

**ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра прикладной математики и теории систем управления



**УТВЕРЖДАЮ:**

проректор по научно-методической  
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**«ВВЕДЕНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ**  
**ПРОГРАММИРОВАНИЕ»**

Направление подготовки:

02.03.02 Фундаментальная информатика  
и информационные технологии

Образовательная программа:

бакалавриат

Квалификация:

Академический бакалавр

Форма обучения:

очная, заочная, в том числе  
с ускоренным сроком обучения

нужное подчеркнуть

Донецк 2020

**УТВЕРЖДАЮ:**

Декан факультета математики  
и информационных технологий

И. А. Моисеенко

«16» апреля 2020

МП

Программа учебной дисциплины «Введение в объектно-ориентированное программирование» составлена на основании Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) Донецкой Народной Республики (ДНР) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, утвержденного приказом Министерства образования и науки ДНР от 04 апреля 2016 г. № 283;

Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки ДНР № 1171 от «10» ноября 2017 г.;

учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 02.03.02 Фундаментальная информатика и информационные технологии, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

старший преподаватель кафедры прикладной  
математики и теории систем управления



Н.А. Дмитренко

Программа учебной дисциплины утверждена на заседании кафедры прикладной математики и теории систем управления

Протокол № 12 от « 9 » апреля 2020 г.  
Заведующий кафедрой



Д.В. Шевцов

Программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий  
Протокол № 8 от «15» апреля 2020 г.

Председатель учебно-методической  
комиссии факультета



Л.И. Селякова

## 1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Введение в объектно-ориентированное программирование» относится к вариативной части профессионального блока.

Содержание дисциплины является логическим продолжением содержания дисциплин:

- Архитектура вычислительных систем;
- Основы программирования.

и формирует основу для освоения дисциплин:

- Алгоритмы и анализ сложности;
- Интеллектуальные системы;
- Компьютерная графика;
- Программная инженерия;
- Прикладные информационные технологии.
- 

## 2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	02.03.02 Фундаментальная информатика и информационные технологии			
Профиль	общий			
Образовательная программа	бакалавриат			
Квалификация	академический бакалавр			
Количество содержательных модулей	модулей: 1 содержательных модулей: 1			
Дисциплина базовой / вариативной части образовательной программы	вариативная часть профессионального блока			
Формы контроля (МК, экзамен, зачет)	модульный контроль – 1, экзамен – 1			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачетных единиц (кредитов)	3	3		
Год подготовки	2	1		
Семестр	3	1		
Количество часов	108	108		
- лекционных	36	36		
- практических, семинарских	-			
- лабораторных	36	36		
- самостоятельной работы	36	36		
в т.ч. индивидуальное задание				
Недельное количество часов,	6	6		
в т.ч. аудиторных	4	4		

## 3. ОПИСАНИЕ ДИСЦИПЛИНЫ

### Цели и задачи

**Цель** – изучение основных понятий объектно-ориентированного программирования (ООП), алгоритмизации и создания классов; объектно-ориентированного подхода; ООП среды программирования, изучение инструментов языка C# для реализации объектно-ориентированного программирования.

**Задачи** – Ознакомить студентов с концепцией объектно-ориентированного программирования, понятиями и терминами объектно-ориентированного подхода.

**Требования к результатам освоения дисциплины:** Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО

по данному направлению подготовки (профилю):

**а) общекультурных (ОК):**

- способностью использовать основы философских знаний для формирования мировоззренческой позиции (ОК-1);
- способностью к самоорганизации самообразованию (ОК-7).

**б) общепрофессиональных (ОПК)**

- способностью использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с фундаментальной информатикой и информационными технологиями (ОПК-1);
- способностью применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (ОПК-2);
- способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);
- способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4).

**в) профессиональных (ПК):**

**1. научно-исследовательская деятельность:**

- способностью понимать, совершенствовать и применять современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий (ПК-2);
- способностью использовать современные инструментальные и вычислительные средства (ПК-3);
- способностью критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности (ПК-5);

**2. проектная и производственно-технологическая деятельность:**

- способностью применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные и вычислительные средства (ПК-8);

**3. организационно-управленческая деятельность:**

- способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК-11).

