий условия поиска элементов. Возвращаемое значение - список, содержащий все элементы, удовлетворяющие условиям указанного предиката, если такие элементы найдены; в противном случае пустой список.

`public int IndexOf(T value, int startIndex, int count)` Выполняет поиск объекта T value

и возвращает индекс (с нуля) первого вхождения в диапазоне элементов списка BookList, начинающемся с заданного индекса и содержащем указанное число элементов.

`public void Sort(System.Comparison<T> comparison)` Сортирует элементы во всем списке с использованием указанного `Comparison<T>`. `Comparison<T>` - делегат. Представляет метод, сравнивающий два объекта одного типа.

`public void Sort (System.Collections.Generic.IComparer<T> comparer)` Сортирует элементы во всем списке. Реализацию

метода `int Compare(T x, T y)` интерфейса `Comparer<T>`, следует использовать при сравнении элементов.

Свойства:

`Count` - число элементов, содержащихся в списке.

`Capacity` - число элементов, которые может вместить внутренняя структура данных без изменения размера.

Индексатор

Разработать класс `Book`. Использовать тип `Book` в качестве параметра класса `BookList<T>`

Класс содержит

Закрытые поля:

`string title;` // Title of the book.

`string author;` // Author of the book.

`decimal price;` // Price of the book.

`DateTime data;`

Конструктор экземпляра

Для каждого поля определить свойство

Переопределить методы:

`public override string ToString()`

`public override bool Equals(object obj)`

`public override int GetHashCode()`

Перегрузить операции:

`public static bool operator ==(Book d1, Book d2)`

`public static bool operator !=(Book d1, Book d2)`

`public static bool operator <(Book d1, Book d2)`

`public static bool operator >(Book d1, Book d2)`

Протестировать код в методе `static void Main(string[] args)`

I. Тест по теоретическому материалу дисциплины:

1. Что из перечисленного обязательно нужно указывать при создании класса:

- `attributes`
- `class-base`
- `identifier`
- `class-modifiers`

2. При использовании какого ключевого слова в определении типа, члены должны быть реализованы с помощью типа, производного от данного типа:

- `abstract`
- `static`
- `public`
- `sealed`
- `new`

3. В каких пользовательских типах возможно применение модификатора `sealed`:

- Класс
- Структура
- Интерфейс
- Делегат

4. Какие члены не могут быть определены в структурах:

- Поля
- Свойства
- Деструкторы
- События

5. Что из перечисленного является обязательным при объявлении интерфейса:

- Атрибуты
- Модификаторы
- Идентификатор
- Тело интерфейса

6. Какие члены можно определить в статическом классе:

- `public`
- `protected`
- `static`
- `protected internal`
- `private`

1. Какой модификатор доступа предоставляет доступ к типу или члену типа из любого другого кода в той же сборке или другой сборке, ссылающейся на него.

- `public`
- `protected`
- `internal`
- `protected Internal`
- `private`

2. Что из перечисленного не является членом класса

- a. Поля  
 b. Атрибуты  
 c. Методы  
 d. Все перечисленное являются членами класса
3. Где возможно применение модификатора new:  
 a. В определении типа  
 b. В определении члена типа
4. Какой базовый тип не может иметь перечисление:  
 a. char  
 b. byte  
 c. int  
 d. short  
 e. Нет верного ответа
5. Что из перечисленного является обязательным при объявлении делегата:  
 a. Атрибуты  
 b. Модификаторы  
 c. Возвращаемый тип  
 d. Идентификатор
6. Какие модификаторы не могут содержать члены абстрактного класса:  
 a. abstract  
 b. sealed  
 c. static  
 d. virtual  
 e. Все варианты неверны

#### 5.4. Методика оценки освоения дисциплины (модуля, практики. НИР)

Шкала оценивания включает 4 уровня с оценками: отлично, хорошо, удовлетворительно, неудовлетворительно.

Критерии оценивания выполнения лабораторных работ.

Лабораторная работа считается выполненной если:

Даны исчерпывающие и обоснованные ответы на все поставленные вопросы, разработанная на компьютере программа не содержит ошибок, соответствует заданию.

