

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
 ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
 ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
 ВЫСШЕГО ОБРАЗОВАНИЯ
 «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
 (ФГБОУ ВО «ВГТУ», ВГТУ)

«УТВЕРЖДАЮ»

Председатель Ученого совета
 Факультета информационных
 технологий и компьютерной
 безопасности

Пасмурнов С.М.

(подпись)

23 09 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Лингвистическое и программное обеспечение САПР

(наименование дисциплины (модуля) по УП)

Закреплена за кафедрой: компьютерных интеллектуальных технологий проектирования

Направление подготовки (специальности):

09.03.01 Информатика и вычислительная техника

(код, наименование)

Профиль: системы автоматизированного проектирования в машиностроении

(название профиля по УП)

Часов по УП: 288; Часов по РПД: 288;

Часов по УП (без учета часов на экзамены): 252; Часов по РПД: 252;

Часов на интерактивные формы (ИФ) обучения по УП: 0

Часов на интерактивные формы (ИФ) обучения по РПД: 0

Часов на самостоятельную работу по УП: 126 (43,7 %);

Часов на самостоятельную работу по РПД: 126 (43,7 %)

Общая трудоемкость в ЗЕТ: 8;

Виды контроля в семестрах (на курсах): Экзамены -6; Зачеты – 5; Зачеты с оценкой – 0;

Курсовые проекты - 6; Курсовые работы - 0.

Форма обучения: очная;

Срок обучения: нормативный.

Распределение часов дисциплины по семестрам

Вид занятий	№ семестров, число учебных недель в семестрах																			
	1 / 18		2 / 18		3 / 18		4 / 18		5 / 18		6 / 18		7 / 18		8 / 12		Итого			
	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД		
Лекции									36	36	18	18							54	54
Лабораторные									36	36	36	36							72	72
Практические																			0	0
Ауд. занятия									72	72	54	54							126	126
Сам. работа									36	36	90	90							126	126
Итого									108	108	144	144							252	252

Сведения о ФГОС, в соответствии с которым разработана рабочая программа дисциплины (модуля) – 09.03.01 «Информатика и вычислительная техника», утвержден приказом Министерства образования и науки Российской Федерации от 12 марта 2015 № 219

Программу составил:  к.т.н., П.Ю. Гусев
(подпись, ученая степень, ФИО)

Рецензент (ы):  к.ф.-м.н., доцент Н.А. Тюкачев
(подпись, ученая степень, ФИО)

Рабочая программа дисциплины составлена на основании учебного плана направления подготовки (специальности) 09.03.01 Информатика и вычислительная техника, профиль Системы автоматизированного проектирования в машиностроении.

Рабочая программа обсуждена на заседании кафедры компьютерных интеллектуальных технологий проектирования

протокол № 1 от 30.08 2017 г.

Зав. кафедрой КИТП  д.т.н., проф. М.И. Чижов

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	Цель дисциплины – изучение основных положений по управлению задачами и памятью в операционных системах, разработке прикладных программных средств, разработке языков для реализации интерфейса и написания программ, решению вопросов трансляции программ, написанных на различных языках.
1.2	Для достижения цели ставятся задачи:
1.2.1	изучение основных стандартов, поддерживающих создание мобильных программ в информационных системах
1.2.2	ознакомление с базовыми алгоритмами сортировки и поиска, основными комбинаторными понятиями и методами вычислений
1.2.3	изучение динамического представления данных в памяти и работа с подобными данными
1.2.4	изучение теории формальных грамматик и языков
1.2.5	формирование единой системы понятий, связанных с различными видами обеспечения САПР
1.2.6	обучение основным приемам эффективного использования памяти компьютера

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Цикл (раздел) ООП: Б1	код дисциплины в УП Б1.В.ОД.5
2.1 Требования к предварительной подготовке обучающегося	
Изучение данной дисциплины базируется на следующих курсах: «Математика» «Информатика», «Объектно-ориентированное программирование».	
2.2 Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее	
Б1.В.ОД.7	Разработка САПР
Б1.В.ОД.8	Автоматизация конструкторского и технологического проектирования
Б1.В.ДВ.71	Основы проектирования производственных систем

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

(ПВК-1)	Способностью разрабатывать компоненты программных комплексов и информационных систем, используя технологии программирования и инструментальные средства разработки
(ПВК-7)	способностью создавать модели объектов и процессов систем автоматизированного проектирования

В результате освоения дисциплины обучающийся должен

3.1	Знать:
	основные понятия формальных языков и формальных грамматик (ПВК-7)
	назначение и принципы работы трансляторов, интерпретаторов и компиляторов (ПВК-1)
	последовательность разработки прикладных программных средств в САПР (ПВК-1)
	основные проблемы подготовки программного и лингвистического обеспечения в САПР (ПВК-7)
	стандарты, поддерживающие создание программ в информационных системах (ПВК-1)
	способы представления данных в памяти компьютера (ПВК-1)
3.2	Уметь:
	формировать техническое задание на разработку прикладного программного обеспечения (ПВК-1)
	использовать современные инструментальные средства разработки прикладных программ (ПВК-1)
	разрабатывать программное и лингвистическое обеспечения САПР (ПВК-7)
3.3	Владеть:
	навыками разработки прикладных программных продуктов в САПР (ПВК-1)
	методологией составления лингвистического обеспечения в САПР (ПВК-7)
	методологией составления программного обеспечения в САПР (ПВК-7)