**В результате изучения учебной дисциплины студент должен**

**Знать:**

- ✓ основные принципы объектно-ориентированного программирования;
- ✓ базовые концепции программной инженерии;
- ✓ синтаксис и назначение основных конструкций языка программирования C# .

**Уметь:**

- ✓ использовать основные методы и средства разработки алгоритмов;
- ✓ грамотно проектировать программы на языке C# в рамках процедурной и объектно-ориентированной парадигм программирования;
- ✓ определять критерии, каким должен удовлетворять проект, чтобы его легко

- было сопровождать и модифицировать;
- ✓ использовать классы-шаблоны стандартной библиотеки языка C# (STL) и обобщенные алгоритмы при написании программ;
- ✓ применять шаблоны (паттерны) проектирования;
- ✓ уверенно создавать приложения в средах быстрой разработки.

**Владеть:**

- ✓ представлением об объектно-ориентированной парадигме программирования;
- ✓ представлением о наследовании, как одного из эффективных инструментов повторного использования кода;
- ✓ представлением о роли методах и средствах проектирования программного обеспечения.

#### **4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА**

Курс дисциплины «Введение в объектно-ориентированное программирование» предусматривает следующие формы организации учебного процесса: лекции, лабораторных занятия, самостоятельная работа студента.

Материал излагается с использованием объяснительно-иллюстративных, эвристических и исследовательских методов преподавания. При проведении лекций для обсуждения материала широко используются мультимедийные презентации.

В учебном процессе широко применяются активные и интерактивные формы проведения занятий (разбор конкретных ситуаций, дискуссия, полемика), внеаудиторная самостоятельная работа, балльно-рейтинговая система оценки успеваемости, блочно-модульное обучение.

Самостоятельная работа студентов предусматривает выполнение индивидуальных заданий, подготовку к лабораторным занятиям, изучение учебной и методической литературы, составление конспектов, защита индивидуальных заданий.

Порядковый номер и тема	Краткое содержание темы
<b>МОДУЛЬ 1</b>	
<b>Содержательный модуль 1</b> <b>Информационные системы</b>	
<b>Тема 1. Разработка программ и объектно-ориентированное программирование</b>	Разработка программ и объектно-ориентированное программирование. Жизненный цикл программного продукта. Абстракция. Абстрактные типы данных. Основные парадигмы программирования. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Основные цели объектно-ориентированного программирования. Объекты. Классы. Базовые принципы ООП: пакетирование (инкапсуляция), наследование, полиморфизм, передача сообщений. Объектно-ориентированные языки программирования
<b>Тема 2. Язык C#</b>	Язык C#. Цели создания языка. Операции языка. Распределение памяти в C#. Операция расширения области видимости. Ссылки: особенности использования. Ввод-вывод в стиле C#.

Порядковый номер и тема	Краткое содержание темы
<b>Тема 3. Классы в языке C#</b>	Классы в языке C# Поддержка АД в C#. Классы и объекты. Описание класса. Компонентные данные и компонентные функции. Спецификаторы доступа: открытые, защищенные и закрытые компоненты класса. Доступ к открытым и закрытым компонентным данным. Прямой и косвенный доступ к компонентам классов. Операции доступа. Скрытый указатель this.
<b>Тема 4. Конструкторы и деструкторы. Конструктор копирования</b>	Конструкторы и деструкторы Конструктор по умолчанию. Конструктор инициализации. Конструктор копий. Свойства конструкторов. Перегрузка конструктора. Конструктор с присваиванием и со списком инициализации. Деструктор, его свойства. Конструктор копирования. Поверхностное копирование. Побочные эффекты при работе функций. Глубокое копирование. Конструктор копирования. Свойства конструктора копирования. Предотвращение передачи объектов класса по значению.
<b>Тема 5. Статические компоненты классов. Константы</b>	Константы в классе. Инициализация нестатических полей класса. Статические компоненты классов Статические переменные и функции. Статические компоненты классов. Особенности статических полей. Инициализация статических полей. Статические методы. Инициализация статических массивов объектов. Паттерн проектирования «синглет»
<b>Тема 6. Перегрузка операций</b>	Перегрузка операций Механизмы реализации перегрузки операций. Особенности перегрузки операций. Компонентные функции операции. Перегрузка унарных и бинарных операций. Перегрузка операций инкремента и декремента, префиксная и постфиксная формы. Перегрузка операции присваивания. Перегрузка операции индексирования. Функциональная и операторная формы перегрузки операций.
<b>Тема 7. Наследование</b>	Наследование Базовые и производные классы. Простое наследование. Множественное наследование. Ненаследуемые члены класса. Управление доступом к базовому классу. Виды наследования. Моделирование объектно-ориентированных проектов с помощью языка UML. Конструкторы, деструкторы и наследование. Предопределенные стандартные преобразования. Отношения между классами: «является», «содержит» и «подобен»
<b>Тема 8. Полиморфизм в языке C#</b>	Полиморфизм в языке C# Связывание. Статическое и динамическое связывание. Виртуальные функции. Полиморфные объекты. Особенности виртуальных методов. Таблица виртуальных методов VTBL. Указатель VPTR. Абстрактные базовые классы. Чистые виртуальные функции. Интерфейсы
<b>Тема 9. Шаблоны</b>	Полиморфизм функций Перегрузка функций. Неоднозначность и перегрузка. Определение адреса перегруженной функции. Шаблоны классов. Определение метода вне класса-шаблона. Инстанцирование (актуализация) шаблонов. Шаблоны функций:

Порядковый номер и тема	Краткое содержание темы
	глобальные локальные функции. Специализация шаблонов. Композиция и шаблоны. Шаблоны и наследование. Особенности шаблонов.
<b>Тема 10.</b> <b>Исключительные ситуации в C#</b>	Исключительные ситуации в C# Обработка исключительных ситуаций. Обработка ошибок в C#. Генерирование исключений. Перехватывание исключений. Использование вложенных блоков try/catch. Раскрутка стека. Стандартные исключения в C#. Исключения и конструкторы. Информация о типе времени выполнения. Операторы приведения типа.
<b>Тема 11. Стандартная библиотека шаблонов (STL)</b>	Стандартная библиотека шаблонов (STL) Назначение и состав библиотеки. Контейнеры. Последовательные контейнеры. Векторы. Списки. Деки. Операции с деками. Объявление и инициализация дека. Стеки. Объявление и инициализация стека. Очереди. Объявление и инициализация очереди. Ассоциативные контейнеры. Универсальные алгоритмы.

[illegible]



[illegible]

## 5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Темы лекционных занятий

№ п/п	Название темы	Количество часов
1.	Разработка программ и объектно-ориентированное программирование	2
2.	Язык C#	2
3.	Классы в языке C#	4
4.	Конструкторы и деструкторы. Конструктор копирования	2
5.	Статические компоненты классов. Константы	2
6.	Перегрузка операций	4
7.	Наследование	4
8.	Полиморфизм в языке C#	4
9.	Шаблоны	4
10.	Исключительные ситуации в C#	4
11.	Стандартная библиотека шаблонов (STL)	4
	<b>ВСЕГО</b>	<b>36</b>

### Темы лабораторных занятий

№ п/п	Название темы	Количество часов
1.	Разработка программ в среде C#. Знакомство со средой программирования	2
2.	Написание простейших программ в среде C#.	2
3.	Написание простейшего класса Money	2
4.	Уровни доступа. Примеры закрытия доступа к полям и функциям класса.	2
5.	Конструкторы и деструкторы, конструктор копирования на примере класса Money, Date.	2
6.	Статические компоненты классов, константы на примере класса Money, Date.	2
7.	Перегрузка арифметических операций	2
8.	Перегрузка операций сравнения. Допустимые варианты перегрузки.	2
9.	Простейшее наследование на примере геометрических примитивов.	2
10.	Наследование и механизм позднего связывания. Особенности реализации множественного наследования в C#.	2
11.	Пример реализации шаблонов и их использование.	2
12.	Реализация некоторых структур данных с помощью шаблонов.	2
13.	Обработка исключительных ситуаций. Конструкция try..catch	2
14.	Создание собственных исключительных ситуаций и передача параметров, необходимых для обработки исключения.	2
15.	Знакомство со стандартной библиотекой шаблонов	4
16.	Примеры реализации программ, упрощаемых благодаря использованию STL	4
	<b>ВСЕГО</b>	<b>36</b>

## 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Методические рекомендации по организации самостоятельной работы студентов содержатся в учебно-методическом комплексе дисциплины. Самостоятельная работа над учебной дисциплиной для студентов включает:

- овладение теоретическими основами прослушанного лекционного материала;
- изучение отдельных тем или вопросов, предусмотренных для самостоятельного овладения;
- систематику изученного материала перед экзаменом;
- решение и оформление творческих заданий;
- решение и оформление индивидуальных заданий для самостоятельной работы.

