

Документ подписан простой электронной подписью.
Информация о владельце:
ФИО: Костина Лариса Николаевна
Должность: проректор
Дата подписания: 24.07.2025 05:10:54
Уникальный программный ключ:
1800f7d89cf4ea7507265ba593fe87537eb15a6c

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ"

Факультет

Факультет государственной службы и управления

Кафедра

Информационных технологий

"УТВЕРЖДАЮ"

Проректор

_____ Л.Н. Костина

24.04.2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.11

"Информатика и программирование"

Направление подготовки 09.03.03 Прикладная информатика

Профиль "Прикладная информатика в управлении корпоративными информационными системами"

Квалификация

Бакалавр

Форма обучения

очная

Общая трудоемкость

15 ЗЕТ

Год начала подготовки по учебному плану

2025

Донецк
2025

Составитель(и):
, *ст.препод.*

_____ Лебезова Э. М.

Рецензент(ы):
канд. экон. наук, доцент

_____ Стешенко И. В.

Рабочая программа дисциплины (модуля) "Информатика и программирование" разработана в соответствии с:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (Приказ Министерства образования и науки Российской Федерации от 19.09.2017 г.№ 922)

Рабочая программа дисциплины (модуля) составлена на основании учебного плана Направление подготовки 09.03.03 Прикладная информатика Профиль "Прикладная информатика в управлении корпоративными информационными системами", утвержденного Ученым советом ФГБОУ ВО "ДОНАУИГС" от 24.04.2025 протокол № 12.

Срок действия программы: 2025-2029

Рабочая программа рассмотрена и одобрена на заседании кафедры Информационных технологий

Протокол от 02.04.2025 № 9

Заведующий кафедрой:
канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2026 - 2027 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2026 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2027 - 2028 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2027 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2028 - 2029 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2028 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2029 - 2030 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2029 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

РАЗДЕЛ 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ

1.1. ЦЕЛИ ДИСЦИПЛИНЫ

Целями изучения дисциплины являются формирование у будущих специалистов общего представления о фундаментальных понятиях, используемых во всех курсах Computer Science, практических навыков по основам алгоритмизации вычислительных процессов и программированию решения экономических, вычислительных и других задач, развитие умения работы с персональным компьютером на уровне системного администратора, обучение работе с научно-технической литературой и технической документацией по программному обеспечению ПЭВМ.

1.2. УЧЕБНЫЕ ЗАДАЧИ ДИСЦИПЛИНЫ

1. Изучить принципы сбора, отбора и обобщения информации.
2. Уметь применять основные алгоритмические подходы при автоматизации практических задач.
3. Овладеть основами архитектурных решений при проектировании программы на языках высокого уровня с помощью структурной и объектно-ориентированной парадигм.
4. Овладеть практическим опытом работы с информационными источниками, опытом научного поиска, создания новых программных решений на базе существующих.
5. Овладеть интегрированными средами разработки программного обеспечения, специализированными библиотеками и платформами.

1.3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОПОП ВО: Б1.О

1.3.1. Дисциплина "Информатика и программирование" опирается на следующие элементы ОПОП ВО:

Алгоритмизация игровых задач

1.3.2. Дисциплина "Информатика и программирование" выступает опорой для следующих элементов:

Программная инженерия

Интернет программирование

Разработка программных приложений

Разработка электронного портала

1.4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

ОПК-7.1: Разрабатывает алгоритмы и программы, пригодные для практического применения

Знать:

Уровень 1 Знать стандартные алгоритмы для решения практических задач

Уровень 2 Знать основные структуры данных для реализации алгоритмов и программ, пригодных для практического применения

Уровень 3 Знать технологии реализации базовых алгоритмов программирования и структур данных на языках программирования Python и C#

Уметь:

Уровень 1 Уметь создавать программы на языках высокого уровня с помощью структурной парадигмы

Уровень 2 Уметь создавать программы на языках высокого уровня с помощью объектно-ориентированной парадигмы

Уровень 3 Уметь анализировать и реализовывать базовые алгоритмы программирования и структуры данных

Владеть:

Уровень 1 Владеть интегрированными средами разработки для создания прикладных программ

Уровень 2 Владеть архитектурными паттернами для создания программ, пригодных для практического применения

Уровень 3 Владеть навыками разработки алгоритмов для проведения экспериментальных исследований в области информатики

В результате освоения дисциплины "Информатика и программирование" обучающийся

3.1 Знать:

– основные понятия информатики;

– изобразительные средства описания алгоритмов;

	– основные приемы алгоритмизации и программирования на языке высокого уровня;
	– основные типы алгоритмов и их использование для решения вычислительных, инженерных, экономических и других типов прикладных задач;
	– основные структуры данных, способы их представления и обработки;
	– системы программирования на алгоритмических языках высокого уровня (C++, C#);
	– принципы разработки программ;
	– основные методы автономной и комплексной отладки и тестирования простых программ;
	– технологический процесс подготовки и решения задач на ПЭВМ
3.2	Уметь:
	– разрабатывать алгоритмы решения и программировать задачи обработки данных в предметной области;
	– разрабатывать проект тестирования программы, выполнять тестирование и отладку программ;
	– оформлять программную документацию.
3.3	Владеть:
	– культурой мышления, способностью к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения;
	– способностью к саморазвитию, повышению своей квалификации и мастерства;
	– способностью работать с информацией в глобальных компьютерных сетях;
	– пониманием сущности и значение информации в развитии современного информационного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны;
	– способностью использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии в соответствии с целями образовательной программы бакалавра.
1.5. ФОРМЫ КОНТРОЛЯ	
Текущий контроль успеваемости позволяет оценить уровень сформированности элементов компетенций (знаний, умений и приобретенных навыков), компетенций с последующим объединением оценок и проводится в форме: устного опроса на лекционных и семинарских/практических занятиях (фронтальный, индивидуальный, комплексный), письменной проверки (тестовые задания, контроль знаний по разделу, ситуационных заданий и т.п.), оценки активности работы обучающегося на занятии, включая задания для самостоятельной работы.	
Промежуточная аттестация	
Результаты текущего контроля и промежуточной аттестации формируют рейтинговую оценку работы студента. Распределение баллов при формировании рейтинговой оценки работы студента осуществляется в соответствии с действующим локальным нормативным актом. По дисциплине "Информатика и программирование" видом промежуточной аттестации является Курсовая работа	

РАЗДЕЛ 2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. ТРУДОЕМКОСТЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ						
Общая трудоёмкость дисциплины "Информатика и программирование" составляет 15 зачётные единицы, 540 часов.						
Количество часов, выделяемых на контактную работу с преподавателем и самостоятельную работу обучающегося, определяется учебным планом.						
2.2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ						
Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
Раздел 1. Рвздел №1. Алгоритмизация и программирование на языках высокого уровня						
Тема 1.1. Представление информации в	1	2	ОПК-7.1	Л1.2	0	

ЭВМ. Состав языка программирования Python /Лек/				Л1.3Л2.2 Э1 Э4		
Тема 1.1. Представление информации в ЭВМ. Состав языка программирования Python /Пр/	1	6	ОПК-7.1	Л1.2 Л1.3Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.1. Представление информации в ЭВМ. Состав языка программирования Python /Ср/	1	7	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.2. Ввод-вывод, базовые типы данных, основные операции Python /Лек/	1	2	ОПК-7.1	Л1.2 Л1.3Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.2. Ввод-вывод, базовые типы данных, основные операции Python /Пр/	1	6	ОПК-7.1	Л1.2 Л1.3Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.2. Ввод-вывод, базовые типы данных, основные операции Python /Ср/	1	7	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.3. Ветвления в программах с помощью операторов принятия решений /Лек/	1	2	ОПК-7.1	Л1.2 Л1.3Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.3. Ветвления в программах с помощью операторов принятия решений /Пр/	1	6	ОПК-7.1	Л1.2 Л1.3Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.3. Ветвления в программах с помощью операторов принятия решений /Ср/	1	7	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.4. Циклические алгоритмы, операторы циклов /Лек/	1	2	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.4. Циклические алгоритмы, операторы циклов /Пр/	1	6	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 1.4. Циклические алгоритмы, операторы циклов /Ср/	1	5	ОПК-7.1	Л1.2Л2.2Л3 .1 Л3.2 Л3.3 Л3.4 Э1 Э4	0	