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

№ п/п	Наименование раздела дисциплины	Семестр	Неделя семестра	Вид учебной нагрузки и их трудоемкость в часах				
				Лекции	Практические занятия	Лабораторные работы	СРС	Всего часов
1	Программное обеспечение САПР. Введение	5	1-3	6		6	6	18
2	Стандарты, поддерживающие создание прикладных программ в информационных системах	5	4-6	6		6	6	18
3	Динамические структуры данных	5	7-9	6		6	6	18
4	Алгоритмы сортировки	5	10-12	6		6	6	18
5	Алгоритмы поиска	5	13-15	6		6	6	18
6	Комбинаторика	5	16-18	6		6	6	18
7	Лингвистическое обеспечение САПР. Введение	6	1-3	3		6	15	24
8	Формальные языки и грамматики	6	4-6	3		6	15	24
9	Регулярные выражения	6	7-9	3		6	15	24
10	Лексический анализатор	6	10-12	3		6	15	24
11	Синтаксический анализатор	6	13-15	3		6	15	24
12	Трансляторы, интерпретаторы и компиляторы	6	16-18	3		6	15	24
Итого				54		72	126	252

4.1 Лекции

4.1.1 Лекции 5 семестр

Неделя семестра	Тема и содержание лекции	Объем часов	В том числе, в интерактивной форме (ИФ)
5 семестр		36	
Программное обеспечение САПР. Введение		6	
1	Вводная лекция	2	
2	САПР Назначение и структура программного обеспечения	2	
3		2	
Методическое обеспечение программного обеспечения			
Стандарты, поддерживающие создание прикладных программ в информационных системах		6	
4	Введение в использование стандартов разработки интерфейсов пользователей с операционной средой	2	
5	Модель зрелости процесса разработки ПО	2	
6	Стандарт оценки программных процессов	2	
Динамические структуры данных		6	
7	Динамическое представление данных в памяти компьютера	2	
8		2	
9	Списки, стеки, очереди	2	
9	Деревья. Бинарные деревья. Алгоритмы прохождения деревьев	2	
Алгоритмы сортировки		6	
10	Постановка задачи методов сортировки	2	
11	Сортировка вставками, сортировка выбором, сортировка слияние	2	
12		2	
Хеширование. Хеш-функции			
Алгоритмы поиска		6	
13	Постановка задачи алгоритмов поиска	2	
14	Алгоритм быстрого поиска. Последовательный поиск	2	
15	Интерполяционный поиск. Поиск с использованием индексации по ключам	2	
Комбинаторика		6	
16	Комбинаторика и переборные алгоритмы	2	
17	Лексикографический порядок перестановок	2	
18	Сочетание. Описание алгоритма	2	
Итого часов		36	

4.1.1 Лекции 6 семестр

Неделя семестра	Тема и содержание лекции	Объем часов	В том числе, в интерактивной форме (ИФ)
6 семестр		18	
Лингвистическое обеспечение САПР. Введение		3	
1-2	Вводная лекция	1	
1-2	САПР Назначение и структура лингвистического обеспечения	1	
3-4		1	
Методическая поддержка лингвистического обеспече-			

	ния		
Формальные языки и грамматики		3	
3-4	Определение формальной грамматики и языка	1	
5-6	Типы формальных языков и грамматик. Классификация по Хомскому	1	
5-6	Способы задания схем грамматик. Форма Наура-Бэкуса	1	
Регулярные выражения		3	
7-8	Элементы регулярных выражений	1	
7-8	Коды символов и литеральные константы	1	
9-10	Группировка и ограничение регулярных фрагментов	1	
Лексический анализатор		3	
9-10	Лексический анализ	1	
11-12	Структура файла спецификации лексем	1	
11-12	Спецификация правил	1	
Синтаксический анализатор		3	
13-14	Синтаксический анализ в лингвистическом обеспечении САПР	1	
13-14	Подготовка спецификаций	1	
15-16	Описание входного синтаксиса. Пример использования синтаксического анализатора	1	
Трансляторы, интерпретаторы и компиляторы		3	
15-16	Трансляция	1	
17-18	Интерпретаторы	1	
17-18	Компиляторы	1	
Итого часов		18	

4.3 Лабораторные работы

Неделя семестра	Наименование лабораторной работы	Объем часов	В том числе в интерактивной форме (ИФ)	Виды контроля
1	2	3	4	5
5 семестр		36	0	
1-2	Работа с WinForms с#. Линейные алгоритмы	4	0	Отчет и защита
3-4	Работа с файлами. Работа со строками	4	0	Отчет и защита
5-6	Практическое применение классов в программировании	4	0	Отчет и защита
7-8	Перегрузка методов и операторов	4	0	Отчет и защита
9-10	Работа с динамическими структурами данных: списки, стеки, очереди	4	0	Отчет и защита
11-12	Деревья. Бинарные деревья	4	0	Отчет и защита
13-14	Алгоритмы сортировки	4	0	Отчет и защита
15-16	Поисковые алгоритмы	4	0	Отчет и защита
6 семестр		36	0	
1-2	Решение задач по составлению формальных грамматик	4	0	Отчет и защита