Самостоятельная работа подразумевает подготовку студентов к лекционным и лабораторным занятиям, на основании материалов лекций и рекомендованных программой учебников и учебных пособий.

Во второй половине семестра студенты выполняют самостоятельную работу в соответствии с определенным преподавателем вариантом. Самостоятельная работа выполняется студентом в рамках консультаций с преподавателем или в любое другое свободное время студента. Для выполнения работы студентам необходима специализированная аудитория, оснащенной персональными компьютерами.

### Организация самостоятельной работы студентов

№ п/п	Название темы	Количество часов
12.	Разработка программ и объектно-ориентированное программирование	2
13.	Язык C#	2
14.	Классы в языке C#	4
15.	Конструкторы и деструкторы. Конструктор копирования	2
16.	Статические компоненты классов. Константы	2
17.	Перегрузка операций	4
18.	Наследование	4
19.	Полиморфизм в языке C#	4
20.	Шаблоны	4
21.	Исключительные ситуации в C#	4
22.	Стандартная библиотека шаблонов (STL)	4
	<b>ВСЕГО</b>	<b>36</b>

## 7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

В течение года студенты выполняют индивидуальные задания № 1, 2, по каждому из которых осуществляется устная защита с изложением полученных результатов и объяснениями проделанных действий. Индивидуальные задания в количестве 10 вариантов находятся в электронном виде на кафедре.

### ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ (ПРИМЕР) ЗАДАНИЕ № 1.

1. Описать класс «Комплексное число в алгебраической форме», моделирующий число в виде  $x + yi$ . Члены класса: действительная  $x$  и мнимая  $y$  части числа, свойства для работы с действительной и мнимой частями числа. Реализовать конструктор, деструктор, метод вычисления модуля комплексного числа и корня  $n$ -й степени, метод вывода комплексного числа. Осуществить перегрузку операций сложения  $+$ , вычитания  $-$ ,

умножения \*, деления (/) комплексных чисел, увеличения на 1 (++) действительной, мнимой части.

2. Создать базовый класс «массив», описав в нём функции добавления, удаления, вывода на экран и подсчета суммы положительных элементов массива. Порождать от него класс «упорядоченный массив», переопределив функцию добавления элементов массива.

3. Реализовать имитационное моделирование объектно-ориентированной модели реального объекта «принтер», осуществляющей реализацию следующего функционала: принтер имеет три кнопки: «печать» и «заправка бумаги», «печать пробной страницы». Обычная печать происходит в течение 1 минуты, пробная печать – в течение 2х минут. Печать без заправленной бумаги производиться не может. Во время печати можно заправить ещё один лист. Во время печати нажатием на кнопку «печать» ничего не производится. Если бумага заправлена, то повторно заправляться не может. Начальная конфигурация: принтер без бумаги. Разработать объект лазерный черно-белый и цветной принтер, который может печатать различные заданные форматы и тратит на печать при этом различное время.

4. Разработать программу с интерфейсом на основе форм, предусмотреть проверки на вводимые значения с помощью исключений. Разработать класс для представления множества целых чисел на основе односвязного списка. Операции, реализуемого для класса: включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств

## **ЗАДАНИЕ 2.**

Реализация объектно-ориентированной модели игры «Шашки». Необходимо реализовать подмножество правил классических шашек на доске 8x8. Предусмотреть возможность сохранения и загрузки игры. Предусмотреть 2 вида шашек (дамки и обычная шашка), определить операции наследования.

### **Система оценивания.**

Оценка индивидуального задания осуществляется на основе вычисления среднего арифметического оценок, полученных за программную часть и защиту.