Тема 1.5. Строки. /Лек/	1	4	ОПК-7.1	Л2.2Л3.2 Л3.3 Л3.4	0	
Тема 1.5. Строки. /Пр/	1	12	ОПК-7.1	Л1.2Л2.2Л3 .2 Л3.3 Л3.4	0	
Тема 1.5. Строки. /Ср/	1	7		Л1.2Л2.2Л3 .3	0	
Тема 1.6. Списки. Списочные выражения /Лек/	1	4		Л1.2 Л1.3Л2.2Л3 .3	0	
Тема 1.6. Списки. Списочные выражения /Пр/	1	12		Л1.2Л2.2Л3 .3 Л3.4	0	
Тема 1.6. Списки. Списочные выражения /Ср/	1	18		Л1.1Л2.2Л3 .4	0	
/Конс/	1	2		Л2.2	0	
Раздел 2. Раздел № 2. Структуры данных и файлов на Python						
Тема 2.1. Генерация случайных чисел в программировании задач моделирования и игр. /Лек/	2	2	ОПК-7.1	Л1.2Л2.1 Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.1. Генерация случайных чисел в программировании задач моделирования и игр. /Пр/	2	6	ОПК-7.1	Л1.2Л2.1 Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.1. Генерация случайных чисел в программировании задач моделирования и игр. /Ср/	2	14	ОПК-7.1	Л1.2Л2.1 Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.2. Парадигмы программирования. Функции и их применение. /Лек/	2	4	ОПК-7.1	Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	

Тема 2.2. Парадигмы программирования. Функции и их применение. /Пр/	2	10		Л1.2	0	
Тема 2.2. Парадигмы программирования. Функции и их применение. /Ср/	2	12		Л1.2Л2.2	0	
Тема 2.3. Кортежи. Поиск и сортировка. /Лек/	2	4	ОПК-7.1	Л1.2Л2.1 Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.3. Кортежи. Поиск и сортировка. /Пр/	2	18	ОПК-7.1	Л1.2Л2.1Л3 .2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.3. Кортежи. Поиск и сортировка. /Ср/	2	12	ОПК-7.1	Л1.2Л2.1Л3 .2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.4. Множества. Анализ сложности алгоритмов. /Лек/	2	2	ОПК-7.1	Л1.2Л2.1Л3 .2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.4. Множества. Анализ сложности алгоритмов. /Пр/	2	6	ОПК-7.1	Л1.2Л2.1Л3 .2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.3. Множества. Словари. Файлы. Анализ сложности алгоритмов. /Ср/	2	20	ОПК-7.1	Л1.2Л2.1Л3 .2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.5. Словари /Лек/	2	2	ОПК-7.1	Л2.2Л3.2 Л3.3 Л3.4 Э1 Э4	0	
Тема 2.5. Словари /Пр/	2	6	ОПК-7.1	Л1.2Л2.2Л3 .2 Л3.3 Л3.4 Э1	0	
Тема 2.5. Словари /Ср/	2	12		Л1.2 Э1	0	
Тема 2.6. Файлы /Лек/	2	4		Л1.2Л2.2 Э1	0	

Тема 2.6. Файлы /Пр/	2	8		Л1.2 Э1	0	
Консультация /Конс/	2	2		Э1	0	
Раздел 3. Раздел №3. Объектно-ориентированное программирование на C#						
Тема 3.1. Введение в объектно-ориентированное программирование. Классы и объекты /Лек/	3	4	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 3.1. Введение в объектно-ориентированное программирование. Классы и объекты /Пр/	3	16	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.1. Введение в объектно-ориентированное программирование. Классы и объекты /Ср/	3	14	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. /Лек/	3	8	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. /Пр/	3	16	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. /Ср/	3	12	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.3. Построение программ на основе фундаментальных принципов ООП. Шаблоны проектирования /Лек/	3	4	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.3. Построение программ на основе фундаментальных принципов ООП. Шаблоны проектирования /Пр/	3	16	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	
Тема 5.3. Построение программ на основе фундаментальных принципов ООП. Шаблоны проектирования /Ср/	3	12	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Л3.4 Э2 Э3	0	

Консультация /Конс/	3	2			0	
Раздел 4. Раздел №4. Технологии создания приложений на платформе .NET						
Тема 4.1. Разработка в среде MS Visual Studio. Графика, анимация. /Лек/	4	8	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема 4.1. Разработка в среде MS Visual Studio. Графика, анимация. /Пр/	4	12	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема 4.1. Технологии построения оконных приложений в среде MS Visual Studio /Ср/	4	10	ОПК-7.1	Л1.2Л2.1Л3 .4 Э2 Э3	0	
Тема 4.2. Структура проектов WindowsForm и WPF. Компонировка. Каркас приложения. Модель данных. /Лек/	4	8	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема 4.2. Структура проектов WindowsForm и WPF. Компонировка. Каркас приложения. Модель данных. /Пр/	4	12	ОПК-7.1	Л1.2Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема 4.2. Структура проектов WindowsForm и WPF. Компонировка. Каркас приложения. Модель данных. /Ср/	4	10	ОПК-7.1	Л1.2Л2.1Л3 .4 Э2 Э3	0	
Тема 4.3 Интеграция приложений через API. /Лек/	4	6	ОПК-7.1	Л1.2 Э1 Э3	0	
Тема 4.3. Интеграция приложений через API. /Пр/	4	10	ОПК-7.1	Л1.2Л3.2 Л3.3 Э2 Э3	0	
Тема 4.3. Интеграция приложений через API. /Ср/	4	10	ОПК-7.1	Л1.2Л3.4 Э2 Э3	0	
Тема 4.4 Создание приложений для Баз данных. /Лек/	4	10	ОПК-7.1	Л1.1Л3.2 Л3.3 Э2 Э3	0	

Тема 4.4 Создание приложений для Баз данных. /Пр/	4	14	ОПК-7.1	Л1.1Л3.1 Э2 Э3	0	
Тема 4.4 Создание приложений для Баз данных. /Ср/	4	5	ОПК-7.1	Л1.1Л3.4 Э2 Э3	0	
Консультация по темам раздела /Конс/	4	2	ОПК-7.1	Л1.1Л3.4 Э2 Э3	0	

РАЗДЕЛ 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе освоения дисциплины используются следующие образовательные технологии: лекции (Л), практические занятия (ПР), самостоятельная работа студентов (СР) по выполнению различных видов заданий.

1. В процессе освоения дисциплины используются следующие интерактивные образовательные технологии: проблемная лекция (ПЛ). Лекционный материал представлен в виде слайд-презентации в формате «Power Point». Для наглядности используются материалы различных научных и технических экспериментов, справочных материалов, научных статей т.д. В ходе лекции предусмотрена обратная связь со студентами, активизирующие вопросы, просмотр и обсуждение видеофильмов. При проведении лекций используется проблемно-ориентированный междисциплинарный подход, предполагающий творческие вопросы и создание дискуссионных ситуаций.

2. При изложении теоретического материала используются такие методы:

- монологический;
- показательный;
- диалогический;
- эвристический;
- исследовательский;
- проблемное изложение.

3. Используются следующие принципы дидактики высшей школы:

- последовательность обучения;
- систематичность обучения;
- доступность обучения;
- принцип научности;
- принципы взаимосвязи теории и практики;
- принцип наглядности и др.

В конце каждой лекции предусмотрено время для ответов на проблемные вопросы.