3-4	Описание условных структур языка программирования с помощью формальной грамматики	4	0	Отчет и защита
5-6	Классификация грамматик	4	0	Отчет и защита
7-8	Задание грамматик в различных формах записи	4	0	Отчет и защита
9-10	Знакомство с регулярными выражениями	4	0	Отчет и защита
11-12	Применение регулярных выражений в с#	4	0	Отчет и защита
13-14	Лексический анализатор	4	0	Отчет и защита
15-16	Синтаксический анализатор	4	0	Отчет и защита
17-18	Разработка транслятора	4	0	Отчет и защита
17-18	Разработка компилятора	4	0	Отчет и защита
Итого часов		36	24	

4.4 Самостоятельная работа студента (СРС)

Неделя семестра	Содержание СРС	Виды контроля	Объем часов
5 семестр			36
1	Подготовка к выполнению лаб. работы	допуск к выполнению	2
2	Работа с конспектом лекций, с учебником	тестовые вопросы	2
3	Подготовка к выполнению лаб. работы	допуск к выполнению	2
4	Подготовка к защите лаб. работ	защита лаб. работы	2
5	Работа с конспектом лекций, с учебником	тестовые вопросы	2
6	Подготовка конспекта по теме для самостоятельного изучения	проверка конспекта	2
7	Подготовка конспекта по теме для самостоятельного изучения	проверка конспекта	2
8	Подготовка к защите лаб. работ	защита лаб. работы	2
9	Подготовка к выполнению лаб. работы	допуск к выполнению	2
10	Работа с конспектом лекций, с учебником	тестовые вопросы	2
11	Подготовка домашнего задания	отчет	2
12	Подготовка конспекта по теме для самостоятельного изучения	проверка конспекта	2
13	Подготовка к защите лаб. работ	отчет, защита	2
14	Подготовка домашнего задания	отчет	2
15	Подготовка к защите лаб. работ	отчет, защита	2
16	Работа с конспектом лекций, с учебником	тестовые вопросы	2
17	Подготовка конспекта по теме для само-	зачет	2

	стоятельного изучения		
6 семестр			90
1	Подготовка к выполнению лаб. работы	допуск к выполнению	5
2	Работа с конспектом лекций, с учебником	тестовые вопросы	5
3	Выполнение курсового проекта	пояснительная записка	5
4	Подготовка к защите лаб. работ	защита лаб. работы	5
5	Работа с конспектом лекций, с учебником	тестовые вопросы	5
6	Подготовка конспекта по теме для самостоятельного изучения	проверка конспекта	5
7	Выполнение курсового проекта	пояснительная записка	5
8	Подготовка к защите лаб. работ	защита лаб. работы	5
9	Подготовка к выполнению лаб. работы	допуск к выполнению	5
10	Работа с конспектом лекций, с учебником	тестовые вопросы	5
11	Выполнение курсового проекта	пояснительная записка	5
12	Подготовка конспекта по теме для самостоятельного изучения	проверка конспекта	5
13	Подготовка к защите лаб. работ	отчет, защита	5
14	Подготовка домашнего задания	отчет	5
15	Подготовка к защите лаб. работ	отчет, защита	5
16	Выполнение курсового проекта	пояснительная записка	5
17	Выполнение курсового проекта	пояснительная записка	5
18	Подготовка к экзамену	зачет	5

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

	В рамках изучения дисциплины предусмотрены следующие образовательные технологии:
5.1	Информационные лекции;
5.2	лабораторные работы: <ul style="list-style-type: none"> – выполнение лабораторных работ в соответствии с индивидуальным графиком, – защита выполненных работ;
5.3	самостоятельная работа студентов: <ul style="list-style-type: none"> – изучение теоретического материала, – подготовка к лекциям, лабораторным работам и практическим занятиям, – работа с учебно-методической литературой, – выполнение курсового проекта; – подготовка к текущему контролю успеваемости, зачету и экзамену.
5.4	консультации по всем вопросам учебной программы.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6.1	Контрольные вопросы и задания
6.1.1	Используемые формы текущего контроля: <ul style="list-style-type: none">– защита лабораторных работ;– теоретические вопросы;– выполнение курсового проекта.
6.1.2	Рабочая программа дисциплины обеспечена фондом оценочных средств для проведения входного, текущего контроля и промежуточной аттестации. Фонд включает примерные варианты домашних заданий и вопросы к зачету.

**7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ
ДИСЦИПЛИНЫ (МОДУЛЯ)**

№ п/п	Авторы, составители	Заглавие	Годы издания. Вид издания.	Обеспеченность
1. Основная литература				
Л1.1	Толоконникова, С.В.	Основы проектирования производственных систем. Учебное пособие.	2007 электрон.	1
2. Дополнительная литература				
Л2.1	Норенков И.П.	Основы автоматизированного проектирования. Учебник.	2006 печат.	0,4
3 Методические разработки				
Л3.1	Е. Д. Федоров, С. В. Голоконникова	№ 404-2007 Методические указания для организации самостоятельной работы по дисциплине "Основы проектирования производственных систем" для студентов специальности 230104 "Системы автоматизированного проектирования"	2009 печат.	1
Л3.2	Е. Д. Федоров, Е. И. Асташева	№ 328-2009 Методические указания к практическим занятиям по дисциплине "Основы проектирования производственных систем" для студентов специальности 230104 "Системы автоматизированного проектирования"	2009 печат.	1
Л3.3	М. И. Чижов, Ю. С. Скрипченко, П. Ю. Гусев.	№ 293-2011 Методические указания к выполнению лабораторных работ по дисциплине "Основы проектирования производственных систем" для студентов специальности 230104 "Системы автоматизированного проектирования"	2011 электрон.	1

