

Рабочая программа дисциплины: разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО 3++) по направлению подготовки 01.03.04 Прикладная математика, утвержденного приказом МИНОБРНАУКИ РФ от 10 января 2018 года № 11 на основании учебного плана, принятого УС ДПИ НГТУ

протокол от 05.06.2024 № 10

Рабочая программа одобрена на заседании кафедры-разработчика РПД Автоматизация, энергетика, математика и информационные системы
протокол от 10.06.2024 № 7

Зав. кафедрой к.т.н, доцент

_____ Л.Ю. Вадова
(подпись)

СОГЛАСОВАНО:

Заведующий выпускающей кафедрой Автоматизация, энергетика, математика и информационные системы
к.т.н, доцент

_____ Л.Ю. Вадова
(подпись)

Начальник ОУМБО

_____ И.В. Старикова
(подпись)

Рабочая программа зарегистрирована в ОУМБО: 01.03.04 - 36

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	4
3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	4
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
5. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	14
6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	22
7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	22
8. ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ	242
9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....	24
10. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	25
11. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	29

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1. Цель освоения дисциплины:

Целью освоения дисциплины является практическое освоение методов объектно-ориентированного программирования и приобретение навыков их применения при анализе и управлении современными техническими системами.

1.2. Задачи освоения дисциплины (модуля):

- знание основных методов объектно-ориентированного программирования и их возможностей для решения сложных инженерных задач.
- применение методов объектно-ориентированного программирования для решения профессиональных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Учебная дисциплина Объектно-ориентированное программирование включена в перечень дисциплин вариативной части (формируемой участниками образовательных отношений), определяющий направленность ОП. Дисциплина реализуется в соответствии с требованиями ФГОС, ОП ВО и УП.

Дисциплина базируется на следующих дисциплинах: Технология программирования.

Дисциплина Объектно-ориентированное программирование является основополагающей для изучения следующих дисциплин: Многопоточное программирование; Программирование для Интернет; Архитектура ЭВМ и системное ПО.

Рабочая программа дисциплины Объектно-ориентированное программирование для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся, по их личному заявлению.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Таблица 1 – Формирование компетенции ПКС-1 дисциплинами

Наименование дисциплин, формирующих компетенцию совместно	Семестры формирования компетенций дисциплинами.							
	1	2	3	4	5	6	7	8
Код компетенции ПКС-1								
Объектно-ориентированное программирование								
Технологии программирования								
Ознакомительная практика								
Проектно-технологическая практика								
Многопоточное программирование								
Программирование для Интернет								
Архитектура ЭВМ и системное ПО								
Преддипломная практика								
Выполнение и защита выпускной квалификационной работы								

ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОП

Таблица 2

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

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине			Оценочные средства	
					Текущего контроля	Промежуточной аттестации
ПКС-1. Способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования	ИПКС-1.1 Использует методы и инструменты современных объектно-ориентированных языков программирования для разработки компонентов программного обеспечения	Знать: базовые понятия объектно-ориентированного программирования, основные методы и приемы разработки объектно-ориентированных программ.	Уметь: разрабатывать алгоритмы и программы на языке программирования С#, проводить отладку и тестирование программы.	Владеть: основными теоретическими знаниями и эвристическими методами, применяемыми при объектно-ориентированном подходе к разработке компьютерных программ.	Тестирование в системе MOODLE. (2 тестирования, в базе каждого тестирования 100- 110 вопросов), выполнение 12 контрольных работ (по 10 вариантов в каждой контрольной работе)	Вопросы для устного собеседования: билеты (20 билетов)

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 10 зач.ед./360 часов, распределение часов по видам работ семестрам представлено в табл. 3 и 4.

Формат изучения дисциплины: с использованием элементов электронного обучения.

Таблица 3 – Распределение трудоёмкости дисциплины по видам работ по семестрам
Для очной формы обучения

Вид учебной работы	Всего часов	Семестры		
		2	3	4
1. Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего), в том числе:	146	71	36	39
1.1. Аудиторные занятия (всего), в том числе:	136	68	34	34
- лекции (Л)	68	34	17	17
- лабораторные работы (ЛР)	68	34	17	17
- практические занятия (ПЗ)				
- практикумы (П)				
1.2. Внеаудиторные занятия (всего), в том числе:	10	3	2	5
- групповые консультации по дисциплине	6	2	2	2
- групповые консультации по промежуточной аттестации (зачет, экзамен)	2	1		1
- курсовые работы	2			2
2. Самостоятельная работа студента (СРС) (всего)	133	64	54	15
Вид промежуточной аттестации зачет, экзамен	81	45 экз.	зач.	36 экз.
Общая трудоемкость, часы/зачетные единицы	360/10	180/5	72/2	144/4

4.2 Содержание дисциплины, структурированное по темам

Таблица 4 – Содержание дисциплины, структурированное по темам для обучающихся очной формы обучения

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
2 семестр									
Раздел 1 Основы платформы .NET Framework и среды Visual Studio .NET.									
ПКС-1, ИПКС-1.1.	Тема 1.1. Платформа разработки .NET Framework. Введение. Главные компоненты .NET. Библиотека базовых классов. Языки программирования с поддержкой .NET. Сборки. Общая система типов. Общезыковые спецификации. Общезыковая среда выполнения. Различия между сборками, пространствами имен и типами. Объектно-ориентированное программирование. Понятие класса.	5			8	Подготовка к лекциям и лабораторным занятиям (6.1.2: стр.12-75, 123-158)	Чтение основной и дополнительной литературы, рекомендованной по курсу Оформление отчетов по лабораторным работам Выполнение курсовой работы Подготовка к промежуточной аттестации		
	Тема 1.2. Интегрированная среда разработки Visual Studio .NET Консольные приложения. Создание проекта. Основные окна среды Visual Studio .NET. Заготовка консольной программы. Компиляция программы и запуск ее на выполнение.	5	4		8				

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
Итого по разделу 1		10	4		16				
Раздел 2 Язык программирования C#.									
Тема 2.1. Основные конструкции языка C#. Элементы языка. Алфавит: буквы и цифры, пробельные символы, разделители, специальные (управляющие) символы или escape-последовательности. Константы (литералы): логические, целые, вещественные, символьные, строковые, константа null. Идентификаторы. Ключевые (служебные) слова. Знаки операций. Комментарии. Структура программы. Типы данных. Классификация типов данных. Встроенные типы данных. Типы-значения и ссылочные типы. Упаковка и распаковка. Переменные, операции и выражения. Переменные. Именованные константы. Операции и выражения. Преобразования встроенных арифметических типов-значений. Основные операции C#. Приоритет и ассоциативность операций. Некоторые первичные операции: new, операция приведения типа. Арифметические операции, операции отношения, логические операции, поразрядные операции, условная операция.		5	5		8	Подготовка к лекциям и лабораторным занятиям (6.1.1: стр.9-240); (6.1.2: стр. 105-120, 158-203); (6.1.3: стр. 8-234); (6.1.4: стр. 12-89)	Чтение основной и дополнительной литературы, рекомендованной по курсу Оформление отчетов по лабораторным работам Выполнение курсовой работы Подготовка к промежуточной аттестации		