4. Практические занятия проходят в компьютерных классах для приобретения и закрепления практических навыков программирования. Выполняя задания для практических работ, студенты осваивают новые технологии, изучают современные языки программирования и накапливают опыт разработки программного обеспечения.

5. Самостоятельная работа предназначена для внеаудиторной работы студентов, связанной с конспектированием источников, учебного материала, изучением дополнительной литературы по дисциплине, подготовкой к текущему и семестровому контролю, а также выполнением индивидуального задания в форме реферата, эссе, презентации, эмпирического исследования.

РАЗДЕЛ 4. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

4.1. Рекомендуемая литература			
1. Основная литература			
	Авторы,	Заглавие	Издательство, год
Л1.1	Н. А. Тюкачев, В. Г. Хлебостроев	С#. Основы программирования: учебное пособие для вузов (272 с.)	4-е изд., стер. — Санкт-Петербург :Лань : электронно-библиотечная система, 2021
Л1.2	А. А. Андрианова, Л. Н. Исмаилов, Т. М. Мухтарова.	Алгоритмизация и программирование: Практикум : учебное пособие (240 с.)	Санкт-Петербург - Лань : электронно-библиотечная система, 2022
Л1.3	Гегечкори Е. Т.	Программирование на языке Python: учеб. пособие : Учебное пособие (172)	Омский государственный технический университет, 2023
2. Дополнительная литература			
	Авторы,	Заглавие	Издательство, год
Л2.1	Залогова, Л. А.	Основы объектно-ориентированного программирования на базе языка С# : учебное пособие для вузов (192 с.)	3-е изд., стер. — Санкт-Петербург, 2021
Л2.2	Шихи Д.Р.	Структуры данных в Python: начальный курс: Учебное пособие (186 с.)	ДМК Пресс, 2022
3. Методические разработки			
	Авторы,	Заглавие	Издательство, год
Л3.1	Н. В. Брадул, Э. М. Лебезова Н. В. Брадул, Э. М. Лебезова	Информатика и программирование: Методические рекомендации по выполнению курсовой работы для студентов 1 курса направления подготовки 09.03.03 «Прикладная информатика» очной и заочной форм обучения (31 с.)	Донецк : ФГБОУ ВО "ДОНАУИГС", 2025
Л3.2	Лебезова Э. М. Лебезова Э. М.	Методические рекомендации для проведения практических занятий(по организации самостоятельной работы обучающихся) по учебной дисциплине «Информатика и программирование» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (30 с.)	Донецк : ФГБОУ ВО "ДОНАУИГС", 2025
Л3.3	Лебезова Э.М.	Конспект лекций по учебной дисциплине «Информатика и программирование» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (180 с.)	Донецк : ФГБОУ ВО "ДОНАУИГС", 2025
Л3.4	Лебезова Э.М.	Методические рекомендации по организации самостоятельной работы обучающихся по учебной дисциплине «Информатика и программирование» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (22 с.)	Донецк : ФГБОУ ВО "ДОНАУИГС", 2025
4.2. Перечень ресурсов			
информационно-телекоммуникационной сети "Интернет"			
Э1	Электронный курс в системе Moodle		https://elearn.donampa.ru/course/view.php?id=4671
Э2	Электронный курс в системе Moodle		https://elearn.donampa.ru/course/view.php?id=18
Э3	Изучение языка Python		https://www.tutorialspoint.com/python/index.htm
Э4	Изучение языка C#		https://www.tutorialspoint.com/csharp/index.htm

		m
Э5	ЭБС «ЗНАНИУМ»	https://znanium.ru
Э6	Научная электронная библиотека «КиберЛенинка»	https://cyberleninka.ru/
Э7	ЭБС «ЛАНЬ»	https://e.lanbook.com
Э8	ЭБС «SOCHUM»	https://sochum.ru

4.3. Перечень программного обеспечения

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства:

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства:

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства:

- Libre Office (лицензия Mozilla Public License v2.0.)
- 7-Zip (лицензия GNU Lesser General Public License)
- AIMP (лицензия LGPL v.2.1)
- STDU Viewer (freeware for private non-commercial or educational use)
- GIMP (лицензия GNU General Public License)
- Inkscape (лицензия GNU General Public License).

4.4. Профессиональные базы данных и информационные справочные системы

www.citforum.ru/ - портал аналитических и научных статей в области информационных технологий;
www.rsdn.ru - сайт Российской сети разработчиков ПО, содержит статьи по современным средствам программирования.

<http://biblioclub.ru/> - «ЭБС Университетская библиотека онлайн», Каталог курсов «Информационные технологии»;

<https://rucont.ru/collections/5610> - «ЭСБ Руконт», Каталог курсов «Информатика и вычислительная техника».

4.5. Материально-техническое обеспечение дисциплины

Для проведения учебных занятий, предусмотренных образовательной программой, закреплены аудитории согласно расписанию учебных занятий:

рабочее место преподавателя, посадочные места по количеству обучающихся, доска меловая, персональный компьютер с лицензированным программным обеспечением общего назначения, мультимедийный проектор, экран, интерактивная панель.

РАЗДЕЛ 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

Вопросы к Зачёту с оценкой (1 семестр)

1. Структура программы на С и С++.
2. Заголовочные файлы и компоновка программы.
3. Стандартные библиотеки С и С++.
4. Ввод и вывод в С и С++.
5. Математические вычисления в С и С++.
6. Алфавит и лексемы С++.
7. Операторы и операции.
8. Фундаментальные типы: логический тип, символьные типы, целые типы, типы с плавающей точкой.
9. Тип void.
10. Указатели.
11. Массивы.
12. Ссылки.
13. Объявления и определения.
14. Константы.
15. Область видимости.
16. Инициализация.
17. Указатели и адресная арифметика.
18. Операции над указателями.
19. Стандартные операторы С++.
20. Инструкции С++: инструкция-объявление, инструкция-выражение, составная инструкция,

помеченная инструкция, инструкция выбора, инструкция итерации, инструкция передачи управления, блок try.

21. Объявления и определения функций.
22. Аргументы функций.
23. Типы функций.
24. Вызовы функций.
25. Аргументы по умолчанию.
26. Неуказанное количество аргументов.
27. Встраивание функций.
28. Перегрузка функций.
29. Указатели на функции.
30. Объявления массивов.
31. Массивы и указатели.
32. Одномерные и многомерные массивы.
33. Массивы как аргументы функции.

Вопросы к Зачёту с оценкой (2 семестр)

1. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
2. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.
3. Функции, ориентированные на работу со строками. Примеры.
4. Процедуры, ориентированные на работу со строками. Примеры.
5. "Пузырьковая" сортировка (обменами).
6. Сортировка выбором.
7. Сортировка вставками.
8. Примеры рекурсивных функций: количество цифр в записи данного натурального числа; поиск минимального элемента в массиве; проверка, является ли число простым.
9. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.
10. Оценка сложности алгоритмов. O-нотация.
11. Модульные тесты. Внедрение тестирования.
12. Список List в C#
13. Словарь Dictionary в C#
14. Файлы и каталоги
15. Однонаправленные списки и действия с ними.
16. Двухнаправленные списки.
17. Кольцевые списки. Списки с ключом.
18. Стек.
19. Очередь.
20. Дек.
21. Двоичные деревья. Формирование дерева. Добавление элемента в дерево. Печать дерева. Способы обхода дерева.
22. Двоичные деревья. Исключение компонента из дерева.
23. Библиотека STL. Классы контейнеры d C++.
24. Работа с классом Vector библиотеки STL
25. Работа с классом Map библиотеки STL
26. Работа с классом Dictionary библиотеки STL

Вопросы к экзамену (3 семестр)