Критерии оценивания контрольных работ (разработка программы на компьютере)

Оценка «неудовлетворительно» – выставляется в случае, если: программа имеет синтаксические ошибки, не соответствует заданию, выполнена не полностью, при выполнении программы возникают исключения.

Оценка «отлично», «хорошо», «удовлетворительно» – зависит от сложности поставленной задачи. Разработанная на компьютере программа не содержит ошибок, соответствует заданию.

Критерии оценивания тестов

Критерии выставления оценок за тест, состоящий 25 вопросов.

Время выполнения работы: 45 мин.

Оценка «отлично» – 90% правильных ответов;

Оценка «хорошо» – 80% правильных ответов;

Оценка «удовлетворительно» – 70% правильных ответов;

Оценка «неудовлетворительно» – ниже 70% правильных ответов.

Экзаменационная письменная работа предусматривает разработку программы на компьютере. Время выполнения работы: 90 мин

Оценка «неудовлетворительно» – выставляется в случае, если: программа имеет синтаксические ошибки, не соответствует заданию, выполнена не полностью, при выполнении программы возникают исключения.

Оценка «отлично», «хорошо», «удовлетворительно» – зависит от сложности поставленной задачи. Разработанная на компьютере программа не содержит ошибок, соответствует заданию.

Если студент получил оценку «неудовлетворительно» за разработанную на компьютере программу, то за экзамен выставляется оценка «неудовлетворительно».

Если студент получил оценку «удовлетворительно» и выше за разработанную на компьютере программу, то оценка выставляется по формуле:

Итоговая оценка = 0.4\*Оценка за тесты + 0.6\* Оценка за программу.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

Авторы, составители	Заглавие	Библиотека	Издательство, год
---------------------	----------	------------	-------------------

	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л1.1	Седжвик Р.	Алгоритмы на C++: курс лекций	Электронная библиотека	Москва: Национальный Открытый Университет «ИНТУИТ», 2016

#### 6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л2.1	Царёв Р. Ю., Прокопенко А. В.	Алгоритмы и структуры данных (CDIO): учебник	Электронная библиотека	Красноярск: Сибирский федеральный университет (СФУ), 2016

#### 6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Э1		<a href="https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/">https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/</a>
----	--	---

#### 6.3 Перечень программного обеспечения

П.1	Microsoft Visual Studio 2015
П.2	LMS Canvas

#### 6.4. Перечень информационных справочных систем и профессиональных баз данных

### 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Ауд.	Назначение	Оснащение
Б-904а	Компьютерный класс	20 стационарных компьютеров (core i5-3470 8gb RAM), пакет лицензионных программ MS Office, демонстрационное оборудование: доска, проектор мультимедийный, экран, колонки, комплект учебной мебели
Б-902	Учебная аудитория	12 стационарных компьютеров (2 x core i5-3470 8gb RAM, 10 x ryzen5 2400g 32gb RAM), пакет лицензионных программ MS Office, демонстрационное оборудование: доска, проектор мультимедийный, комплект учебной мебели на 19 мест
Б-934	Лекционная аудитория	4 кабины для синхронного перевода, мультимедийные экраны и проектор, ноутбук, пакет лицензионных программ MS Office, комплект учебной мебели на 130 посадочных мест
Читальный зал электронных ресурсов		комплект учебной мебели на 55 мест для обучающихся, 50 ПК с доступом к ИТС «Интернет», ЭИОС университета через личный кабинет на платформе LMS Canvas, лицензионные программы MS Office, MS Teams, ESET Antivirus.

### 8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

Освоение дисциплины требует значительного объема самостоятельной работы. Отдельные учебные вопросы выносятся на самостоятельную проработку и контролируются посредством текущей аттестации. При этом организуются групповые и индивидуальные консультации.

Качественное освоение дисциплины возможно только при систематической самостоятельной работе, что поддерживается защитой лабораторных работ, а также системой текущей и рубежной аттестацией.

Самостоятельная работа обучающихся направлена на углубленное изучение тем дисциплины и предполагает изучение основных и дополнительных источников учебной и научной литературы. Полученные знания и навыки в дальнейшем будут использованы освоения других дисциплин по программированию и компьютерным технологиям, выполнении студенческих научных исследований и стать основой для выступления на студенческих научно-практических конференциях, конкурсах студенческих работ, при подготовке ВКР и пр.