

«

»

“ ”

“ ”

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Языки программирования

: 09.03.01

, :

: 1, : 2

		2
1	()	4
2		144
3	, .	64
4	, .	18
5	, .	0
6	, .	36
7	, .	18
8	, .	2
9	, .	8
10	, .	80
11	(, ,)	
12		

(): 09.03.01

5 12.01.2016 ., : 09.02.2016 .

: 1,

(): 09.03.01

, 7 20.06.2017

, 6 21.06.2017

:

, . . .

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; в части следующих результатов обучения:	
11.	
11.	
5.	
Компетенция ФГОС: ПК.3 способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности; в части следующих результатов обучения:	
6.	
Компетенция НГТУ: ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"; в части следующих результатов обучения:	
4.	

2.

2.1

	(
--	---	--

.3. 6	
1.о парадигмах проектирования сложных модульных программ;	; ;
.5. 11	
2.о распределении памяти при компиляции и выполнении программы и динамическом выделении памяти для данных;	; ;
3.о стандартных и пользовательских типах данных в программе;	; ;
.3. 6	
4.о основных понятиях структурного программирования (управляющие структуры программирования, нисходящее пошаговое проектирование алгоритма и данных, схема иерархии функций программы);	; ;
.5. 11	
5.о основных понятиях объектно-ориентированного программирования (объект, класс, метод, инкапсуляция, наследование, полиморфизм, интерфейс) и механизмах их реализации в системах программирования;	; ;
6.о абстрактных базовых классах и их интерфейсах, обеспечивающих реализацию динамического полиморфизма;	; ;
.9. / . 4	
7.о технологии создания приложений с использованием языков программирования высокого уровня Си С++ и Паскаль;	; ;
8.о принципах организации параллельной и сетевой обработки данных в системах программирования на языке С++ и Паскаль;	; ;
9.о современных технологиях разработки надежного программного кода;	; ;
.5. 11	

10.способы определения стандартных и пользовательских типов данных;	;	;
11.основы модульного проектирования программ на языках С и Паскаль(время жизни и область действия переменных, порядок связывания функций, назначение и структуру заголовочных файлов, объектных модулей и файла проекта);	;	;
12.методы динамического размещения в памяти переменных и массивов стандартных типов, а также экземпляров структур, массивов структур, объектов классов и массивов объектов классов;	;	;
13.способы формирования массивов указателей и алгоритмы работы с ними;	;	;
14.динамические линейные структуры данных (списки, очереди, стеки), способы их представления в памяти и обработки в программе;	;	;
.9. / . 4		
15.принципы программирования рекурсивных функций и их использования в алгоритмах решения задач поиска;	;	;
.5. 11		
16.динамические нелинейные структуры данных (графы, деревья), способы их представления в памяти и обработки в программе;	;	;
17.простые алгоритмы поиска и сортировки данных;	;	;
18.принципы перегрузки унарных и бинарных операций;	;	;
19.методы и средства обработки исключительных ситуаций;	;	;
.5. 11		
20.создавать, компилировать и отлаживать программный код на языках С и С++ и Паскаль в современной инструментальной среде визуальной разработки программ;	;	;
21.применять технологию структурного программирования для формального описания проблемной области в терминах "действие - функция" и нисходящего пошагового проектирования иерархии функции в качестве основы адекватной модульной программы решения проблемы;	;	;
.5. 5		
22.применять технологию объектно-ориентированного программирования для формального описания проблемной области в терминах "совокупность данных - класс" и последующего итерационного формирования иерархии классов в качестве основы адекватной модульной программы решения проблемы.	;	;

3.