1. Понятие об ООП. Отличие ООП подхода к программированию от структурного.
2. Основные понятия ООП. Примеры.
3. Дать определение класса
4. Дать определение экземпляра класса
5. Дать определение интерфейса
6. Чем свойства отличаются от полей
7. Как в C# реализуется множественное наследование
8. Перегрузка методов
9. Виртуальные классы и виртуальные методы

10. Что такое индексы классов?
11. Для чего создают абстрактные классы?
12. Дать определение 4 концепциям ООП
13. Перегрузка операций, особенности реализации в C#
14. Модификаторы доступа
15. Реализация полиморфизма в C#
16. Ключевое слово `sealed`
17. Как в приложениях Windows Forms интерфейс отделён от бизнес-логики
18. Какие управляющие графические элементы вы знаете?
19. Как сделать элемент невидимым, и как недоступным?
20. С помощью какого элемента можно имитировать временные задержки?
21. Из каких основных файлов состоит проект Windows Forms?
22. От какого класса наследуются все классы в C#?
23. Назовите отличия между интерфейсом и абстрактным классом?
24. Чем отличается `event` от `delegate`?
25. Инкапсуляция. Примеры.
26. Наследование. Примеры.
27. Полиморфизм. Примеры.
28. Что такое комбинированный тип данных (запись)? Способы описания (примеры).
29. Способы обращения к компонентам записи.
30. Оператор присоединения и его использование.
31. Записи с вариантами. Примеры.
32. Понятие множества. Обозначение множества в C++.
33. Задание множественного типа и множественная переменная.
34. Операции над множествами. Примеры.
35. Понятие файла. Логические и физические файлы.
36. Файловые типы C++ (общие понятия).
37. Понятие текущей длины файла, указателя или окна файла, буфера файла.
38. Общие процедуры и функции для работы с файлами.
39. Компонентные (типизированные) файлы: описание, отличие от других видов файлов.
40. Текстовые файлы, принципы работы. Процедуры и функции, ориентированные на работу с текстовыми файлами, ввод-вывод информации.
41. Нетипизированные файлы и процедуры ввода-вывода. Примеры использования.
42. Прямой и последовательный доступ к компонентам файлов.
43. Работа с файлами записей. Примеры.
44. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.
45. Построение диаграмм.
46. Построение графиков функций.

5.2. Темы письменных работ

Тематика курсовых работ

1. Информационная система. Личная библиотека
2. Информационная система. Картотека сотрудников
3. Информационная система. Картотека студентов
4. Информационная система. Каталог статей
5. Информационная система. Телефонный справочник
6. Информационная система. Каталог товаров
7. Информационная система. Справочник фирм
8. Информационная система. Справочник жильцов
9. Информационная система. Каталог автомобилей
10. Информационная система. Библиотечный каталог
11. Информационная система. Справочник программ
12. Информационная система. Каталог сотовых телефонов
13. Информационная система. Расписание движения поездов
14. Информационная система. Каталог комплектующих
15. Информационная система. Справочник специальностей

5.3. Фонд оценочных средств

Фонд оценочных средств дисциплины "Информатика и программирование" разработан в соответствии с локальным нормативным актом ФГБОУ ВО "ДОНАУИГС".

Фонд оценочных средств дисциплины "Информатика и программирование" в полном объеме

представлен в виде приложения к данному РПД.

5.4. Перечень видов оценочных средств

Тестирование (проводится на практических занятиях; контроль знаний раздела учебной дисциплины)
 Практические задания (выполняются на практических занятиях за компьютером)
 Индивидуальные задания (разноуровневые задачи и задания для самостоятельной работы)
 Контрольные задания (выполняются на практических занятиях)

РАЗДЕЛ 6. СРЕДСТВА АДАПТАЦИИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ К ПОТРЕБНОСТЯМ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

- 1) с применением электронного обучения и дистанционных технологий.
- 2) с применением специального оборудования (техники) и программного обеспечения, имеющихся в ФГБОУ ВО "ДОНАУИГС".

В процессе обучения при необходимости для лиц с нарушениями зрения, слуха и опорно-двигательного аппарата предоставляются следующие условия:

- для лиц с нарушениями зрения: учебно-методические материалы в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные задания и консультации.

- для лиц с нарушениями слуха: учебно-методические материалы в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: учебно-методические материалы в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

РАЗДЕЛ 7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО УСВОЕНИЮ ДИСЦИПЛИНЫ

Аудиторные занятия по дисциплине проводятся в форме лекционных и практических занятий.

На лекционных занятиях, согласно учебному плану дисциплины, обучающимся предлагается рассмотреть основные темы курса. Студенту предлагается участвовать в диалоге с преподавателем, в ходе которого могут обсуждаться моменты, актуальные для его будущей практической деятельности; он может высказать свое мнение после сопоставления разных фактов и разнообразных точек зрения на них.

К числу важнейших умений, являющихся неотъемлемой частью успешного учебного процесса, относится умение работать с различными литературными источниками, содержание которых так или иначе связано с изучаемой дисциплиной.

Подготовку к любой теме курса рекомендуется начинать с изучения презентационных материалов или учебной литературы, в которых дается систематизированное изложение материала, разъясняется смысл разных терминов и сообщается об изменениях в подходах к изучению тех или иных проблем данного курса.

В курсе широко используются актуальные и эффективные техники для более качественного обучения, социализации:

- обсуждения тем в специальных группах популярных социальных сетей;
- командная работа;
- удаленные технические консультации и видеоконференции;
- наглядные демонстрации современных интернет технологий.

Методические указания по организации самостоятельной работы

Самостоятельная работа по дисциплине организована в следующих видах:

1. изучение теоретического материала по заданной теме;
2. анализ методов решения поставленной задачи;
3. выполнение индивидуальных заданий;
4. оценка достоверности полученных результатов;
5. отчет перед преподавателем по теоретической и практической части индивидуальной работы;
6. самостоятельная работа над индивидуальными заданиями представлены в виде элементов электронного курса в системе elearn

<http://elearn.dsum.org/course/view.php?id=10>

<http://elearn.dsum.org/course/view.php?id=18>

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ»**

**Факультет государственной службы и управления
Кафедра информационных технологий**

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

«Информатика и программирование»

Направление подготовки	09.03.03 Прикладная информатика
Профиль	«Прикладная информатика в управлении корпоративными информационными системами»
Квалификация	бакалавр
Форма обучения	очная

Донецк
2025

Фонд оценочных средств по дисциплине «Информатика и программирование» для обучающихся 1-2 курса образовательной программы бакалавриата направления подготовки 09.03.03 Прикладная информатика (профиль: «Прикладная информатика в управлении корпоративными информационными системами») очной формы обучения

Автор,
разработчик:

стар. преподаватель Лебезова Э. М.

ФОС рассмотрен на заседании
кафедры

информационных технологий

Протокол заседания кафедры от

02.04.2025 г.

№

№ 9

Заведующий кафедрой
Брадул

Н.В.

РАЗДЕЛ 1.**ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине «Информатика и программирование»****1.1. Основные сведения об учебной дисциплине**

Таблица 1

Характеристика учебной дисциплины
(сведения соответствуют разделу РПУД)

Образовательная программа	бакалавриат
Направление подготовки	09.03.03 Прикладная информатика «Прикладная информатика в управлении корпоративными информационными системами»
Количество разделов учебной дисциплины	8
Часть образовательной программы	Б1.О.11
Формы текущего контроля	Индивидуальные задания, практические задания, контрольные задания, тестирование
<i>Показатели</i>	Очная форма обучения
Количество зачетных единиц (кредитов)	15
Семестр	1,2,3,4
Общая трудоемкость (академ. часов)	540 (144/144/108/144)
Аудиторная контактная работа:	66/74/66/82
Лекционные занятия	16/18/16/32
Практические занятия	48/54/48/48
Консультации	2/2/2/2
Самостоятельная работа	51/70/38/35
Контроль	27/4/27
<i>Форма промежуточной аттестации</i>	экзамен/экзамен/зачёт с оценкой/экзамен

1.2. Перечень компетенций с указанием этапов формирования в процессе освоения образовательной программы.