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Специализированная лекционная аудитория , оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой
8.2	Учебные лаборатории: <ul style="list-style-type: none"> - "Лаборатория интеллектуальных систем проектирования" - "Лаборатория компьютерного моделирования и дизайна" - "Интернет-лаборатория"
8.4	Требуемое программное обеспечение: Microsoft Visual Studio 2015 (минимально 2010)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ»)

Фонд оценочных средств по дисциплине

«Лингвистическое и программное обеспечение САПР»

Направление подготовки **09.03.01 «Информатика и вычислительная техника»**

профиль **«Системы автоматизированного проектирования в машиностроении»**

Форма обучения очная

Срок обучения нормативный

1. Структурная матрица компетенций по дисциплине «Лингвистическое и программное обеспечение САПР»

Индекс компетенции / результата обучения	Наименование компетенции / результата обучения
ПВК-1	Способностью разрабатывать компоненты программных комплексов и информационных систем, используя технологии программирования и инструментальные средства разработки
P1.ПВК-1	Знает: - назначение и принципы работы трансляторов, интерпретаторов и компиляторов;
P2.ПВК-1	Знает: - последовательность разработки прикладных программных средств в САПР;
P3.ПВК-1	Знает: - стандарты, поддерживающие создание программ в информационных системах;
P4.ПВК-1	Знает: - способы представления данных в памяти компьютера;
P5.ПВК-1	Умеет: - формировать техническое задание на разработку прикладного программного обеспечения;
P6.ПВК-1	Умеет: - использовать современные инструментальные средства разработки прикладных программ;
P7.ПВК-1	Владеет: - навыками разработки прикладных программных продуктов в САПР;
ПВК-7	Способностью разрабатывать математическое, программное, лингвистическое, информационное, техническое, эргономическое, организационное и правовое обеспечение САПР
P1.ПВК-7	Знает: - основные понятия формальных языков и формальных грамматик;
P2.ПВК-7	Знает: - основные проблемы подготовки программного и лингвистического обеспечения в САПР
P3.ПВК-7	Умеет: - разрабатывать программное и лингвистическое обеспечения САПР
P4.ПВК-7	Владеет: - методологией составления лингвистического обеспечения в САПР;
P5.ПВК-7	Владеет: - методологией составления программного обеспечения в САПР;

2. Индексированные результаты обучения

Индекс	Результат	Индекс	Показатель
P1.ПВК-1	Знает назначение и принципы работы трансляторов, интерпретаторов и компиляторов;	П1.P1.ПВК-1	Знание назначения и принципов работы трансляторов
		П2.P1.ПВК-1	Знание назначения и принципов работы интерпретаторов и компиляторов
P2.ПВК-1	Знает последовательность разработки прикладных программных средств в САПР;	П1.P2.ПВК-1	Знание последовательности разработки ПО
P3.ПВК-1	Знает стандарты, поддерживающие создание программ в информационных системах;	П1.P3.ПВК-1	Знание модели зрелости процесса разработки ПО
		П2.P3.ПВК-1	Знает стандарт оценки программных процессов
P4.ПВК-1	Знает способы представления данных в памяти компьютера;	П1.P4.ПВК-1	Знание различных способов представления данных в памяти
P5.ПВК-1	Умеет формировать техническое задание на разработку прикладного программного обеспечения;	П1.P5.ПВК-1	Умение формировать техническое задание на разработку ПО
		П2.P51.ПВК-1	Умение формировать график выполнения этапов по техническому заданию
P6.ПВК-1	Умеет использовать современные инструментальные средства разработки прикладных программ;	П1.P6.ПВК-1	Умение использовать современные среды программирования
P7.ПВК-1	Владеет навыками разработки прикладных программных	П1.P7.ПВК-1	Владение практическими навыками программирования

	продуктов в САПР;		
P1.ПВК-7	Знает основные понятия формальных языков и формальных грамматик;	P1.P1.ПВК-7	Знание правил построения формальной грамматики
		P2.P1.ПВК-7	Знание классификации формальных грамматик
P2.ПВК-7	Знает основные проблемы подготовки программного и лингвистического обеспечения в САПР	P1.P1.ПВК-7	Знание основных проблем подготовки программного и лингвистического обеспечения в САПР и пути их решения
P3.ПВК-7	Умеет разрабатывать программное и лингвистическое обеспечения САПР	P1.P1.ПВК-7	Умение разрабатывать программное и лингвистическое обеспечения САПР
P4.ПВК-7	Владеет методологией составления лингвистического обеспечения в САПР	P1.P1.ПВК-7	Владение методологией составления лингвистического обеспечения в САПР
P5.ПВК-7	Владеет методологией составления программного обеспечения в САПР	P1.P1.ПВК-7	Владение методологией составления программного обеспечения в САПР

3. Оценочные средства

За. Курсовой проект.

Проверяемые результаты и показатели: П1.Р1.ПВК-1, П2.Р1.ПВК-1, П1.Р2.ПВК-1, П1.Р3.ПВК-1, П2.Р3.ПВК-1, П1.Р4.ПВК-1, П1.Р5.ПВК-1, П2.Р51.ПВК-1, П1.Р6.ПВК-1, П1.Р7.ПВК-1, П1.Р1.ПВК-7, П2.Р1.ПВК-7, П1.Р1.ПВК-7, П1.Р1.ПВК-7, П1.Р1.ПВК-7, П1.Р1.ПВК-7

Задание выполняется в среде разработки программного обеспечения Visual Studio (версия не ниже 2010). Тип проекта – «Приложение Windows Forms». Программа должна иметь дружелюбный пользовательский интерфейс.