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	операции присваивания. Математические функции – класс Math.								
	Тема 2.2. Основы ввода-вывода данных. Консольный ввод-вывод. Ввод-вывод в файлы.	3	5		8				
	Тема 2.3. Операторы Пустой оператор. Составной оператор (блок). Операторы ветвления if, switch. Операторы цикла for, while, do, foreach. Операторы передачи управления goto, break, continue, return.	3	5		8				
	Тема 2.4. Классы. Основные понятия. Члены класса. Определение класса. Данные: поля и константы. Методы. Параметры-значения, параметры-ссылки, выходные параметры. Методы с переменным числом параметров. Ключевое слово this. Конструкторы. Свойства.	4	5		8				
	Тема 2.5. Массивы и строки. Массивы. Одномерные, прямоугольные, ступенчатые массивы. Класс System.Array. Символы и строки. Символы. Массивы символов. Строки типа string. Форматирование строк. Строки типа StringBuilder. Класс Random.	4	5		8				
	Тема 2.6. Более подробно о классах.	5	5		8				

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Унарные операции. Бинарные операции. Операции преобразования типа. Деструкторы. Вложенные типы.								
	Итого по разделу 2	24	30		48				
	Итого во 2 семестре	34	34		64				
3 семестр									
Раздел 2 Язык программирования C#.									
	Тема 2.7. Иерархии классов. Наследование. Виртуальные методы. Абстрактные классы. Бесплодные классы. Класс object.	2	2		7	Подготовка к лекциям и лабораторным занятиям (6.1.1: стр.9-240); (6.1.2: стр. 105-120, 158-203); (6.1.3: стр. 8-234); (6.1.4: стр. 12-89)	Чтение основной и дополнительной литературы, рекомендованной по курсу Оформление отчетов по лабораторным работам Выполнение курсовой работы Подготовка к		
	Тема 2.8. Интерфейсы и структурные типы. Синтаксис интерфейса. Реализация интерфейса. Работа с объектами через интерфейсы. Операции is и as. Интерфейсы и наследование. Стандартные интерфейсы .NET. Сравнение объектов (интерфейс IComparable). Сортировка (интерфейс IComparer). Перегрузка операций отношения. Клонирование объектов (интерфейс ICloneable).	4	3		8				

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	Перебор объектов (интерфейс IEnumerable) и итераторы. Структуры. Перечисления. Операции с перечислениями: арифметические, отношения, определение размера.					промежуточной аттестации			
	Тема 2.9. Обработка исключительных ситуаций Исключения. Операторы try, throw. Класс Exception. Операторы checked и unchecked.	1	2		7				
	Тема 2.10. Делегаты, события и потоки выполнения. Делегаты. Объявление делегатов. Использование делегатов. Передача делегатов в методы. Обработка исключений при вызове делегатов. События. Многопоточные приложения. Класс Thread. Асинхронные делегаты.	2	2		8				
	Тема 2.11. Работа с файлами. Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Консольный ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).	1	2		8				
	Тема 2.12. Сборки, библиотеки, атрибуты, директивы. Сборки. Создание библиотеки. Использование библиотеки. Рефлексия.	4	3		8				

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	Атрибуты. Пространства имен. Директивы препроцессора.								
	Тема 2.13. Структуры данных, коллекции и классы-прототипы. Абстрактные структуры данных. Пространство имен System.Collections. Класс ArrayList. Классы-прототипы. Создание класса-прототипа. Обобщенные методы. Частичные типы. Обнуляемые типы.	3	3		8				
	Итого по разделу 2	17	17		54				
	Итого в 3 семестре	17	17		36				
4 семестр									
Раздел 3 Программирование в Windows									
	Тема 3.1 Графический интерфейс пользователя	1	1		1	Подготовка к лекциям и лабораторным занятиям (6.1.1: стр. 241-380); (6.1.2: стр. 218-232,); (6.1.3: стр. 235-451); (6.1.4: стр. 92-248)	Чтение основной и дополнительной литературы, рекомендованной по курсу Оформление отчетов по лабораторным работам Выполнение		
	Тема 3.2 Многозадачность операционной системы Windows	1	1		1				
	Тема 3.3 Управление памятью в ОС Windows	2	2		1				
	Тема 3.4 Библиотеки динамической компоновки	2	2		1				
	Тема 3.5 Windows – система, управляемая сообщениями	2	2		1				
	Тема 3.6 Интерфейс прикладного программирования Win API	2	2		2				

Планируемые (контролируемые) результаты освоения: ПКС и индикаторы достижения компетенций	Наименование тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках практической подготовки (трудоемкость в часах)	Наименование разработанного электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	Тема 3.7 Разработка Windows-приложений	2	2		2	курсовой работы Подготовка к промежуточной аттестации			
	Тема 3.8 Независимость Windows от аппаратных средств	2	2		2				
	Тема 3.9 Программирование при использовании Win API	2	2		2				
	Тема 3.10 Приложения Windows Forms	1	1		2				
	Итого по разделу 3	17	17		15				
	Итого в 4 семестре	17	17		15				
	ИТОГО ПО ДИСЦИПЛИНЕ	68	68		133				

5. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.

5.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

1) Примерная тематика лабораторных работ:

Базовые типы данных. Начальные сведения о вводе-выводе.

Линейные программы.

Разветвляющиеся вычислительные процессы.

Организация циклов.

Работа с целыми числами.

Сложные циклические вычислительные процессы.

Одномерные массивы.

Прямоугольные массивы.

Ступенчатые массивы.

Массивы. Методы, параметры методов.

Строки типа string.

Строки типа StringBuilder.

Перечисления.

Структуры.

Классы. Поля, конструкторы, методы, свойства.

Классы. Перегрузка методов и операций, индексаторы.

Классы. Наследование.

Классы. Интерфейсы.

Классы. Делегаты и события.

Многопоточные приложения.

Создание и использование библиотек.

Класс ArrayList.