Таблица 2

Перечень компетенций и их элементов			
Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
ОПК-7.1	Разрабатывает алгоритмы и программы, пригодные для практического применения	Знать:	
		1. Знать стандартные алгоритмы для решения практических задач	ОПК 7.1 3-1
		2. Знать основные структуры данных для реализации алгоритмов и программ, пригодных для практического применения	ОПК 7.1 3-2
		3. Знать технологии реализации базовых алгоритмов программирования и структур данных на языках программирования Python и C#	ОПК 7.1 3-3
		Уметь:	
		1. Уметь создавать программы на языках высокого уровня с помощью структурной парадигмы	ОПК 7.1 У-1
		2. Уметь создавать программы на языках высокого уровня с помощью объектно-ориентированной парадигмы	ОПК 7.1 У-2
		3. Уметь анализировать и реализовывать базовые алгоритмы программирования и структуры данных	ОПК 7.1 У-3
		Владеть:	
		1. Владеть интегрированными средами разработки для создания прикладных программ	ОПК 7.1 В-1
		2. Владеть архитектурными паттернами для создания программ, пригодных для практического применения	ОПК 7.1 В-2

Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
		3. Владеть навыками разработки алгоритмов для проведения экспериментальных исследований в области информатики	ОПК 7.1 В-3

Таблица 3

Этапы формирования компетенций в процессе освоения основной образовательной программы

№ п/п	Контролируемые разделы (темы) учебной дисциплины	Этапы формирования компетенций, номер семестра	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
Раздел №1. Алгоритмизация и программирование на языках высокого уровня				
1.	Тема 1.1. Представление информации в ЭВМ. Состав языка программирования Python	1	ОПК-7 У-1 ОПК-7 В-1	Индивидуаль-ные задания, практические задания
2.	Тема 1.2. Ввод-вывод, базовые типы данных, основные операции Python	1	ОПК-7 3-1 ОПК-7 3-4	Индивидуаль-ные задания, , практические задания
3	Тема 1.3. Ветвления в программах с помощью операторов принятия решений	1	ОПК-7 3-1 ОПК-7 У-2	Индивидуаль-ные задания, практические задания
4.	Тема 1.4. Циклические алгоритмы, операторы циклов	1	ОПК-7 У-1	Индивидуаль-ные задания, практические задвния
5.	Тема 1.5. Строки	1	ОПК-7 3-1 ОПК-7 У-1	Индивидуаль-ные задания, практические задвния
6.	Тема 1.6. Списки. Списочные выражения	1	ОПК-7 3-2 ОПК-7 У-2	Индивидуаль-ные задания, практические задвния, тест

Раздел № 2. Структуры данных и файлов на Python				
7.	Тема 2.1. Генерация случайных чисел в программировании задач моделирования и игр	1	ОПК-7 3-1 ОПК-7 В-1	Индивидуаль-ные задания, практические задания
8.	Тема 2.2. Парадигмы программирования. Функции и их применение.	1	ОПК-7 У-1 ОПК-7 В-1	Индивидуаль-ные задания, , практические задания
9.	Тема 2.3. Кортежи. Поиск и сортировка.	1	ОПК-7 У-2 ОПК-7 В-2	Индивидуаль-ные задания, практические задания
10.	Тема 2.4. Множества. Анализ сложности алгоритмов.	1	ОПК-7 У-2 ОПК-7 В-2	Индивидуаль-ные задания, практические задания
11	Тема 2.5. Словари	1	ОПК-7 У-2 ОПК-7 В-2	Индивидуаль-ные задания, практические задания
12	Тема 2.6. Файлы	1	ОПК-7 У-2 ОПК-7 В-2	Индивидуаль-ные задания, практические задания, тест
Раздел № 3. Объектно-ориентированное программирование на C#				
13.	Тема 3.1. Введение в объектно-ориентированное программирование. Классы и объекты	2	ОПК-7 У-1 ОПК-7 3-1	Индивидуаль-ные задания, практические задания
14.	Тема 3.2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	2	ОПК-7 У-1, ОПК-7 В-1	Индивидуаль-ные задания, практические задания, тест
15.	Тема 3.3. Построение программ на основе фундаментальных принципов ООП. Шаблоны проектирования	2	ПК-2 В-1 ПК-24 3-1	Индивидуаль-ные задания, практические задания, тест
Раздел №4. Технологии создания приложений на платформе .NET (				
16.	Тема 4.1. Разработка в среде MS Visual Studio. Графика, анимация.	2	ОПК-7 3-2 ОПК-7 В-1	Индивидуаль-ные задания, практические задания

17.	Тема 4.2. Структура проектов WindowsForm и WPF. Компоновка. Каркас приложения. Модель данных.	2	ОПК-7 У-1 ОПК-7 В-1	Индивидуаль-ные задания, практические задания, тест
18.	Тема 4.3. Интеграция приложений через API.	2	ОПК-7 В-1	Индивидуаль-ные задания, практические задания, тест
19.	Тема 5.1. Создание приложений для Баз данных.	3	ОПК-7 У-1 ОПК-7 В-1	Индивидуаль-ные задания, практические задания, тест

1.3. Описание показателей и критериев оценивания компетенций на различных этапах формирования, описание шкалы оценивания.

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
Знает	ОПК-7 3-1, ОПК-7 3-2, ОПК-7 3-3, ОПК-7 3-4	Отлично	90-100	теоретическое содержание дисциплины освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные рабочей программой дисциплины задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному
Умеет	ОПК-7 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1			
Знает	ОПК-7 3-1, ОПК-7 3-2, ОПК-7 3-3, ОПК-7 3-4	Хорошо	75-89	теоретическое содержание дисциплины освоено полностью, без пробелов; некоторые практические навыки

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
Умеет	ОПК-7 У-1, ОПК-7 У-2			работы с освоенным материалом сформированы недостаточно, все предусмотренные рабочей программой дисциплины задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Владеет	ОПК-7 В-1			
Знает	ОПК-7 З-1, ОПК-7 З-2, ОПК-7 З-3, ОПК-7 З-4	Удовлетворительно	60-74	теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных рабочей программой дисциплины учебных задания выполнено, некоторые из выполненных заданий содержат ошибки
Умеет	ОПК-7 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1, ПК-2 В-1			
Знает	ОПК-7 З-1, ОПК-7 З-2, ОПК-7 З-3, ОПК-7 З-4	Неудовлетворительно	0-59	теоретическое содержание дисциплины не освоено полностью; необходимые

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
Умеет	ОПК-7 У-1, ОПК-7 У-2			практические навыки работы не сформированы, все предусмотренные рабочей программой дисциплины задания выполнены с грубыми ошибками либо совсем не выполнены, качество их выполнения оценено числом баллов, близким к минимальному
Владеет	ОПК-7 В-1			

РАЗДЕЛ 2 Текущий контроль

Текущий контроль знаний используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной работой) обучающихся. В условиях балльно-рейтинговой системы контроля результаты текущего оценивания обучающегося используются как показатель его текущего рейтинга. Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе преподавателя. Данный вид контроля стимулирует у обучающегося стремление к систематической самостоятельной работе по изучению учебной дисциплины.