#### **Программная часть**

- Неудовлетворительно. Программа заимствована более чем на 75%; программа не работоспособна в принципе.
- Удовлетворительно. Программа заимствована более чем на 25%; уровень сложности не соответствует отведенному под выполнение проекта времени.
- Хорошо. Авторская разработка, основанная на материалах занятий.
- Отлично. Полностью самостоятельная разработка (использование отдельных модулей и библиотек допустимо с указанием источника кода) соответствующего уровня сложности; использование дополнительных материалов по алгоритмам и технологиям программирования; соблюдение принципов объектно-ориентированного программирования, грамотное оформление текста программы (в том числе — использование комментариев); грамотное использование структур данных.

#### **Защита**

- Неудовлетворительно. Отказ от защиты в установленные сроки; неспособность объяснить существенные аспекты работы программы.
- Удовлетворительно. Затруднения с ответом на вопросы; отсутствие логики выступления; неграмотная речь.

- Хорошо. Неполное соответствие требованиям на "Отлично"

Отлично. Грамотная речь с правильным использованием терминологии; заранее продуманная логика выступления; полнота освещения проекта (не путать с демонстрацией программы, для освещения проекта нужно доказать ЧЕТЫРЕ основных положения, а не показывать все варианты работы программы); соблюдение регламента; свободный ответ на вопросы. Приветствуется наличие презентации.

## 8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ И ИТОГОВОЙ АТТЕСТАЦИИ

1. Средства ввода-вывода в языке C#
2. Операции языка C#. Операция выделения памяти в C#.
3. Абстракция данных и классы в языке C#.
4. Конструкторы и деструкторы в классах.
5. Наследование классов.
6. Композиция классов.
7. Статические члены классов.
8. Статическое и динамическое связывание. Виртуальные функции.
9. Перегрузка операций. Особенности перегрузки унарных и бинарных операций.
10. Абстрактные базовые классы.
11. Паттерны проектирования. Итератор и их важность в библиотеке STL.
12. Шаблоны функций и классов в языке C#.
13. Стандартная библиотека языка C#. Шаблоны классов.
14. Стандартная библиотека языка C#. Обобщенные алгоритмы.
15. Перегрузка функций в языке C#.

## 9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

**ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Факультет математики и информационных технологий**

*Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии*

*Профиль:*

*Программа подготовки: бакалавриат*

*Семестр 3*

*Учебная дисциплина Введение в объектно-ориентированное программирование*

### МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №1

#### ВАРИАНТ №1

1. Объяснить понятие паттернов проектирования. Пример.
2. Класс. Виртуальные (virtual) функции. Раннее и позднее связывание.
3. Создать базовый класс «массив», описав в нём функции добавления, удаления, вывода на экран и подсчета суммы положительных элементов массива. Порождать от него класс «упорядоченный массив», переопределив функцию добавления элементов массива

Утверждено на заседании кафедры прикладной математики и теории систем управления, протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ г.

Заведующий кафедрой  
Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

Д.В. Шевцов  
Н.А. Дмитренко

**Критерии оценивания модульного контроля №1**

<i>Номер задания</i>	<i>Количество баллов</i>
1	6
2	6
3	18
<i>Всего</i>	<b>30</b>

**10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА****ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»****Факультет математики и информационных технологий***Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии**Профиль:**Программа подготовки: бакалавриат**Семестр 3**Учебная дисциплина Введение в объектно-ориентированное программирование***ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1**

1. Динамическое и статическое связывание. Для поддержки какого принципа ООП необходим данный механизм.
2. Что такое подход программирования. Дать описание структурного подхода.
3. Построить класс для описания плоской геометрической фигуры: квадрат. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров и вращения на заданный угол.
4. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям.

**Утверждено на заседании кафедры прикладной математики и теории систем управления**

протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

**Заведующий кафедрой**

\_\_\_\_\_

**Д.В. Шевцов****Экзаменатор**

\_\_\_\_\_

**Н.А. Дмитренко****Критерии оценивания экзамена**

<i>Номер задания</i>	<i>Количество баллов</i>
1	20
2	20
3	30
4	30
<i>Всего</i>	<b>100</b>

**11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ**

Не предусмотрено рабочей программой

## 12. КРИТЕРИИ ОЦЕНИВАНИЯ

(разрабатываются и утверждаются кафедрой).