Пример заданий к лабораторной работе:

Текст программы (файл simple.c)

```
/*
  ФИО, группа
  Лабораторная работа №1
  Простейшая программа
  Вариант №0
*/
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

void main()
{
  int i;
  float a,b;
  double c;
  FILE *fout;

  setlocale(LC_CTYPE, "Russian");

  fout=fopen("D:\\simple.out","w");
  if(fout==NULL)
  { printf("Невозможно открыть файл \\\"simple.out\\\"\\n"); exit(1); }

  i=5;
  a=1.; b=1.25;
  c=5.99;

  printf("Целое значение i = %d\\n",i);
  fprintf(fout,"Целое значение i = %d\\n",i);
  printf("Веществ. значения типа float a = %f b = %f\\n",a,b);
  fprintf(fout,"Веществ. значения типа float a = %f b = %f\\n",a,b);
  a += b; /* или a=a+b; */
  printf("Веществ. значение типа float суммы a+b = %f\\n",a);
  fprintf(fout,"Веществ. значение типа float суммы a+b = %f\\n",a);
  printf("Веществ. значение типа double c = %lf\\n",c);
  fprintf(fout,"Веществ. значение типа double c = %lf\\n",c);

  fclose(fout);
}
```

Результаты выполнения программы (файл simple.out)

```
Целое значение i = 5
Веществ. значения типа float a = 1.000000 b = 1.250000
Веществ. значение типа float суммы a+b = 2.250000
Веществ. значение типа double c = 5.990000
```

2) Тесты для промежуточного контроля знаний обучающихся

Тесты, проводимые на электронной платформе Moodle на сайте ДПИ НГТУ по адресу: <http://dpingtu.ru/Moodle>

Включают ответы на теоретические вопросы и решение задач по темам курса с выбором правильного варианта ответа.

3) Вопросы для подготовки к контрольным мероприятиям (текущий контроль в форме устного опроса):

1. Составить программу для вычисления высот треугольника по известным сторонам a, b, c .
2. Составить программу для вычисления периметра и площади прямоугольного треугольника, у которого длина одного катета в два раза больше длины другого, а длина гипотенузы равна c .
3. Составить программу для вычисления периметра и площади равнобедренного треугольника, у которого длина основания равна a , а длина боковой стороны в два раза больше длины высоты, опущенной на основание.
4. Составить программу для вычисления периметра и радиусов вписанной r и описанной R окружностей прямоугольного треугольника по его известным катетам a и b .
5. Составить программу для вычисления периметра и площади равнобедренного треугольника, у которого длина основания равна a , а угол при основании в два раза меньше угла при вершине.
6. Составить программу для вычисления периметра и площади прямоугольного треугольника, у которого длина одного катета в два раза меньше длины другого, а радиус окружности, описанной около треугольника, равен R .
7. Составить программу для вычисления периметра и площади равнобедренного треугольника, у которого длина боковой стороны равна b , а длина основания в два раза больше длины высоты, опущенной на основание.
8. Составить программу для вычисления периметра и площади прямоугольного треугольника, у которого длина одного катета в два раза меньше длины гипотенузы, а длина гипотенузы равна c .
9. Составить программу для вычисления длин сторон равнобедренного треугольника, площадь которого равна S , а угол при основании в два раза меньше угла при вершине.
10. Составить программу для вычисления длин сторон прямоугольного треугольника, у которого длина одного катета в два раза больше длины другого, а площадь равна S .
11. Составить программу для вычисления периметра и площади равнобедренного треугольника, у которого длина боковой стороны равна b , а длина основания в два раза больше длины высоты, опущенной на боковую сторону.
12. Составить программу для вычисления медиан треугольника по его известным сторонам a, b, c .

$$m_a^2 = \frac{b^2 + c^2}{2} - \left(\frac{a}{2}\right)^2$$

13. Составить программу для вычисления биссектрис треугольника по его известным сторонам a, b, c .

$$l_a^2 = b \cdot c \cdot \left(1 - \left(\frac{a}{b+c}\right)^2\right)$$

14. Составить программу для вычисления все углов треугольника по его известным сторонам a, b, c . (значения углов получить в радианах и градусах).
15. Составить программу для вычисления радиусов вписанной r и описанной R окружностей треугольника по его известным сторонам a, b, c .
16. Составить программу для вычисления высот треугольника по его известным сторонам a, b, c .
17. Составить программу для вычисления длины стороны c треугольника и его углов α, β по известным сторонам a, b и углу γ между этими сторонами (угол γ задан в градусах).
18. Составить программу для вычисления периметра и площади треугольника по известным сторонам a, b и углу γ между ними (угол γ задан в градусах).
19. Составить программу для вычисления радиусов вписанной r и описанной R окружностей треугольника по его известным сторонам a, b и углу γ между ними (угол γ задан в градусах).
20. Составить программу для вычисления медиан треугольника по его известным сторонам a, b и углу γ между ними (угол γ задан в градусах).

$$m_a^2 = \frac{b^2 + c^2}{2} - \left(\frac{a}{2}\right)^2$$

21. Составить программу для вычисления высот треугольника по его известным сторонам a , b и углу γ между ними (угол γ задан в градусах).
22. Составить программу для вычисления биссектрис треугольника по его известным сторонам a , b и углу γ между ними (угол γ задан в градусах).

$$l_a^2 = b \cdot c \cdot \left(1 - \left(\frac{a}{b+c} \right)^2 \right)$$

23. Составить программу для вычисления длин сторон b , c и угла α треугольника по его известной стороне a и двум прилежащим к этой стороне углам β и γ (углы β и γ заданы в градусах).
24. Составить программу для вычисления периметра и площади треугольника по его известной стороне a и двум прилежащим к этой стороне углам β и γ (углы β и γ заданы в градусах).
25. Составить программу для вычисления радиусов вписанной r и описанной R окружностей треугольника по его известной стороне a и двум прилежащим к этой стороне углам β и γ (углы β и γ заданы в градусах).
26. Составить программу для вычисления периметра и площади треугольника, а также радиуса описанной R окружности по известным сторонам треугольника a , b , c .
27. Составить программу для вычисления площади S , радиусов вписанной r и описанной R окружностей равностороннего треугольника по его известной стороне a .
28. Составить программу для вычисления периметра и площади прямоугольного треугольника по известному катету a и острому углу α (угол α задан в градусах).
29. Составить программу для вычисления периметра, площади и острых углов α , β прямоугольного треугольника по его известным катетам a , b (значения углов получить в радианах и градусах).
30. Составить программу для вычисления периметра и площади прямоугольного треугольника по его известной гипотенузе c и острому углу α (угол α задан в градусах).
31. Составить программу для вычисления периметра и площади равностороннего треугольника, радиус вписанной окружности которого равен r .
32. Составить программу для вычисления длин сторон равнобедренного треугольника, площадь которого равна S , а угол при основании в два раза меньше угла при вершине.

4) Перечень вопросов, выносимых на промежуточные аттестации

ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ к экзаменам и зачету:

1. Основы платформы разработки .NET Framework. Главные компоненты .NET. Библиотека базовых классов. Сборки. Общая система типов. Общеязыковые спецификации. Общеязыковая среда выполнения.
2. Структура программы на языке C#. Алфавит. Константы: логические, целые, вещественные, символьные, строковые, константа null.
3. Структура программы на языке C#. Идентификаторы. Знаки операций. Комментарии.
4. Типы данных. Встроенные типы данных. Типы-значения и ссылочные типы. Упаковка и распаковка.
5. Переменные. Именованные константы. Преобразования встроенных арифметических типов-значений. Основные операции C#. Приоритет и ассоциативность операций.
6. Обработка исключительных ситуаций. Операторы try, throw. Класс Exception. Операторы checked и unchecked.
7. Классы: основные понятия. Члены класса. Определение класса. Присваивание и сравнение объектов. Данные: поля и константы. Методы.
8. Классы: основные понятия. Ключевое слово this. Конструкторы. Свойства.
9. Массивы. Одномерные, прямоугольные, ступенчатые массивы. Класс System.Array. Оператор foreach. Массивы объектов.