3.1

: 2			
:			
1.	0	2	1, 10, 11, 15, 17, 2, 20, 21, 3, 4, 7

2.	0	2	11, 13, 14, 20, 21, 4, 7, 9
3.	0	2	1, 11, 15, 16, 17, 2, 20, 21, 4, 7, 9
:			
4.	0	4	1, 10, 11, 15, 16, 17, 20, 21, 3, 4, 6, 7, 8, 9
:			
5.	0	4	10, 11, 14, 15, 16, 18, 19, 21, 3, 4, 7, 8
:			
6. ++.	0	4	10, 18, 19, 22, 3, 5, 6, 7, 8

3.2

	,	.		
: 2				
:				
1.	2	4	1, 10, 11, 12, 15, 17, 20, 21, 3, 4, 7, 9	- ; - ; (-) ; - ; , () ; - ; , ; - ; - ;

2.	2	4	11, 12, 13, 14, 17, 2, 20, 21, 4, 7, 9	- : , ,
3.	2	4	1, 11, 15, 16, 17, 2, 20, 21, 4, 7, 9	: - , ,
:				
4.	4	4	1, 10, 11, 15, 16, 17, 20, 21, 3, 4, 7, 8, 9	: , , ,
:				
5.	2	6	10, 11, 14, 17, 18, 19, 20, 21, 3, 4, 7, 8	.
6.	2	6	10, 11, 15, 16, 17, 18, 19, 2, 20, 21, 3, 4, 6, 7, 9	" "
:				
7.	4	8	10, 14, 15, 16, 17, 18, 19, 22, 3, 5, 6, 7, 9	- (++).

<p>230100 " . - ;[. . .] . - , 2013. - 18, [2] .. - : http://elibrary.nstu.ru/source?bib_id=vtls000179535</p>				
5		1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, 22, 3, 4, 5, 6, 7, 8, 9	40	0
<p>2013. - 18, [2] .. - : http://elibrary.nstu.ru/source?bib_id=vtls000179535</p>				

5.

(. 5.1).

5.1

	e-mail
	e-mail;

6.

(),

15- ECTS.

. 6.1.

6.1

: 2		
<i>Лабораторная: Рекурсивные алгоритмы</i>	4	5
<i>Лабораторная: Односвязные списки</i>	4	5
<i>Лабораторная: Бинарные деревья</i>	4	5
<i>Лабораторная: Текстовые файлы</i>	4	5
<i>Лабораторная: Шаблоны функций и шаблоны структур</i>	4	5
<i>Лабораторная: Обобщенные типы в Паскале</i>	4	5
<i>Лабораторная: Объектно-ориентированное программирование на C++</i>	8	10
<i>Лабораторная: Шаблоны классов</i>	9	10
<i>РГЗ:</i>	9	10
<i>Экзамен:</i>	0	40

		/		
.5	11.	+	+	+
	11.	+	+	+
	5.		+	
.3	6.	+	+	+
	.9. / 4.	+	+	

1

7.

1. Никлаус Вирт Алгоритмы и структуры данных [Электронный ресурс]/ Вирт Никлаус— Электрон. текстовые данные.— Саратов: Профобразование, 2017.— 272 с.— Режим доступа: <http://www.iprbookshop.ru/63821.html>.— ЭБС «IPRbooks»

2. Романов Е. Л. Си/Си ++. От дилетанта до профессионала [Электронный ресурс] : электронное учебное пособие : для 1-2 курсов направления 230100 "Информатика и вычислительная техника / Романов Е. Л. - Новосибирск, 2010. - 1 электрон. опт. диск (CD-ROM). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024. - Загл. с этикетки диска. - Рег. свидетельство №18891.

3. Павловская Т. А. Паскаль. Программирование на языке высокого уровня : [учебник по направлению "Информатика и вычислительная техника"] / Т. А. Павловская. - СПб. [и др.], 2010. - 460 с.

1. Керниган Б. У. Язык программирования С : пер. с англ. / Брайан Керниган, Деннис Ритчи. - М. [и др.], 2006. - 289 с. : ил.

2. Пярнпуу А. А. Программирование на современных алгоритмических языках : учебное пособие для вузов / А. А. Пярнпуу. - М., 1990. - 383 с. : ил.

3. Тассел Д. В. Стиль, разработка, эффективность, отладка и испытание программ / Д. Ван Тассел ; пер. с англ. Е. К. Масловского, В. А. Прониной под ред. Э. А. Трахтенгерца. - М., 1985. - 332 с. : ил.