2.1. Описание оценочных средств по видам заданий текущего контроля

Распределение баллов по рейтинговой системе оценивания
по видам учебной деятельности

Сумма баллов по разделу	Раздел 1		Раздел 2		Сумма баллов за дисциплину 100	
Темы	Т.1.1-1.6	Контроль знаний раздела учебной дисциплины - 5	Т. 2.1 - 2.6	Контроль знаний раздела учебной дисциплины - 5		
Виды работ:						
Практические работы	20		20			20
Индивидуальные задания	20		20			20
Контрольные задания	5		5			
Сумма баллов	45		45			

Сумма баллов по разделу	Раздел 3		Раздел 4		Сумма баллов за дисциплину 100	
Темы	Т.3.1-3.3	Контроль знаний раздела учебной дисциплины - 5	Т. 4.1 - 4.3	Контроль знаний раздела учебной дисциплины - 5		
Виды работ:						
Практические работы	20		20			20
Индивидуальные задания	20		20			20
Контрольные задания	5		5			
Сумма баллов	45		45			

2.1.1. Рекомендации по оцениванию индивидуальных и практических работ обучающихся

Максимальное количество баллов (государственная оценка)	Критерии
10-9 (отлично)	выставляется обучающемуся: если выполнены все пункты работы самостоятельно, без ошибок, если предложен более рациональный алгоритм решения задачи.
8-7 (хорошо)	выставляется обучающемуся: если самостоятельно выполнены все пункты работы, допущены незначительные ошибки, если предложен более рациональный алгоритм решения задачи.
6-5 (удовлетворительно)	выставляется обучающемуся: если самостоятельно (или с помощью преподавателя) выполнены все пункты работы, допущены грубые ошибки.
4 и менее (неудовлетворительно)	выставляется обучающемуся: если с помощью преподавателя выполнены не все пункты работы, допущены грубые ошибки.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

Индивидуальная работа

Раздел №1. Алгоритмизация и программирование на языках высокого уровня.

Тема 1.1. Представление информации в ЭВМ. Состав языка программирования Python

Вариант 1:

1. Написать программу, которая выводит первую букву вашего имени в виде звёздочек, а затем все остальные буквы.

* * * * *

*

*

*

*

аня

2. На вход программе через стандартный ввод передаются два целых числа, по модулю не превышающие 100000. Выведите в стандартный вывод их сумму.

stdin	stdout
2 3	5
-4 -9	-13
-1 2	1

3. Написать программу, которая, реализует диалог с пользователем: запрашивает с клавиатуры два целых числа, и выводит на экран сумму данных чисел.

4. Написать программу, которая подсчитывает площадь равностороннего треугольника, периметр которого равен p .

5. Написать программу, которая определяет наибольшую цифру в натуральном двухзначном числе.

Индивидуальная работа

Раздел № 2. Структуры данных и файлов на Python

Тема 2.2. Парадигмы программирования. Функции как основа процедурного подхода

Вариант 4:

1. Напишите функцию, которая

- называется Factorial
- возвращает int
- принимает int и возвращает факториал своего аргумента. Гарантируется, что аргумент функции по модулю не превышает 10. Для отрицательных аргументов функция должна возвращать 1.

Аргумент функции	Результат
1	1
-2	1
4	24

2. Напишите функцию, которая

- называется IsPalindrom
- возвращает bool
- принимает параметр типа string и возвращает, является ли переданная строка палиндромом

Палиндром - это слово или фраза, которые одинаково читаются слева направо и справа налево.

Аргумент функции Результат

madam	true
gentleman	false

3. Напишите функцию UpdateIfGreater, которая принимает два целочисленных аргумента: first и second. Если first оказался больше second, Ваша функция должна записывать в second значение параметра first. При этом изменение параметра second должно быть видно на вызывающей стороне.

Пример:

```
int a = 4;
int b = 2;
UpdateIfGreater(a, b);
// b должно стать равно 4
```

4. Напишите функцию MoveStrings, которая принимает два вектора строк, source и destination, и дописывает все строки из первого вектора в конец второго. После выполнения функции вектор source должен оказаться пустым.

Чтобы очистить содержимое вектора, надо вызвать у него метод clear:

```
vector<string> words = {"hey", "hi"};
words.clear();
// Теперь вектор words пуст
```

Пример

```
vector<string> source = {"a", "b", "c"};
vector<string> destination = {"z"};
MoveStrings(source, destination);
// source должен оказаться пустым
// destination должен содержать строки "z", "a", "b", "c" именно в таком порядке
```

Практическая работа

Раздел №3. Объектно-ориентированное программирование на C#

Тема 3.1. Введение в объектно-ориентированное программирование. Классы и объекты

Вариант 5:

Создать класс Money, содержащий следующие члены класса:

1. Поля:
 - int first; //номинал купюры
 - int second; //количество купюр

2. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.
 3. Методы, позволяющие:
 - вывести номинал и количество купюр;
 - определить, хватит ли денежных средств на покупку товара на сумму N рублей.
 - определить, сколько штук товара стоимости n рублей можно купить на имеющиеся денежные средства.
 4. Свойство:
 - позволяющее получить-установить значение полей (доступное для чтения и записи);
 - позволяющее рассчитать сумму денег (доступное только для чтения).
 5. Индексатор, позволяющий по индексу 0 обращаться к полю first, по индексу 1 – к полю second, при других значениях индекса выдается сообщение об ошибке.
 6. Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей first и second;
 - операции !: возвращает значение true, если поле second не нулевое, иначе false;
 - операции бинарный +: добавляет к значению поля second значение скаляра.
- Продемонстрировать работу класса.

3.3. Рекомендации по оцениванию устных ответов обучающихся

Оценка «5» - 5 баллов - ставится, если обучающийся:

- 1) полно и аргументировано отвечает по содержанию вопроса;
 - 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике;
 - 3) умеет достаточно глубоко и доказательно обосновать свои суждения и применяемый инструментарий для решения задания;
- Оценка «4» - 4 балла - ставится, если обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.
- Оценка «3» - 3 балла - ставится, если обучающийся обнаруживает знание и понимание основных положений данного задания, но:

- 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;
- 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и применяемый инструментарий для решения задания;
- 3) излагает материал непоследовательно и допускает ошибки.

Оценка «2» - 1-2 балла - ставится, если обучающийся обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает применяемый инструментарий для решения задания. Оценка «2» отмечает такие недостатки в подготовке обучающегося, которые являются серьезным препятствием к успешному овладению последующим материалом.

ТИПОВЫЕ ВОПРОСЫ ДЛЯ САМОПОДГОТОВКИ

Контролируемые разделы (темы) учебной дисциплины	Вопросы, выносимые на самостоятельное обучение по разделам дисциплины
Раздел 1. Информация и программирование	
Тема 1.1. Представление информации в ЭВМ. Системы исчисления	1. Общая характеристика и классификация ЭВМ. 2. Представление информации в ЭВМ. 3. Информация и данные. 4. Двоичная система исчисления 5. Шестнадцатеричная система исчисления 6. Восьмеричная система исчисления

Тема 1.2. Состав языка программирования Python	1. Как записывается и как работает оператор FOR? 2. Для организации каких циклов применим оператор FOR? 3. В чем отличие оператора WHILE от оператора DO..WHILE? 4. Как программируются циклические алгоритмы с явно заданным числом повторений цикла? 5. Как программируются циклические алгоритмы с заранее неизвестным числом повторений цикла? 6. Напишите оператор цикла, который не выполняется ни разу. 7. Напишите оператор цикла, который выполняется неограниченное число раз.
Тема 1.3. Базовые типы данных языка Python	1. Каким образом определяются переменные типа массив (одномерный и двумерный)? 2. Как осуществляется доступ к отдельному элементу одномерного и двумерного массива? 3. Каким образом выводятся элементы массива на экран? 4. Приведите пример фрагмента программы, который выводит на экран двумерный массив в виде матрицы. 5. Как описываются строковые переменные? 6. Какие операции допустимы над строковыми данными? 7. В чем отличие строковой переменной от массива символов? 8. Какие стандартные процедуры и функции для работы со строками вы знаете? 9. Какая встроенная функция библиотеки может быть использована для сравнения двух строк?
Раздел 2. Алгоритмизация и программирование на языках высокого уровня	
Тема 2.1. Алгоритмы и программы	1. Понятие алгоритма. Сложность алгоритма. 2. Как оценить сложность алгоритма? 3. Какие нотации оценки алгоритма вы знаете? 4. Графические способы описания алгоритмов. 5. Рекурсивные алгоритмы.