10. Символы и строки. Символы. Массивы символов. Строки типа `string`. Форматирование строк. Строки типа `StringBuilder`. Класс `Random`.
11. Классы: подробности. Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод `Main`. Индексаторы.
12. Классы: подробности. Операции класса. Унарные операции `+` `-` `!` `~` `++` `--` `true` `false`. Бинарные операции `+` `-` `*` `/` `%` `^` `<<` `>>` `==` `!=` `<` `>` `<=` `>=`. Операции преобразования типа. Деструкторы. Вложенные типы.
13. Иерархии классов. Наследование. Виртуальные методы.
14. Иерархии классов. Абстрактные классы. Бесплодные классы. Класс `object`.
15. Интерфейсы. Синтаксис интерфейса. Реализация интерфейса. Работа с объектами через интерфейсы. Операции `is` и `as`. Интерфейсы и наследование.
16. Интерфейсы. Стандартные интерфейсы `.NET`. Сравнение объектов (интерфейс `IComparable`). Сортировка (интерфейс `IComparer`). Перегрузка операций отношения. Клонирование объектов (интерфейс `ICloneable`). Перебор объектов (интерфейс `IEnumerable`).
17. Структуры.
18. Перечисления. Операции с перечислениями: арифметические, поразрядные, определение размера. Базовый класс `System.Enum`.
19. Делегаты. Объявление делегатов. Использование делегатов. Передача делегатов в методы. Обработка исключений при вызове делегатов.
20. События. Многопоточные приложения. Класс `Thread`. Асинхронные делегаты.
21. Работа с файлами. Потоки байтов. Асинхронный ввод-вывод.
22. Работа с файлами. Потоки символов. Двоичные потоки.
23. Консольный ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).
24. Сборки. Создание библиотеки. Использование библиотеки.
25. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.
26. Абстрактные структуры данных. Пространство имен `System.Collections`. Класс `ArrayList`.
27. Классы-прототипы. Создание класса-прототипа. Обобщенные методы. Частичные типы. Обнуляемые типы.

5) Задания для выполнения курсовой работы

1. Создать класс `Complex` для выполнения с комплексными числами $a + bi$ операций сложения, вычитания, умножения, деления, возведение в степень, нахождение модуля и вывода комплексных чисел в виде (a, b) . Составить программу для тестирования класса.
2. Создать класс `Complex` для выполнения с комплексными числами $re^{i\varphi}$ операций сложения, вычитания, умножения, деления, возведение в степень, нахождение модуля и вывода комплексных чисел в виде (r, φ) . Составить программу для тестирования класса.
3. Создать класс `RationalNumber` для выполнения с дробями вида $\frac{p}{q}$ операций сложения, вычитания, умножения, деления, сокращения дробей (результат операции должен быть сохранен в несокращенном виде) и вывода рационального числа в виде $\frac{p}{q}$ и в формате с плавающей точкой. Составить программу для тестирования класса.
4. Создать класс `Time` для установки и вывода времени в виде «часы : минуты : секунды». Попытка установить элемент данных (часы, минуты или секунды) в некорректное состояние должно приводить к установке этого элемента в нулевое значение. В класс `Time` включить метод `tick`, который увеличивает на 1 секунду время, хранимое в объекте `Time`. Составить программу для тестирования метода `tick` в цикле, который выводит время на каждом шаге цикла. Проверить случаи:
 - переход при увеличении времени к следующей минуте;
 - переход при увеличении времени к следующему часу;

- переход при увеличении времени к следующему дню.
5. Создать класс Rect для выполнения над прямоугольниками, лежащими в плоскости (x,y) , операций по изменению размеров прямоугольника, перемещения на плоскости, вращения на заданный угол, вычисления площади. Составить программу для тестирования класса.
 6. Создать класс Matrix для выполнения над симметричными матрицами операций сложения, вычитания, умножения на число, умножения матриц, сравнение на равенство и на неравенство, вычисление нормы, вывод матрицы. Составить программу для тестирования класса.
 7. Создать класс для работы с восьмеричным числом, хранящимся в виде строки символов. Реализовать операции присваивания, сравнения, преобразование в десятичное число, форматный вывод, доступ к заданной цифре числа по индексу. Составить программу для тестирования класса.
 8. Создать класс Triangle для выполнения над треугольниками, лежащими в плоскости (x,y) , операции перемещения на плоскости и операций отношения (мера сравнения – площадь треугольника). Составить программу для тестирования класса.
 9. Создать класс Vector для выполнения над векторами операций сложения, вычитания, умножения на число, нахождение нормы, скалярное произведение векторов, косинуса угла между векторами, вывод вектора. Составить программу для тестирования класса.
 10. Создать класс Polynom для выполнения над многочленами одной переменной операции вычисления значения многочлена для заданного аргумента, операций сложения, вычитания и умножения многочленов с получением нового объекта – многочлена, печать (вывод на экран) описания многочлена. Многочлен задается степенью и массивом коэффициентов. Составить программу для тестирования класса.

5.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости обучающихся очной формы. Основные требования балльно-рейтинговой системы по дисциплине и шкала оценивания приведены в таблицах 5 - 7.

Таблица 5 – Требования балльно-рейтинговой системы по дисциплине

во 2 семестре

Виды работ	Количество подвидов работы	Максимальные баллы за подвид работы	Сроки выполнения подвидов работы	Дополнительные баллы за подвид работы		Штрафные баллы за подвид работы	
				За своевремен. выполнение	За качество	За нарушение сроков	За качество
Выполнение лабораторных работ	8	80/8=10	В течение 2-х недель после выдачи задания	0	0	до -2	до -6
Посещение занятий (лекции+лаб. работы)	34+34=68	20/68≈0,294	Согласно расписанию	0	0	0	0

в 3 семестре

Виды работ	Количество подвидов работы	Максимальные баллы за подвид работы	Сроки выполнения подвидов работы	Дополнительные баллы за подвид работы		Штрафные баллы за подвид работы	
				За своевремен. выполнение	За качество	За нарушение сроков	За качество
РГР	1	10	до 07.12.2021	0	0	до -2	до -6
Выполнение лабораторных работ	8	80/8=10	В течение 2-х недель после выдачи задания	0	0	до -4	до -6
Посещение занятий (лекции + лаб. работы)	17+17=34	10/34≈0,294	Согласно расписанию	0	0	0	0