Разрабатываемый проект представляет собой систему управления производственным подразделением. Система хранит данные о следующих объектах:

- производственное оборудование;
- детали, обрабатываемые на производстве;
- рабочие и вся информация о них;

- информация о производственных помещениях;
- транспортная система (транспортные тележки, кран-балки и т.д.).

Каждый объект должен быть описан как отдельный класс. Возможно также описание подтипов объектов с использованием классов-наследников (например: тип оборудования, специализация рабочего и т.д.).

Система должна поддерживать удобное добавление новых объектов. Также необходимо реализовать привязку объектов друг к другу, то есть: на конкретном оборудовании работает конкретный рабочий, и обрабатываются следующие типы деталей и т.д. По завершении работы приложения вся информация должна сохраняться в базу данных.

После ввода объектов необходимо реализовать графическое представление производственного участка: на форме рисуются контуры оборудования, указываются рабочие и детали, указывается транспортная система.

Система должна поддерживать управленческие функции. Необходимо реализовать возможность изменения порядка расстановки оборудования и перераспределения рабочих по оборудованию.

Индивидуальное задание:

1. Обеспечить возможность сравнения оборудования по заданным параметрам. Также осуществить возможность подбора оборудования по заданным параметрам.

2. Обеспечить возможность сравнения рабочих по заданным параметрам. Обеспечить возможность подбора рабочего для конкретного задания.

3. Обеспечить возможность сравнения помещений по заданным параметрам. Также осуществить возможность подбора помещения под определенные производственные задачи.

4. Осуществить возможность составления расписания отпусков рабочих. Необходимо предусмотреть такие параметры рабочих как специализация и квалификация. Рабочие с одинаковыми специализациями не могут уходить в отпуск одновременно. В программном средстве должен быть предусмотрен календарь.

5. Осуществить возможность составления графика планового ремонта оборудования. Одновременно 2 однотипных оборудования находиться на ремонте не могут. В программном средстве должен быть предусмотрен календарь.

6. Добавить в программное средство информацию о складских помещениях. Обеспечить возможность расчета загруженности склада в зависимости от:

- поставленного производственного плана;
- прогнозируемого спроса по месяцам (задается пользователем);
- возможностей склада (автоматизированный или нет).

В программном средстве должен быть предусмотрен календарь.

7. Добавить в программное средство информацию о транспортной системе предприятия (транспортные тележки, автоматические погрузочно-разгрузочные системы, электрокары и т.д.). Обеспечить возможность расчета планового выпуска продукции с использованием конкретного типа транспортной системы. Расчет должен показывать в какие сроки и какой объем продукции возможно изготовить. В программном средстве должен быть предусмотрен календарь.

8. Реализовать возможность подбора производственного помещения, в зависимости от производственного задания. Должно учитываться требуемое оборудование, его размеры и т.д.

3б. Проверочная работа.

Проверяемые результаты: П1.Р1.ПВК-1, П2.Р1.ПВК-1, П1.Р4.ПВК-1, П1.Р5.ПВК-1, П1.Р1.ПВК-7, П2.Р1.ПВК-7, П1.Р1.ПВК-7, П1.Р1.ПВК-7, П1.Р1.ПВК-7

Проверочная работа представляет собой задания для индивидуального выполнения. Примеры заданий на 5 семестр представлены ниже.

Вариант №1.

Написать код программы на C#, реализующую метод сортировки двухпутевыми вставками, являющейся некоторой модификацией сортировки простыми вставками, являющейся некоторой модификацией сортировки простыми вставками в следующем виде. Отдельно вводится некоторый класс, содержащий выходной массив размером n . Этот выходной массив действует как некоторая циклическая очередь. Элемент $x(1)$ помещается в средний этого массива. Когда непрерывная группа элементов находится в этом массиве, место для нового элемента отводится при помощи сдвига всех элементов с меньшими значениями на один шаг влево или элементов большего значения на один шаг вправо. Выбор, в какую сторону выполнять сдвиг, зависит от того, какой из них приведет к меньшему количеству сдвигов.

Вариант №2.

Написать код программы на C#, реализующую сортировку методом "четных и нечетных транспозиций" которая выполняется следующим образом. Выполняйте несколько раз просмотры файла. На первом просмотре сравнивайте $x(i)$ с $x(i+1)$ для всех нечетных $i(1,3\dots)$. Каждый раз, когда $x(i) > x(i+1)$ выполняйте транспозицию этих двух элементов (т.е. меняйте их местами). Продолжайте эти просмотры до тех пор пока файл не будет отсортирован определите критерий окончания этой сортировки.

Вариант №3.

Написать код программы на C#, в которой генерируется случайно заданный файл размером 100 при использовании стандартной библиотечной функции (генератора случайных чисел). Реализуйте сортировку "Методом пузырька" и "Шейкер-сортировку" таким образом, чтобы накапливалась информация о действительном числе сравнений и действительном числе сделанных транспозиций.

Вариант №4.

Написать код программы на C#, в которой генерируется несколько файлов размером 10, 100 и 1000 при использовании стандартной библиотечной функции – генератора случайных чисел. Реализуйте сортировку методом двоичных включений. Измерьте затраты времени для каждого из этих файлов.