Тема 2.2. Реализация базовых структур алгоритма в языках программирования.	1. Функции. Вызов функции в программах. Формальные и фактические параметры. Соответствие между параметрами. Примеры. 2. Параметры-переменные, параметры-значения, параметры-константы. Примеры. 3. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры. 4. Область действия переменных и других идентификаторов.
Тема 2.3. Массивы.	1. Статические и динамические переменные. 2. Карта памяти. Динамическое распределение памяти. Создание и уничтожение динамических переменных. 3. Фрагментация динамической памяти. Освобождение динамической памяти. 4. Примеры использования динамической памяти. 5. Принципы структурного программирования. Блочный (модульный) подход к построению алгоритмов решения задач.

2.1.3. Рекомендации по оцениванию тестовых заданий обучающихся

Максимальное количество баллов (государственная оценка)	Критерии
5 (отлично)	Выставляется обучающемуся, если обучающийся представил 90-100% правильных ответов.
4 (хорошо)	выставляется обучающемуся, если обучающийся представил 75-89% правильных ответов.
3 (удовлетворительно)	выставляется обучающемуся, если обучающийся представил 60-74% правильных ответов.
2 и менее (неудовлетворительно)	выставляется обучающемуся, если обучающийся представил менее 59% правильных ответов.

ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ КОМПЕТЕНЦИЙ

Раздел № 2. Структурное программирование на языке Python. Темы 2.1 – 2.6 Тестовые задания

1. Какой оператор используется для вывода данных в Python?

- A) input()
- B) print()
- C) write()
- D) output()

2. Что вернет выражение `10 // 3`?

- A) 3.33
- B) 3
- C) 3.0
- D) 4

3. Какое значение вернет выражение `bool('False')`?

- A) False
- B) True
- C) None
- D) Ошибка

4. Какой из циклов в Python выполняется минимум один раз?

- A) for
- B) while
- C) do...while
- D) В Python нет do...while

5. Чем `input()` отличается от `print()`?

6. Какие существуют способы форматирования строк в Python?

7. Что такое срезы строк? Приведите примеры.

8. Какой результат вернет `range(1, 10, 2)`?

9. Какой тип данных у переменной `x`, если `x = 10.5`?

- A) int
- B) float
- C) str
- D) bool

10. Какое ключевое слово используется для определения функции в Python?

- A) define
- B) function
- C) def
- D) fun

11. Что делает оператор `is`?

- A) Проверяет равенство значений
- B) Проверяет, указывают ли переменные на один объект в памяти
- C) Преобразует переменную в другой тип
- D) Выполняет логическую операцию "и"

12. Какой метод списка удаляет первый найденный элемент по значению?

- A) remove()
- B) pop()
- C) del
- D) discard()

13. Как создать пустой список в Python?

- A) list = { }
- B) list = []
- C) list = None
- D) list = ()

14. Какой результат вернет bool(0)?

- A) True
- B) False
- C) None
- D) Ошибка

Раздел №4. Структуры данных и файлов на C#

Темы 4.1 – 4.3

1. В программе объявлен вектор целых чисел *v* некоторого размера *n*:

`vector<int> v(n);`

Какие из перечисленных операций над вектором *v* могут изменить его размер?

- a) `cout << v[5];` (при условии, что $5 < n$)
- b) `v[5] = 0;` (при условии, что $5 < n$)
- c) `v.assign(1, 0);`
- d) `v.clear();`
- e) `v.resize(1);`
- f) `v.push_back(0);`

2. В программе объявлен вектор целых чисел *v* некоторого размера *n*:

`vector<int> v(n);`

Какие из перечисленных операций над вектором *v* в любом случае изменяют его размер?

- a) `cout << v[5];` (при условии, что $5 < n$)
- b) `v[5] = 0;` (при условии, что $5 < n$)
- c) `v.assign(1, 0);`
- d) `v.clear();`
- e) `v.resize(1);`
- f) `v.push_back(0);`

3. В программе объявлен вектор целых чисел *v* некоторого размера *n*:

`vector<int> v(n);`

Какие из перечисленных операций очищают вектор *v*, то есть делают его длину равной нулю?

- a) `v[0] = 0;` (при условии, что $0 < n$)
- b) `v.assign(0, 8);`

- c) v.clear();
- d) v.resize(0);

4. Метод back позволяет обратиться к последнему элементу любого непустого вектора:
`int last_element = v.back();`

Как можно добиться того же эффекта с помощью квадратных скобок?

- a) `v[v.size() - 1]`
- b) `v[v.size() + 1]`
- c) `v[v.size()]`

5. В переменной m типа `map<string, int>` хранится соответствие значения 8 ключу "Eight". Как удалить это соответствие из словаря (размер словаря при этом должен уменьшиться на 1)?

- a) `m.erase("Eight", 8);`
- b) `m.erase("Eight");`
- c) `m.erase(8);`
- d) `m.count("Eight") = 0;`
- e) `m.count(8) = 0;`
- f) `m["Eight"] = 0;`

6. В программе объявлена переменная m типа `map<int, string>`. Какие операции могут изменить размер словаря m?

- a) `m.size();`
- b) `m.erase(5);`
- c) `string s = m[5];`
- d) `m[5] += 'x';`
- e) `m.count(5);`

7. Функция BuildReversedMap возвращает «перевернутый» словарь, в котором ключами являются значения переданного словаря, а значениями – соответствующие ключи:

```
map<string, bool> BuildReversedMap(const map<bool, string>& m) {
```

```
    map<string, bool> result;
```

```
    for (const auto& item : m) {
```

```
        result[item.second] = item.first; }
```

```
    return result; }
```

Чему может равняться значение выражения `BuildReversedMap(m).size();`

- a) значение выражения может быть больше 3
- b) 0
- c) 1
- d) 2
- e) 3

8. Что выведет данная программа?

```
map<int, string> m = {{1, "odd"}, {2, "even"}, {1, "one"}};
```

```
m[2] = "two";
```

```
m.erase(1);
```

```
m[3] = "three";
```

```
m.erase(4);
```

```
m[5] = "five";
```

```
cout << m.size();
```

а) код не скомпилируется, так как нельзя вызывать метод erase от несуществующего ключа

- b) 5
- c) 8
- d) 10
- e) 4
- f) 6
- g) 3

9. Что выведет следующая программа?

```
set<char> s = {'a', 'b', 'a'};
```

```
cout << s.count('a');
```

- a) 1 или 2 в зависимости от компилятора и системного окружения
- b) 1
- c) 2

10. В программе объявлена переменная s типа set <int>. Всегда ли следующий код оставляет размер множества s неизменным? Переменная x имеет тип int и не изменяется.

```
s.insert(x);
```

```
s.erase(x);
```

- a) нет, размер множества может увеличиться
- b) нет, размер множества может уменьшиться
- c) нет, размер множества может измениться как в бóльшую, так и в меньшую сторону
- d) да, размер множества всегда останется неизменным

11. В программе объявлена переменная s типа set <int>. Всегда ли следующий код оставляет размер множества s неизменным? Переменная x имеет тип int и не изменяется.

```
s.erase(x);
```

```
s.insert(x);
```

- a) нет, размер множества может увеличиться
- b) нет, размер множества может уменьшиться
- c) нет, размер множества может измениться как в бóльшую, так и в меньшую сторону
- d) да, размер множества всегда останется неизменным.