в 4 семестре

Виды работ	Количество подвидов работы	Максимальные баллы за подвид работы	Сроки выполнения подвидов работы	Дополнительные баллы за подвид работы		Штрафные баллы за подвид работы	
				За своевремен. выполнение	За качество	За нарушение сроков	За качество
Выполнение лабораторных работ	8	80/8=10	В течение 2-х недель после выдачи задания	0	0	до -2	до -6
Посещение занятий (лекции+лаб. работы)	17+34=51	20/51≈0,392	Согласно расписанию	0	0	0	0

Таблица 6 – Критерии оценивания результата обучения по дисциплине и шкала оценивания

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Критерии оценивания результатов обучения			
		Оценка «неудовлетворительно» / «не зачтено» 0-54% от тах рейтинговой оценки контроля	Оценка «удовлетворительно» / «зачтено» 55-70% от тах рейтинговой оценки контроля	Оценка «хорошо» / «зачтено» 71-85% от тах рейтинговой оценки контроля	Оценка «отлично» / «зачтено» 86-100% от тах рейтинговой оценки контроля
ПКС-1. Способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования	ИПКС-1.1 Использует методы и инструменты современных объектно-ориентированных языков программирования для разработки компонентов программного обеспечения	Изложение учебного материала бессистемное, неполное, не знает основ объектно-ориентированного программирования, не может использовать методы объектно-ориентированного программирования в рамках поставленных целей и задач, что препятствует усвоению последующего материала	Фрагментарные, поверхностные знания по основам объектно-ориентированного программирования. Изложение полученных знаний неполное, однако это не препятствует усвоению последующего материала. Допускаются отдельные существенные ошибки, исправленные с помощью преподавателя. Затруднения при формулировании основных положений и их применении	Знает материал на достаточно хорошем уровне; представляет основные задачи в рамках постановки целей и выбора оптимальных способов их достижения.	Имеет глубокие знания всего материала, понимает структуру дисциплины; освоил новации лекционного курса по сравнению с учебной литературой; изложение полученных знаний полное, системное; допускаются единичные ошибки, самостоятельно исправляемые при собеседовании

Таблица 7 – Критерии оценивания

Оценка	Критерии оценивания
Высокий уровень «5» (отлично)	оценку «отлично» заслуживает обучающийся, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы.
Средний уровень «4» (хорошо)	оценку «хорошо» заслуживает обучающийся, практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
Пороговый уровень «3» (удовлетворительно)	оценку «удовлетворительно» заслуживает обучающийся, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы.
Минимальный уровень «2» (неудовлетворительно)	оценку «неудовлетворительно» заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебная литература

6.1.1 Павловская Т.А. С#. Программирование на языке высокого уровня: *учебник для вузов / Т. А. Павловская. - СПб. : Питер, 2012. - 432с.: ил.

6.1.2 Кью, Дж. Объектно-ориентированное программирование: учебный курс / Кью Дж., М.Джеанини. - СПб.: Питер, 2005. - 238с.

6.1.3 Павловская, Т.А. С/С++. Программирование на языке высокого уровня: *учебник для вузов (для магистров и бакалавров) / Т. А. Павловская. - СПб.: Питер, 2012. - 461с.: ил. - (Учебник для вузов; Стандарт третьего поколения).

6.1.4 Павловская, Т.А. С++: объектно-ориентированное программирование: практикум: *учебное пособие для вузов / Т.А. Павловская, Ю.А. Щупак. - СПб. : Питер, 2008. - 265с.

Библиотечный фонд укомплектован печатными изданиями из расчета не менее 0,25 экземпляра каждого из изданий, указанных выше на каждого обучающегося из числа лиц, одновременно осваивающих соответствующую дисциплину (модуль).

6.2. Методические указания, рекомендации и другие материалы к занятиям

7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебный процесс по дисциплине обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного (необходимости).

7.1. Перечень информационных справочных систем

Дисциплина, относится к группе дисциплин, в рамках которых предполагается использование информационных технологий как вспомогательного инструмента.

Информационные технологии применяются в следующих направлениях: при подготовке и оформлении отчетов о лабораторных работах, выполнении заданий для самостоятельной работы.

Таблица 8 – Перечень электронных библиотечных систем

№	Наименование ЭБС	Ссылка к ЭБС
1	Консультант студента	http://www.studentlibrary.ru/
2	Лань	https://e.lanbook.com/

7.2. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины

Таблица 9 – Программное обеспечение

№ п/п	Программное обеспечение, используемое в университете на договорной основе	Программное обеспечение свободного распространения
1	Microsoft Windows 10 (подписка MSDN 700593597, подписка DreamSparkPremium, 19.06.19)	Adobe Acrobat Reader https://acrobat.adobe.com/ru/ru/acrobat/pdf-reader.html
2	Microsoft office 2010 (Лицензия № 49487295 от 19.12.2011)	OpenOffice https://www.openoffice.org/ru/
3	Консультант Плюс	PTC Mathcad Express https://www.mathcad.com/ru

Перечень современных профессиональных баз данных и информационных справочных систем

В таблице 10 указан перечень профессиональных баз данных и информационных справочных систем, к которым обеспечен доступ (удаленный доступ).

Таблица 10 – Перечень современных профессиональных баз данных и информационных справочных систем

№ п/п	Наименование профессиональной базы данных, информационно-справочной системы	Доступ к ресурсу (удаленный доступ с указанием ссылки/доступ из локальной сети университета)
1	2	3
1	База данных стандартов и регламентов РОССТАНДАРТ	https://www.gost.ru/portal/gost//home/standarts
2	Перечень профессиональных баз данных и информационных справочных систем	https://cyberpedia.su/21x47c0.html
3	Инструменты и веб-ресурсы для веб-разработки – 100+	https://techblog.sdstudio.top/blog/instrumenty-i-veb-resursy-dlia-veb-razrabotki-100-plus
4	Справочная правовая система «КонсультантПлюс»	доступ из локальной сети

8. ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ

В таблице 11 указан перечень образовательных ресурсов, имеющих формы, адаптированные к ограничениям их здоровья, а также сведения о наличии специальных технических средств обучения коллективного и индивидуального пользования.

Таблица 11 – Образовательные ресурсы для инвалидов и лиц с ОВЗ

№	Перечень образовательных ресурсов, приспособленных для использования инвалидами и лицами с ОВЗ	Сведения о наличии специальных технических средств обучения коллективного и индивидуального пользования
1	ЭБС «Консультант студента»	озвучка книг и увеличение шрифта
2	ЭБС «Лань»	специальное мобильное приложение - синтезатор речи, который воспроизводит тексты книг и меню навигации
3	ЭБС «Юрайт»	версия для слабовидящих

Согласно Федеральному Закону об образовании 273-ФЗ от 29.12.2012 г. ст. 79, п.8 "Профессиональное обучение и профессиональное образование обучающихся с ограниченными возможностями здоровья осуществляются на основе образовательных программ, адаптированных при необходимости для обучения указанных обучающихся". АОП разрабатывается по каждой направленности при наличии заявлений от обучающихся, являющихся инвалидами или лицами с ОВЗ и изъявивших желание об обучении по данному типу образовательных программ.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Учебные аудитории для проведения занятий по дисциплине, оснащены оборудованием и техническими средствами обучения.