Вариант №5.

Написать код программы на C#, в которой генерируются несколько файлов размером 10, 50, 100 и 1000 при использовании стандартной библиотечной функции – генератора случайных чисел. Реализуйте сортировку методом Шелла, используя соответствующую последовательность шагов. Измерьте затраты времени для каждого из этих файлов.

Вариант №6.

Написать код программы на C#, реализующей метод "быстрой сортировки". Модифицируйте программу "быстрой сортировки" так, чтобы использовалась сортировка "методом пузырька", если некоторый подмассив является небольшим. Определите, используя реальные подсчеты так, чтобы эта смешанная стратегия была более эффективной, чем обычная "быстрая сортировка".

Вариант №7.

Написать код программы на C#, в которой генерируется несколько файлов размером 10, 100 и 1000 при использовании стандартной библиотечной функции – генератора случай-

ных чисел. Реализуйте сортировку этих файлов методом прямого включения и методом двоичных включений. Измерьте затрат времени для каждого из этих файлов при сортировке методом прямого включения и методом двоичных включений.

Вариант №8.

Написать код программы на C#, в которой генерируется несколько файлов размером 10, 100 и 1000 при использовании стандартной функции – генератора случайных чисел. Реализуйте сортировку методом прямого включения и методом прямого выбора. Измерьте затраты времени для каждого из этих файлов при сортировке методом прямого включения и методом прямого выбора.

Вариант № 9.

Предположим, что некоторый файл содержит целые числа в указанные от a до b , причем многие числа повторяются несколько раз. Написать код программы на C#, реализующей сортировку методом распределенного подсчета, которая выполняется следующим образом. Заводится некоторый массив `number` размером $b-a+1$ и элементу `number(i-a+1)` присваивается значение, равное тому, сколько раз число i появляется в данном файле. Затем значения в файле. Затем значения в файле соответственно переустанавливаются. Таким образом необходимо реализовать сортировку массива X методом распределенного подсчета. Массив X имеет размер и содержит целые числа в диапазоне от a до b .

Вариант №10.

Написать код программы на C#, в которой генерируется несколько файлов размером 10, 100, 1000 при использовании стандартной библиотечной функции – генератора случайных чисел. Реализуйте сортировку этих файлов "метод пузырька" и "шейкерную" сортировку таким образом, чтобы накапливалась информация о действительном числе сделанных перестановок (транспозиций).

Вариант №11.

Напишите код программы на C#, которая сортирует файл, сначала 1) применяя поразрядную сортировку для r старших значений цифр (где r является некоторой заданной константой), а затем 2) использует метод сортировки простыми вставками для всего файла.

Вариант №12.

Напишите код программы на C#, в которой: 1) сгенерируйте два случайно заданных файла размером 10 при использовании стандартной библиотечной функции – генератора случайных чисел; 2) осуществите объединение этих двух файлов путем сортировки простым слиянием.

Вариант №13.

Напишите код программы на C#, в которой происходит объединение отсортированных массивов a и b в массив c методом бинарного слияния. Метод сортировки путем бинарного слияния состоит в следующем. Пусть l_a и l_b – это числа элементов в массивах a и b соответственно, и предположим, что $l_a \geq l_b$. Разделим массив a на $(l_b + 1)$ примерно равных подмассивов. Сравним $b(1)$ с наименьшим элементом во втором подмассиве массива a . Если $b(1)$ меньше, тогда найдем $a(i)$ такой, что $a(i) \leq b(1) \leq a(i+1)$, при помощи бинарного поиска в первом подмассиве. Выведем в массив c все элементы первого подмассива до элемента $a(i)$ включительно, а затем выведем в массив c элемент $b(1)$. Повторите этот процесс для элементов $b(2), b(3), \dots, b(j)$, где $b(j)$ оказывается элементом большим, чем наименьший элемент во втором подмассиве. Выведите в массив c все оставшиеся элементы из первого подмассива. Затем сравните $b(j)$ с наименьшим элементом в третьем подмассиве массива a и т. д.

Вариант №14

Напишите код программы на C#, осуществляющей сортировку массива A с помощью "быстрой сортировки". Выбирайте центральный элемент из середины списка. Во время каждого прохода фиксируйте все обмены элементов между нижним и верхним подписками. Осуществите печать состояния последовательности после каждого прохода.

A = 790, 175, 284, 581, 374, 799, 852, 685, 486, 347

Вариант №15

Напишите код программы на C#, в которой: 1) сгенерируйте случайно заданный файл размером 10 при использовании стандартной библиотечной функции – генератора случайных чисел; 2) разбейте файл пополам. Отсортируйте каждую половину с помощью сортировки прямым выбором, а затем осуществить сортировку двух половин методом простого слияния.

Вариант №16

Напишите код программы на C#, в которой сортируется массив целых чисел, изменяющихся от 1 до 20, сгенерированный случайно. Выполните поразрядную сортировку данного массива, используя для представления каждой группы абстрактную очередь. Осуществляйте печать после каждого просмотра.

Вариант №17

Напишите код программы на C#, в которой генерируются два случайно заданных файла размером 100. Необходимо объединить их методом сортировки простым слиянием. Осуществляйте печать после каждого просмотра.