*Распределение баллов, которые могут получить студенты  
в процессе изучения дисциплины*

Организационно учебная работа студента	СРС			Всего
	Индивидуальная работа	Модульный контроль	Индивидуальная творческая работа	
Мах 30 баллов	маx 25 баллов	маx 30 баллов	маx 15 баллов	100 баллов

В процессе работы на протяжении семестра студент может заработать баллы за следующие виды деятельности: индивидуальное задание (домашние работы), самостоятельные и контрольные работы по практике, модульные контрольные работы по теории и практике, активность на занятиях (бонусные баллы), индивидуальные творческие задания (бонусные баллы).

Количество баллов за выполнение индивидуальных заданий представлена в следующей таблице.

Виды контрольных мероприятий	Количество баллов
<b>Индивидуальное задание №1</b>	
Задание 1. Реализация простого класса	5
Задание 2. Реализация простого наследования класса	5
Задание 3. Имитационное объектно-ориентированное моделирование	8
Задание 4. Работа со стандартными шаблонными классами	7
<b>Итого:</b>	<b>25</b>
<b>Индивидуальное задание №2</b>	
Реализация программного комплекса	
<b>Итого:</b>	<b>40</b>

### Критерии оценивания экзамена по курсу

Количество баллов, получаемых на экзамене рассчитывается согласно формуле:

$$x = k + \frac{m-2}{3} \min\{50, 50 - k\},$$

где

$$k = \min\{n, 50\} + \max\{(n-50)/2, 0\}$$

$n$  – кол-во баллов, набранных во время семестра,

$m$  – оценка экзаменационной работы в пятибалльной системе, критерии выставления которой представлены в следующей таблице:

Оценка	Знания, умения, навыки и другие компетенции, которые должен продемонстрировать студент*
Отлично (5)	На вопросы даны исчерпывающие ответы, проиллюстрированные наглядными примерами там, где это необходимо. Ответы изложены грамотным научным языком, все термины употреблены корректно, все понятия раскрыты верно.
Хорошо (4)	На вопросы даны в целом верные ответы, но с отдельными неточностями, не носящими принципиального характера. Не все

	термины употреблены правильно, присутствуют отдельные некорректные утверждения и грамматические / стилистические погрешности изложения. Ответы не проиллюстрированы примерами в должной мере.
Удовлетворительно (3)	Ответы на вопросы носят фрагментарный характер, верные выводы перемежаются с неверными. Упущены содержательные блоки, необходимые для полного раскрытия темы. Студент в целом ориентируется в тематике учебного курса, но испытывает проблемы с раскрытием конкретных вопросов. Также оценка «удовлетворительно» ставится при верном ответе на один вопрос и неудовлетворительном ответе на другой.
Неудовлетворительно (2)	Ответы на вопросы отсутствуют либо не соответствуют содержанию вопросов. Ключевые для учебного курса понятия, содержащиеся в вопросах, трактуются ошибочно.

Оценка за семестр вычисляется как максимальная из полученных за семестр и на экзамене и выставляется согласно шкале, принятой в ДонНУ:

**Шкала оценивания:**

Сумма баллов по 100 балльной шкале	По шкале ECTS	По государственной шкале	Определение
90–100	A	«Отлично» (5)	Отлично (зачтено) – отличное выполнение с незначительным количеством неточностей
80–89	B	«Хорошо» (4)	хорошо (зачтено) – в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
75–79	C		хорошо (зачтено) – в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
70–74	D	«Удовлетворительно» (3)	удовлетворительно (зачтено) – неплохо, но со значительным количеством недостатков
60–69	E		достаточно (зачтено) – выполнение удовлетворяет минимальные критерии
35–59	FX	«Неудовлетворительно» с возможностью повторной аттестации (2)	неудовлетворительно (не зачтено) – надо поработать над тем, как получить положительную оценку
0–34	F	«Неудовлетворительно» с возможностью повторной аттестации при условии обязательного набора дополнительных баллов (2)	неудовлетворительно (не зачтено) – необходимо повторно (дополнительно) пройти курс обучения для возможности получить положительную оценку



В особых случаях, экзамен может быть проведён в дистанционном режиме в системе Moodle. В назначенную дату и время открывается доступ к системе тестирования. Каждому студенту в случайном порядке из заранее заполненного банка экзаменационных вопросов выдаётся компьютерный тест из 30 вопросов, 2 теоретических и 1 практическое задание. На теоретические вопросы студент отвечает письменно, результат отправляет средствами Moodle. Практическое задание может быть выполнено в виде демонстрационной базы данных в Moodle или в виде решения задач по SQL-запросам. Тестирование проводится на время.