В таблице 12 перечислены:

- учебные аудитории для проведения учебных занятий, оснащенные оборудованием и техническими средствами обучения;
- помещения для самостоятельной работы обучающихся, которые оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ДПИ НГТУ.

Таблица 12 – Оснащенность аудиторий и помещений для самостоятельной работы обучающихся по дисциплине

№	Наименование аудиторий и помещений для самостоятельной работы	Оснащенность аудиторий помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
1	1433А Аудитория для лекционных и практических занятий Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	Комплект демонстрационного оборудования: ПК, с выходом на телевизор 32" (80 см) LED Prestigio PTV32SS06Z_CIS_BK черный [HD, 1366x768, Wi-Fi, Android (AOSP), HDMI x 3, USB x 2 шт, VGA (D-Sub)]	<ul style="list-style-type: none"> • Microsoft Windows 10 Домашняя (поставка с ПК) • LibreOffice 6.1.2.1. (свободное ПО) • Foxit Reader (свободное ПО); • 7-zip для Windows (свободное ПО)
2	1234 Научно-техническая библиотека ДПИ НГТУ, студенческий читальный зал; Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	Комплект демонстрационного оборудования: ПК, с выходом на мультимедийный проектор, на базе Intel Pentium G4560 3.5 ГГц, 4 Гб ОЗУ, монитор 20' – 1шт. Мультимедийный проектор Epson- 1 шт; Экран – 1 шт.; Набор учебно-наглядных пособий	<ul style="list-style-type: none"> • Microsoft Windows 10 Домашняя (поставка с ПК) • LibreOffice 6.1.2.1. (свободное ПО) • Foxit Reader (свободное ПО); • 7-zip для Windows (свободное ПО)
3	1443а компьютерный класс - помещение для СРС, курсового проектирования (выполнения курсовых работ), Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	<ul style="list-style-type: none"> • ПК на базе Intel Celeron 2.67 ГГц, 2 Гб ОЗУ, монитор Acer 17' – 4 шт. ПК подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета 	<ul style="list-style-type: none"> • Microsoft Windows 7 (подписка DreamSpark Premium) • Apache OpenOffice 4.1.8 (свободное ПО); • Mozilla Firefox (свободное ПО); • Adobe Acrobat Reader (свободное ПО); • 7-zip для Windows (свободное ПО); • КонсультантПлюс (ГПД № 0332100025418000079 от 21.12.2018);

10. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

10.1. Общие методические рекомендации для обучающихся по освоению дисциплины, образовательные технологии

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

Контактная работа: аудиторная, внеаудиторная, а также может проводиться в электронной информационно-образовательной среде университета (далее - ЭИОС).

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- знакомство с материалами лекций и презентациями в среде MOODLE;
- балльно-рейтинговая технология оценивания;
- промежуточный контроль знаний в форме тестирования в среде MOODLE.

При преподавании дисциплины «Объектно-ориентированное программирование», используются современные образовательные технологии, позволяющие повысить активность обучающихся при освоении материала курса и предоставить им возможность эффективно

реализовать часы самостоятельной работы.

Весь лекционный материал курса сопровождается компьютерными презентациями, в которых наглядно преподносится материал различных разделов курса, что дает возможность обсудить материал с обучающимися во время чтения лекций, активировать их деятельность при освоении материала. Материалы лекций в виде слайдов находятся в свободном доступе в системе MOODLE и могут быть получены до чтения лекций и проработаны обучающимися в ходе самостоятельной работы.

На лекциях и практических занятиях реализуются интерактивные технологии, приветствуются вопросы и обсуждения, используется личностно-ориентированный подход, технология работы в малых группах, что позволяет обучающимся проявить себя, получить навыки самостоятельного изучения материала, выровнять уровень знаний в группе.

Все вопросы, возникшие при самостоятельной работе над домашним заданием, подробно разбираются на практических занятиях и лекциях. Проводятся индивидуальные и групповые консультации с использованием как встреч с обучающимися, так и современных информационных технологий (электронная почта).

Иницируется активность обучающихся, поощряется задание любых вопросов по материалу, практикуется индивидуальный ответ на вопросы обучающегося, рекомендуются методы успешного самостоятельного усвоения материала в зависимости от уровня его базовой подготовки.

Для оценки знаний, умений, навыков и уровня сформированности компетенции применяется балльно-рейтинговая система контроля и оценки успеваемости обучающихся в процессе текущего контроля.

Промежуточная аттестация проводится в форме зачета и экзамена с учетом текущей успеваемости.

Результат обучения считается сформированным на повышенном уровне, если теоретическое содержание курса освоено полностью. При устных собеседованиях обучающийся исчерпывающе, последовательно, четко и логически излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, использует в ответе дополнительный материал. Все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты, проявляет самостоятельность при выполнении заданий.

Результат обучения считается сформированным на пороговом уровне, если теоретическое содержание курса в основном освоено. При устных собеседованиях обучающийся последовательно излагает учебный материал; при затруднениях способен после наводящих вопросов продолжить обсуждение, справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий.

Результат обучения считается несформированным, если обучающийся при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже трех по оценочной системе, что соответствует допороговому уровню.

10.2. Методические указания для занятий лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины (таблица 4). Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала. Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

10.3. Методические указания по освоению дисциплины на практических занятиях

Подготовку к каждому практическому занятию обучающийся должен начать с ознакомления с рекомендуемой литературой (таблица 4), которая отражает содержание предложенной темы. Каждая самостоятельно выполненная контрольная работа по индивидуальному варианту подлежит проверке преподавателем.

При оценивании контрольных работ учитывается следующее:

- качество решения задач и степень соответствия результатов работы заданным требованиям;
- целесообразность использования изученных методов;
- качество комментариев к решению.

10.4. Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающихся к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающихся на занятиях и в качестве выполненных заданий для самостоятельной работы и других форм текущего контроля.

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы, представленной в Разделе 6.

В процессе самостоятельной работы при изучении дисциплины обучающиеся могут работать на компьютере в специализированных аудиториях для самостоятельной работы (таблица 12). В аудиториях имеется доступ через информационно-телекоммуникационную сеть «Интернет» к электронной информационно-образовательной среде университета (ЭИОС) и электронной библиотечной системе (ЭБС), где в электронном виде располагаются учебные и учебно-методические материалы, которые могут быть использованы для самостоятельной работы при изучении дисциплины.

Методические указания для выполнения курсовой работы

Выполнение курсовой работы способствует лучшему освоению обучающимися учебного материала, формирует практический опыт и умения по изучаемой дисциплине, способствует формированию у обучающихся готовности к самостоятельной профессиональной деятельности, является этапом к выполнению выпускной квалификационной работы.