Вариант №18

Напишите код программы на C#, в которой сортируется символьная последовательность V, B, L, A, Z, Y, C, H, S, B, S, H методом прямого выбора и методом прямого включения. Осуществляйте печать после каждого просмотра.

Вариант №19

Напишите код программы на C# для отображения данных в табличные индексы с использованием хеш-функции $hashf(x) = x \% 11$. Данные вставляются в таблицу в следующем порядке: 11, 13, 12, 34, 38, 33, 27, 22. Постройте хеш-таблицу методом открытой адресации.

Методика проведения

Проверочная работа проводится в аудитории на шестом лабораторном занятии, в письменной форме (групповой способ) в течение 45 минут, без использования справочной литературы и средств коммуникации, результат – на следующем занятии.

3в. Задания на лабораторные работы.

При защите лабораторных работ проверяются все результаты. Приведены варианты задач для выполнения лабораторных работ.

1. Составить программу для решения системы двух линейных уравнений с двумя неизвестными.

Указание

Значение неизвестных x, y системы уравнений

$$\begin{cases} ax + by = c, \\ dx + ey = f \end{cases}$$

находятся по формулам

$$\Delta = ae - bd, \quad x = \frac{ce - bf}{\Delta}, \quad y = \frac{af - cd}{\Delta}.$$

Проверить, что $\Delta \neq 0$.

2. Подсчитать, сколько очков набрала команда в первом круге чемпионата по хоккею, если известно, что m встреч она выиграла, n встреч проиграла, k встреч закончились ничьими.

Указание

За выигрыш команда получает два очка, за ничью — 1 очко, за проигрыш — 0 очков.

3. Известны длины сторон a, b, c треугольника. Вычислить высоты этого треугольника.

Указание

Высоты треугольника вычисляются по формулам:

$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)},$$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)},$$

$$h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)},$$

$$\text{где } p = \frac{a+b+c}{2}.$$

4. Составить программу для вычисления времени t встречи автомобилей, движущихся равноускоренно навстречу друг другу, если известны их скорости V_1 и V_2 , ускорения a_1 и a_2 и начальное расстояние S между ними.

Указание.

Расстояние S_1 , пройденное первым автомобилем, вычисляется по формуле

$$S_1 = V_1 t + \frac{a_1 t^2}{2};$$

расстояние, пройденное вторым автомобилем, вычисляется по формуле

$$S_2 = V_2 t + \frac{a_2 t^2}{2}.$$

Время t встречи автомобилей определяется из уравнения

$$V_1 t + \frac{a_1 t^2}{2} = S - \left(V_2 t + \frac{a_2 t^2}{2} \right),$$

$$\text{откуда } t = \frac{-(V_1 + V_2) + \sqrt{(V_1 + V_2)^2 + (a_1 + a_2)2S}}{a_1 + a_2}.$$

5. Найти x из пропорции $\frac{a+b}{x} = \frac{b-c}{a+c}$.
6. Сколько процентов от $A+B-C$ приходится на A ? На B ? На C ?
7. Составить программу вычисления идеального веса человека по его росту, при условии, что идеальный вес(кг) = рост(см) - 100.
8. Вы положили деньги в сбербанк на срочный депозит на квартал из расчета 24% годовых. Составить программу, которая вычислит причитающуюся вам сумму через 4 месяца.

9. Розничная цена мужского костюма составляет R рублей. Наценка магазина составляет $T\%$ от оптовой цены. Составить программу определения оптовой цены костюма.
10. Зарплата сотрудника частной фирмы r рублей в месяц. Сколько денег он получит за полгода после вычета налогов в размере $t\%$ ежемесячно и $s\%$ за полгода?
11. Даны координаты вершин некоторого треугольника. Вычислить его периметр.
12. Смешано V_1 литров воды температуры t_1 с V_2 литрами воды температуры t_2 . Составить программу вычисления объема и температуры образованной смеси.
13. Реализовать функцию поиска элемента E в односвязном списке L .
14. Подсчитать число максимальных элементов списка.
15. В списке A хранится информация о людях (фамилия, имя, отчество, профессия). Имеется список B , содержащий перечень профессий. Удалить из списка A тех людей, чья профессия не указана в списке B .
16. Дан список слов. Из каждой группы подряд идущих одинаковых слов оставить только одно.
17. Дан текстовый файл. Распечатать слова, имеющие максимальную длину.
18. Дан список вещественных чисел. Проверить, упорядочены ли числа по возрастанию или по убыванию.
19. Дан список вещественных чисел. Для каждого элемента списка напечатать число отрицательных элементов, следующих за ним.
20. Реализовать проект "Частотный словарь". В качестве обрабатываемого текста можно использовать, например, модули этого проекта. Результатом должно явиться перечисление всех "слов" в алфавитном порядке с частотой их появления. Отметить, что частота появления таких слов, как *begin* и *end*, всегда одинакова.
21. Создать приложение, проверяющее правильность расстановки скобок в арифметическом выражении.
22. Даны два стека. Используя процедуры ИзСтека, ВСтек и функцию СтекПуст подсчитать общее число элементов в стеках. В качестве вспомогательных структур разрешается использование переменных целых типов. Алгоритм должен предусматривать восстановление исходного расположения элементов в стеках.
23. Дан текстовый файл A . Переписать его содержимое в файл B , перенося при этом в конец каждой строки все входящие в нее знаки препинания.
24. Даны две очереди X и Y , содержащие вещественные числа. Из каждой очереди одновременно извлекается по одному числу, x и y соответственно. Если $x < y$, то число $(x + y)$ помещается в конец очереди X , иначе число $(x - y)$ помещается в конец очереди Y . Необходимо определить число шагов, через которое одна из очередей станет пустой.
25. Создать очередь, информационные поля которой содержат целые числа из текстового файла. Вставить в список новый элемент с информационным полем d после каждого элемента с четным числом в информационном поле.
26. Создать очередь из 20 элементов, информационные поля которой содержат случайные числа из интервала $[-30, 30]$. Вставить в список новый элемент с информационным полем 100 за каждым отрицательным числом.
27. Создать очередь, информационные поля которой содержат строки из файла. Удалить из списка элементы, информационные поля которых равны строке S .
28. Подсчитать число узлов в заданном двоичном дереве.
29. Реализовать рекурсивную процедуру печати всех элементов заданного двоичного дерева.