По истечению определённого времени доступ к заданиям закрывается (система Moodle позволяет предусмотреть льготный период отправки ответов для студентов с медленным интернет-соединением).

### 13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Для изучения курса студенты обеспечиваются доступом к компьютерному (дисплейному) классу с установленным соответствующим программным обеспечением. Все компьютеры объединены в локальную сеть с доступом в Интернет для возможности проектирования и работы с сетевыми базами данных. Лекционные занятия проводятся в аудитории, оснащенной (мультимедийной техникой и) доской.

### 14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экз. в библиотеке ДонНУ	Наличие электр. верс. в ЭБС
<b>Основная литература</b>			
1.	Курс лекций представлен на кафедре в электронном виде	-	+
2.	Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд./Рихтер Д. – СПб: Питер, 2014.–896 с.		+
3.	Джонсон Р. Приемы объектно-ориентированного проектирования. Паттерны проектирования /Джонсон Р., Гамма Э., Халм Р. – СПб: Питер, 2016.–368с.		+
4.	Рамбо, Дж. UML 2.0. Объектно-ориентированное моделирование и разработка [Текст]: пер. с англ. / Джеймс Рамбо, Мартин Блаха. – СПб: Питер, 2007. – 544 с.		+
5.	Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ [Текст]: пер. с англ. / Гради Буч. – М: Бином, СПб: Невский диалект, 1999. – 560 с.		-
6.	Пышкин, Е.В. Основные концепции и механизмы объектно-ориентированного программирования [Текст]/ Е.В.Пышкин. – СПб: БХВ-Петербург, 2005. – 640 с.		+
7.	Шилдт, Г.. C# 4.0: полное руководство [Текст]: Пер. с англ. / Герберт Шилдт. – М.: ООО "И.Д. Вильямс", 2011. – 1056 с.		+
8.	Дейтел, Х. C# в подлиннике. Наиболее полное руководство [Текст]: пер. с англ./ Харви Дейтел, Пол Дейтел. – СПб: БХВ-Петербург, 2006 г.. – 1056 с.		
9.	Троелсен, Э. Язык программирования C# 2010 и платформа .NET 4 [Текст]: Пер. с англ. / Эндрю Троелсен. – М.: ООО "И.Д. Вильямс", 2011. – 1392 с.		
<b>Дополнительная литература</b>			
10.	Албахари Дж. C# 6.0. Справочник. Полное описание языка / Албахари Дж., Албахари Б. – М.: ООО "И.Д. Вильямс", 2016. – 1040 с.		+
11.	Холл Г. Адаптивный код на C#. Проектирование классов и интерфейсов,		+

	шаблоны и принципы SOLID./Холл Г. - М.: ООО «И.Д. Вильямс», 2016.– 432с.		
--	--	--	--

### 15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Донецкий национальный университет. Научная библиотека ДонНУ [электронный ресурс]: офиц.сайт. / URL: <http://donnu.ru/library> (дата обращения 2020.02.20);
2. Интернет-портал образовательных ресурсов по ИТ. URL: <http://www.intuit.ru> (дата обращения 20.02.2020)

### 16. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

#### 1. Microsoft C#.

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной математики и теории систем управления с изменениями (без изменений) на 20\_\_\_\_ год.  
 Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной математики и теории систем управления с изменениями (без изменений) на 20\_\_\_\_ год.  
 Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной математики и теории систем управления с изменениями (без изменений) на 20\_\_\_\_ год.  
 Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной математики и теории систем управления с изменениями (без изменений) на 20\_\_\_\_ год.  
 Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Заведующий. кафедрой \_\_\_\_\_

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной математики и теории систем управления с изменениями (без изменений) на 20\_\_\_\_ год.  
 Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Заведующий. кафедрой \_\_\_\_\_