Примерная тематика курсовых работ

1. Программная реализация алгоритмов динамического программирования;
2. Оценка эффективности алгоритмов сортировки;
3. Хэширование методом цепочек;
4. Топологическая сортировка на графе, представленном списком
5. Программная реализация структуры данных «AVL-дерево» с графической визуализацией AVL-деревьев
6. Программная реализация алгоритма рекурсивного обхода вершин графа;
7. Программная реализация алгоритма Прима и алгоритма Краскала;
8. Программная реализация алгоритма сетевого планирования и управления

Порядок консультирования при выполнении курсовой работы

Тема курсовой работы выбирается студентом из списка тем и утверждается преподавателем за месяц до защиты работы. Пояснения к выполнению работы проводятся во время лабораторных занятий индивидуально для каждого студента. Консультация перед защитой курсовой работы проводится в дополнительное время, отведенное преподавателем в конце семестра.

Правила оформления пояснительной записки к курсовой работе

Пояснительная записка должна содержать следующие разделы:

- 1) Аннотацию (из каких разделов (тем) состоит работа). Чему посвящен каждый из разделов, каков их объем, сколько рисунков, сколько приложений, на каких данных проверена программа;
- 2) Введение (описание предметных областей, где может быть использован алгоритм);

- 3) Постановка задачи (КОРРЕКТНАЯ формулировка задания, описание форматов входных и выходных данных с их интерпретацией в терминах алгоритмов);
- 4) Сведения из теории (описание алгоритма, доказательство его корректности, оценка эффективности и т.д.);
- 5) Описание используемых структур данных и значимых для алгоритма переменных (классов);
- 6) Описание реализованных методов и функций (что они делают, с какими данными работают, как взаимодействуют друг с другом);
- 7) ФРАГМЕНТЫ кода (значимые для алгоритмов) с подробными комментариями;
- 8) Результаты работы (входные и выходные данные с их интерпретацией);
- 9*) интерфейс с "руководством пользователя", представленный в виде сканов с экрана и их описанием;
- 10) список использованной литературы;

В Приложение нужно поместить полный текст программы с комментариями.

В тексте пояснительной записки должны присутствовать ссылки на книги из списка литературы и на Приложения.

Бланки титульного листа и листа Задания есть в приложениях к методичке по курсовой работе. Их нужно грамотно заполнить:

Исходные данные к работе - задание и название реализованных алгоритмов;

Содержание пояснительной записки - какие главы, объем, кол-во рисунков, кол-во источников в списке литературы;

Перечень графического материала - названия всех слайдов презентации, которую необходимо подготовить к защите.

Необходимо указать даты: выдача задания и срока сдачи (дата защиты работы)

Всю работу необходимо оформлять в "рамочках" в которых должен присутствовать шифр, номер страницы.

КР-АСД-НГТУ-021ПМ-000-22 Вместо трех нулей - Ваш номер в журнале.

На листе Содержания рамочка должна быть "большая", где, кроме кода, должны быть подписи и название курсовой работы.

Порядок защиты курсовой работы:

Для защиты работы каждый студент должен подготовить презентацию, которую необходимо демонстрировать при докладе. В презентации поощряется использование всевозможного иллюстративного материала (схем, рисунков, диаграмм, графиков и т.д.).

Количество слайдов, содержащих текстовую информацию (например, определений понятий) должно быть минимально!

В состав презентации к курсовой работе должны быть включены, как минимум, следующие слайды:

- 1) Название работы, кто выполнил (ФИО, группа, курс, направление подготовки, ФИО руководителя);
- 2) Постановка задачи - формально точное определение того, что необходимо сделать;
- 3) Теоретические основы разрабатываемой темы;
- 4) Возможные практические приложения разработанной программы;
- 5) Язык, на котором было программно реализовано приложение и те стандартные библиотеки, средства, шаблоны и методы языка, которые были использованы;
- 6) Описание классов и структур данных (могут содержаться фрагменты кода, наиболее ярко представляющие структуры данных, сземы, представляющие структуры данных);
- 7) Описание алгоритмов (фрагменты кода и комментарии к ним или блок-схемы);
- 8) Схема взаимодействия классов или/и методов внутри одного модуля с пояснением;
- 9) Примеры работы с указанием исходных данных и полученных результатов на разных наборах данных;
- 10) Схема руководства пользователя (скриншот интерфейса).

Время доклада составляет 5 – 6 минут, в которые докладчик должен изложить результаты своей работы и заинтересовать аудиторию. После доклада члены комиссии задают вопросы докладчику. Время, отведенное на прения составляет 5 – 10 минут.

11 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

11.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта в ходе текущего контроля успеваемости

Для текущего контроля знаний обучающихся по дисциплине проводится комплексная оценка знаний, включающая

- проведение лабораторных работ;
- выполнение заданий для самостоятельной работы;
- экзамен, зачет.

11.1.1. Типовые задания для лабораторных работ

1. Дан одномерный массив вещественного типа. Найти минимальное значение из сумм «противоположных» элементов массива $\min(a[1]+a[n], a[2]+a[n-1], a[3]+a[n-2], \dots)$,

а также индексы элементов, для которых соответствующая сумма является минимальной.

Исходный массив и полученные результаты вывести на печать.

2. Дан одномерный массив вещественного типа. Вычислить сумму и произведение элементов массива, значения каждого из которых больше заданного значения.

Исходный массив и полученные результаты вывести на печать.

Предусмотреть вывод сообщения, когда подобных элементов в массиве нет.

3. Дан одномерный массив вещественного типа. Найти среднее геометрическое положительных элементов массива.

Исходный массив и полученный результат вывести на печать.

Предусмотреть вывод сообщения, когда положительных элементов в массиве нет.

4. Дан одномерный массив вещественного типа. Вычислить сумму и количество элементов массива, стоящих слева от первого нулевого элемента.

Исходный массив и полученный результат вывести на печать.

Предусмотреть вывод сообщения, когда первый элемент равен нулю или нулевого элемента в массиве нет.

5. Дан одномерный массив вещественного типа. Изменить порядок следования элементов в массиве на обратный. Дополнительного массива не заводить.

Исходный и полученный массивы вывести на печать.

6. Дан одномерный массив вещественного типа. Найти в массиве третий, считая слева направо, элемент, имеющий заданное значение, т.е. определить индекс этого элемента.

Исходный массив и полученный результат вывести на печать.

Предусмотреть вывод сообщения, когда подобного элемента в массиве нет.

7. Дан одномерный массив вещественного типа. Осуществить циклический сдвиг элементов массива влево на три позиции. Дополнительного массива не заводить.

Исходный и полученный массивы вывести на печать.

8. Дан одномерный массив вещественного типа. Найти наименьший из положительных элементов массива (определить значение и индекс этого элемента).