12. Что выведет следующая программа?

```
set <int> s = {4, 2, 1, 2};
```

```
s.erase(2);
```

```
s.insert(1);
```

```
cout << s.size();
```

- a) 1
- b) 2
- c) 3
- d) 4

ВОПРОСЫ К ЭКЗАМЕНУ(1 семестр)

1. Что представляют собой следующие понятия: данные, информация, система, знания.
2. Дайте определение software engineering и в чем состоит фундаментальная идея технологии программирования.
3. Основные категории языков программирования. Этапы создания *.exe файла.
Структура программы на PYTHON
4. Стандартные библиотеки PYTHON
5. Ввод и вывод в PYTHON
6. Математические вычисления в PYTHON
7. Алфавит и лексемы PYTHON
8. Операторы и операции.
9. Фундаментальные типы: логический тип, символьные типы, целые типы, типы с плавающей точкой.
10. Тип void.
11. Указатели.
12. Массивы.
13. Ссылки.
14. Объявления и определения.
15. Константы.
16. Область видимости.
17. Инициализация.
18. Указатели и адресная арифметика.
19. Операции над указателями.
20. Стандартные операторы PYTHON
21. Объявления и определения функций.
22. Аргументы функций.
23. Типы функций.
24. Вызовы функций.
25. Аргументы по умолчанию.
26. Неуказанное количество аргументов.
27. Встраивание функций.
28. Перегрузка функций.
29. Объявления списков.

Пример практического задания

Дан список из N целых чисел. Написать программу нахождения суммы положительных элементов списка с нечетными индексами.

ВОПРОСЫ К ЗАЧЁТУ(2 семестр)

1. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
2. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.
3. Функции, ориентированные на работу со строками. Примеры.
4. Процедуры, ориентированные на работу со строками. Примеры.
5. "Пузырьковая" сортировка (обменами).
6. Сортировка выбором.

7. Сортировка вставками.
8. Примеры рекурсивных функций: количество цифр в записи данного натурального числа; поиск минимального элемента в массиве; проверка, является ли число простым.
9. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.
10. Оценка сложности алгоритмов. О-нотация.
11. Модульные тесты. Внедрение тестирования.
12. Список List в C#
13. Словарь Dictionary в C#
14. Файлы и каталоги
15. Однонаправленные списки и действия с ними.
16. Двухнаправленные списки.
17. Кольцевые списки. Списки с ключом.
18. Стек.
19. Очередь.
20. Дек.
21. Двоичные деревья. Формирование дерева. Добавление элемента в дерево. Печать дерева. Способы обхода дерева.
22. Двоичные деревья. Исключение компонента из дерева.

Пример практического задания

Написать функцию dayStatus принимающую номер дня недели (целое число от 1 до 7) и возвращающую true, если день рабочий или false, если выходной.

ВОПРОСЫ К ЭКЗАМЕНУ(3 семестр)

1. ВОПРОСЫ К ЭКЗАМЕНУ(3 семестр)

1. Понятие об ООП. Отличие ООП подхода к программированию от структурного.
2. Основные понятия ООП. Примеры.
3. Дать определение класса
4. Дать определение экземпляра класса
5. Дать определение интерфейса
6. Чем свойства отличаются от полей
7. Как в C# реализуется множественное наследование
8. Перегрузка методов
9. Виртуальные классы и виртуальные методы
10. Что такое индексы классов?
11. Для чего создают абстрактные классы?
12. Дать определение 4 концепциям ООП
13. Перегрузка операций, особенности реализации в C#
14. Модификаторы доступа
15. Реализация полиморфизма в C#
16. Ключевое слово sealed
17. Как в приложениях Windows Forms интерфейс отделён от бизнес-логики
18. Какие управляющие графические элементы вы знаете?
19. Как сделать элемент невидимым, и как недоступным?
20. С помощью какого элемента можно имитировать временные задержки?
21. Из каких основных файлов состоит проект Windows Forms?
22. От какого класса наследуются все классы в C#?
23. Назовите отличия между интерфейсом и абстрактным классом?

24. Чем отличается event от delegate?
25. Инкапсуляция. Примеры.
26. Наследование. Примеры.
27. Полиморфизм. Примеры.
28. Что такое комбинированный тип данных (запись)? Способы описания (примеры).
29. Способы обращения к компонентам записи.
30. Оператор присоединения и его использование.
31. Записи с вариантами. Примеры.
32. Понятие множества. Обозначение множества в C#
33. Задание множественного типа и множественная переменная.
34. Операции над множествами. Примеры.
35. Понятие файла. Логические и физические файлы.
36. Файловые типы C++ (общие понятия).
37. Понятие текущей длины файла, указателя или окна файла, буфера файла.
38. Общие процедуры и функции для работы с файлами.
39. Компонентные (типизированные) файлы: описание, отличие от других видов файлов.
40. Текстовые файлы, принципы работы. Процедуры и функции, ориентированные на работу с текстовыми файлами, ввод-вывод информации.
41. Нетипизированные файлы и процедуры ввода-вывода. Примеры использования.
42. Прямой и последовательный доступ к компонентам файлов.
43. Работа с файлами записей. Примеры.
44. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.

ВОПРОСЫ К ЭКЗАМЕНУ(4 семестр)

1. Как в приложениях Windows Forms интерфейс отделён от бизнес-логики
2. Какие управляющие графические элементы вы знаете?
3. Как сделать элемент невидимым, и как недоступным?
4. С помощью какого элемента можно имитировать временные задержки?
5. Из каких основных файлов состоит проект Windows Forms?
6. Какова структура проекта WPF?
7. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.
8. Построение диаграмм.
9. Построение графиков функций.
10. Технология создания приложений для БД.
11. Обработка исключений. Основные классы исключений в C#.
12. Оценка сложности алгоритмов. О-нотация.
13. Модульные тесты. Внедрение тестирования.
14. Список List в C#
15. Словарь Dictionary в C#
16. Файлы и каталоги
17. Однонаправленные списки и действия с ними.
18. Двухнаправленные списки.
19. Кольцевые списки. Списки с ключом.
20. Стек.
21. Очередь.
22. Дек.
23. Двоичные деревья. Формирование дерева. Добавление элемента в дерево. Печать дерева. Способы обхода дерева.
24. Двоичные деревья. Исключение компонента из дерева.

25. Библиотека STL. Классы контейнеры d C++.
26. Работа с классом Vector библиотеки STL
27. Работа с классом Map библиотеки STL
28. Работа с классом Dictionary библиотеки STL
29. Работа с удаленной БД из приложения Window Forms
30. Подсистема Graphic Device Interface
31. System.Drawing и System.Drawing.Drawing2D.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ»

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Прикладная информатика в управлении корпоративными информационными системами»

Кафедра информационных технологий

Учебная дисциплина «Информатика и программирование»

Курс 1 Семестр 2 Форма обучения очная

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

Теоретические вопросы.

1. Основные принципы объектного подхода. Абстрагирование. Класс Object.
2. Понятие контейнера. Работа с классом Vector библиотеки STL.
3. **Практическое задание.**

Построить класс Прямоугольный треугольник. Параметры класса: координаты 3-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура прямоугольным треугольником; вычисления и вывод сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N треугольников, найти количество треугольников, у которых площадь больше средней площади.

Экзаменатор: _____

Утверждено на заседании кафедры « » _____ 20__ г. (протокол № от « »
_____ 20__ г.)

Зав.кафедрой: _____

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ»

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Прикладная информатика в управлении корпоративными информационными системами»

Кафедра информационных технологий

Учебная дисциплина «Информатика и программирование»

Курс 2 Семестр 4 Форма обучения очная

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

Теоретические вопросы.

1. Работа с MS SQL из приложения Window Forms.
2. Обработка исключений. Основные классы исключений в C#.
3. **Практическое задание.**

Построить класс Прямоугольный треугольник. Параметры класса: координаты 3-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура прямоугольным треугольником; вычисления и вывод сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N треугольников, найти количество треугольников, у которых площадь больше средней площади.

Экзаменатор: _____

Утверждено на заседании кафедры «____» _____ 20__ г. (протокол №____ от «____» _____ 20__ г.)

Зав.кафедрой: _____