30. Описать логическую функцию, определяющую, есть ли в заданном двоичном дереве хотя бы два одинаковых элемента.
31. Описать процедуру, которая для заданного N строит двоичное дерево, в котором N полных уровней и на каждом уровне i располагаются узлы, информационные части которых равны i .

Шкала оценивания:

Лабораторная работа считается зачтенной, если выполнена практическая часть, проработан теоретический материал и проведена защита, иначе работа считается не зачтенной.

3г. Вопросы к зачету.

Проверяемые результаты: П1.Р1.ПК-3, П2.Р1.ПК-3, П1.Р2.ПК-3, П2.Р2.ПК-3, П1.Р3.ПК-3, П2.Р3.ПК-3, П1.Р1.ПВК-5, П1.Р2.ПВК-5, П1.Р1.ПВК-7, П1.Р2.ПВК-7.

1. Деревья. Терминология деревьев.
2. Бинарные деревья.
3. Построение бинарного дерева.
4. Рекурсивные методы прохождения деревьев.
5. Использование алгоритмов прохождения деревьев.
6. Бинарные деревья поиска.
7. Работа с бинарным деревом поиска.
8. Интерпретатор. Основные понятия и этапы.
9. Алгоритм работы интерпретатора.
10. Классы в работе простейшего интерпретатора.
11. Динамические структуры данных.
12. Рекурсивный доступ к данным.
13. Динамический список.
14. Упорядоченный список.
15. Слияние списков.
16. Двусвязные и кольцевые списки.
17. Динамический стек.
18. Очереди.
19. Классовая структура программного средства. Конструктор и деструктор. Пример применения в `C#`.
20. Коллекции в `C#`.

Также на зачете проверяются практические навыки. Студентам требуется выполнить в рамках зачета одну из задач лабораторного практикума.

3д. Вопросы к экзамену

1. Трансляторы, интерпретаторы и компиляторы
2. Стадии работы компилятора
3. Определение формальной грамматики и языка

4. Типы формальных языков и грамматик. Классификация по Хомскому
5. Вывод в КС-грамматиках и правила построения дерева вывода
6. Синтаксический разбор
7. Левый и правый выводы
8. Неоднозначные и эквивалентные грамматики
9. Способы задания схем грамматик. Форма Наура-Бэкуса
10. Итерационная форма
11. Синтаксические диаграммы
12. Приведенные грамматики
13. Непроизводящие символы
14. Недостижимые символы
15. Беспольные символы
16. Исключение леворекурсивных правил
17. Исключение цепных правил
18. Преобразование неукорачивающих грамматик
19. Магазинные автоматы
20. Представление арифметических, алгебраических и логических выражений в виде польской записи
21. Введение
22. Регулярные выражения
23. Элементы регулярных выражений
24. Конкатенация литералов
25. Экранирование метасимволов
26. Коды символов и литеральные константы
27. Якорные метасимволы
28. Выбор альтернатив
29. Классы символов
30. Квантификаторы
31. Обработка контекста
32. Группировка и ограничение регулярных фрагментов

КРИТЕРИИ ОЦЕНИВАНИЯ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ **(Зачет с оценкой)**

Итоговой аттестацией по дисциплине «Основы цифровых производств» является зачет (5 семестр) и экзамен (6 семестр). Экзамен проводится по принятой пятибалльной шкале («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»). Экзамен с оценкой проводится в письменной форме. На экзамене оценкой выдается 1 билет, содержащий 2 вопроса.

Оценка «отлично» - за правильные ответы на все вопросы билета и дополнительные вопросы.

Оценка «хорошо» - за правильные ответы на вопросы билета и неудовлетворительные ответы на дополнительные вопросы.

Оценка «удовлетворительно» - за правильный ответ на 1 вопрос.

Оценка «неудовлетворительно» - при отсутствии ответов на вопросы билета.

На зачете оценкой выдается 1 билет, содержащий 1 вопрос и 1 задачу.

Оценка «зачтено» - за правильный ответ на вопрос билета и решение задачи.

При проведении отчетности в форме тестирования:

«отлично» - 80-100% правильных ответов;

«хорошо» - 65-79% ответов;

«удовлетворительно» («зачет») - 50-64%;

«неудовлетворительно» («незачет») - до 50% правильных ответов.

Оценивание курсовой работы

При выполнении всех пунктов курсовой работы – выставляется оценка «отлично».

При выполнении только части индивидуального задания – оценка «хорошо».

При выполнении только общего задания – оценка «удовлетворительно».