Исходный массив и полученный результат вывести на печать.

Предусмотреть вывод сообщения, когда положительных элементов в массиве нет.

9. Дан одномерный массив вещественного типа. За один просмотр найти первый и последний, считая слева направо, элемент, имеющий заданное значение, т.е. определить индексы этих элементов.

Исходный массив и полученный результат вывести на печать.

Предусмотреть вывод сообщения, когда подобных элементов в массиве нет.

11.1.2. Типовые задания для самостоятельной работы обучающихся очной формы

Составить программу, которая для заданной строки s :

1. определяет, какой из двух заданных символов встречается чаще в строке;

2. подсчитывает общее число вхождений символов x и y ;
3. подсчитывает количество букв в строке;
4. определяет, имеются ли в строке два соседствующих одинаковых символа;
5. подсчитывает сумму всех содержащихся в ней цифр;
6. подсчитывает количество содержащихся в ней цифр;
7. находит порядковые номера первого и последнего вхождения символа x ;
8. выводит на экран последовательность символов, расположенных до первого двоеточия;
9. выводит на экран последовательность символов, расположенных после последнего двоеточия;
10. определяет, сколько различных символов встречается в строке.

11.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта в ходе промежуточной аттестации по дисциплине

Форма проведения промежуточной аттестации по дисциплине: зачета, экзамен (по результатам накопительного рейтинга или в форме компьютерного тестирования).

11.2.1. Типовые тестовые задания на экзамене и зачете:

1. Создать класс Point, разработав следующие элементы класса:
 - a. Поля:
 - `int x, y;`
 - b. Конструкторы, позволяющие создать экземпляр класса:
 - с нулевыми координатами;
 - с заданными координатами.
 - c. Методы, позволяющие:
 - вывести координаты точки на экран;
 - рассчитать расстояние от начала координат до точки;
 - переместить точку на плоскости на вектор (a, b) .
 - d. Свойства:
 - получить-установить координаты точки (доступное для чтений и записи);
 - позволяющие умножить координаты точки на скаляр (доступное только для записи).
2. Создать класс Triangle, разработав следующие элементы класса:
 - a. Поля:
 - `int a, b, c;`
 - b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
 - c. Методы, позволяющие:
 - вывести длины сторон треугольника на экран;
 - рассчитать периметр треугольника;
 - рассчитать площадь треугольника.
 - d. Свойства:
 - позволяющее получить-установить длины сторон треугольника (доступное для чтения и записи);
 - позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).
3. Создать класс Rectangle, разработав следующие элементы класса:
 - a. Поля:
 - `int a, b;`
 - b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
 - c. Методы, позволяющие:
 - вывести длины сторон прямоугольника на экран;
 - рассчитать периметр прямоугольника;
 - рассчитать площадь прямоугольника.

- d. Свойства:
- получить-установить длины сторон прямоугольника (доступное для чтения и записи);
 - позволяющее установить, является ли данный прямоугольник квадратом (доступное только для чтения).
4. Создать класс Money, разработав следующие элементы класса:
- a. Поля:
- `int first;`//номинал купюры
 - `int second;`//количество купюр
- b. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.
- c. Методы, позволяющие:
- вывести номинал и количество купюр;
 - определить, хватит ли денежных средств на покупку товара на сумму N рублей.
 - определить, сколько шт товара стоимости n рублей можно купить на имеющиеся денежные средства.
- d. Свойства:
- позволяющее получить-установить значение полей (доступное для чтения и записи);
 - позволяющее расчитать сумму денег (доступное только для чтения).
5. Создать класс для работы с одномерным массивом целых чисел. Разработать следующие элементы класса:
- a. Поля:
- `int [] IntArray;`
 - `int n.`
- b. Конструктор, позволяющий создать массив размерности n.
- c. Методы, позволяющие:
- ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - отсортировать элементы массива в порядке возрастания.
- d. Свойства:
- возвращающее размерность массива (доступное только для чтения);
 - позволяющее домножить все элементы массива на скаляр (доступное только для записи).
6. Создать класс для работы с двумерным массивом целых чисел. Разработать следующие элементы класса:
- a. Поля:
- `int [,] IntArray;`
 - `int n.`
- b. Конструктор, позволяющий создать массив размерности n×n.
- c. Методы, позволяющие:
- ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - вычислить сумму элементов i-того столбца.
- d. Свойства:
- позволяющее вычислить количество нулевых элементов в массиве (доступное только для чтения);
 - позволяющее установить значение всех элементы главной диагонали массива равное скаляру (доступное только для записи).
7. Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:
- a. Поля:

- `double [][] DoubleArray;`
 - `int n, m.`
- b. Конструктор, позволяющий создать массив размерности $n \times m$.
- c. Методы, позволяющие:
- ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - отсортировать элементы каждой строки массива в порядке убывания.
- d. Свойства:
- возвращающее общее количество элементов в массиве (доступное только для чтения);
 - позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи).
8. Создать класс для работы со строками. Разработать следующие элементы класса:
- a. Поля:
- `StringBuilder line;`
 - `int n.`
- b. Конструктор, позволяющий создать строку из n символов.
- c. Методы, позволяющие:
- подсчитать количество пробелов в строке;
 - заменить в строке все прописные символы на строчные;
 - удалить из строки все знаки препинания.
- d. Свойства:
- возвращающее общее количество элементов в строке (доступное только для чтения);
 - позволяющее установить значение поля, в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи)
9. Создать класс для работы с регулярными выражениями. Разработать следующие элементы класса:
- a. Поля:
- `Regex r;`
 - `string text;`
- b. Методы, позволяющие:
- определить, содержит ли текст фрагменты, соответствующие шаблону поля;
 - вывести на экран все фрагменты текста, соответствующие шаблону поля;
 - удалить из текста все фрагменты, соответствующие шаблону поля;
- Свойства:
- позволяющее установить или получить строковое поле класса (доступно для чтения и записи)
 - позволяющее установить или получить регулярное выражение, хранящееся в соответствующем поле класса (доступно для чтения и записи)
10. Создать класс для работы с датой. Разработать следующие элементы класса:
- a. Поле `DateTime data.`
- b. Конструкторы, позволяющие установить:
- заданную дату
 - дату 1.01.2009
- c. Методы, позволяющие:
- вычислить дату предыдущего дня;
 - вычислить дату следующего дня;
 - определить сколько дней осталось до конца месяца.
- d. Свойства:

- позволяющее установить или получить значение поле класса (доступно для чтения и записи)
- позволяющее определить год высокосным (доступно только для чтения)

Регламент проведения промежуточного контроля в форме компьютерного тестирования

Кол-во заданий в банке вопросов	Кол-во заданий, предъявляемых обучающемуся	Время на тестирование, мин.
200	10	90

Полный фонд оценочных средств для проведения промежуточной аттестации в форме компьютерного тестирования размещен в банке вопросов данного курса дисциплины в СДО MOODLE.