4. Лингер Р. Теория и практика структурного программирования / Р. Лингер, Х. Миллс, Б. Уитт ; пер. с англ. под ред. С. Д. Пашкеева. - М., 1982. - 406 с. : ил.

5. Гудман С. Введение в разработку и анализ алгоритмов : пер. с англ. / С. Гудман, С. Хидетниemi; под ред В. В. Мартынюка. - М., 1981. - 366 с. : ил.

6. Кнут Д. Э. Искусство программирования. Т. 1, вып. 1 / Дональд Э. Кнут. - М. [и др.], 2007. - 150 с.

7. Кнут Д. Э. Искусство программирования. Т. 2 : пер. с англ. / Дональд Э. Кнут ; под общ. ред. Ю. В. Козаченко. - М. [и др.], 2007. - 828 с. : ил.

8. Кнут Д. Э. Искусство программирования. Т. 3 : пер. с англ. / Дональд Э. Кнут ; под общ. ред. Ю. В. Козаченко. - М. [и др.], 2007. - 822 с. : ил.
9. Дейтел Х. М. Как программировать на C++ / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - М., 2007. - 799 с. : ил.
10. Подбельский В. В. Программирование на языке Си : учебное пособие для вузов / В. В. Подбельский, С. С. Фомин. - М., 2005. - 600 с. : ил., табл.
11. Крылов Е. В. Техника разработки программ. В 2 кн. Кн. 1 : [учебник для вузов по направлениям "Информатика и вычислительная техника" и "Техника и технологии"] / Е. В. Крылов, В. А. Острейковский, Н. Г. Типикин. - М., 2007. - 374, [1] с. : ил.

1. Лауферман О. В. Языки программирования [Электронный ресурс]: электронный учебно-методический комплекс [для групп АВТ-х12-х13] / О. В. Лауферман. - Новосибирск, 2016. - Режим доступа : <http://dispace.edu.nstu.ru/didesk/course/show/4752>. - Загл. с экрана.

2. ЭБС НГТУ : <http://elibrary.nstu.ru/>

3. ЭБС «Издательство Лань» : <https://e.lanbook.com/>

4. ЭБС IPRbooks : <http://www.iprbookshop.ru/>

5. ЭБС "Znanium.com" : <http://znanium.com/>

6. :

8.

8.1

1. Программирование : методические указания к лабораторным работам для 2 курса АВТФ направления 230100 "Информатика и вычислительная техника" дневного отделения / Новосиб. гос. техн. ун-т ; [сост. Р. Г. Шахмаметов]. - Новосибирск, 2013. - 18, [2] с.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179535

8.2

1 Microsoft Windows

2 Microsoft Office

9.

1	()

1	(Internet)

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра автоматизированных систем управления

“УТВЕРЖДАЮ”
ДИРЕКТОР ИСТР
д.соц.н., профессор Л.А. Осьмук
“ ___ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Языки программирования

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль:
Автоматизированные системы обработки информации и управления в социальной сфере

1. **Обобщенная структура фонда оценочных средств учебной дисциплины**

Языки

Обобщенная структура фонда оценочных средств по дисциплине программирования приведена в Таблице 1.

Таблица 1

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	з11. знать методы и средства проектирования программного обеспечения	Бинарные деревья. Рекурсивное определение узла бинарного дерева. Рекурсивные алгоритмы прямого, симметричного и обратного обхода бинарного дерева. Поиск в бинарном дереве. Записи. Файлы. Потоки ввода - вывода. Открытие и закрытие потока. Стандартные потоки. Библиотека функций для работы с файлами. Файлы последовательного и произвольного доступа. Создание файла последовательного доступа. Чтение файла последовательного доступа. Создание файла произвольного доступа. Произвольная запись данных в файл произвольного доступа. Последовательное чтение файла произвольного доступа. Классы и динамическая память. Линейные списки. Основные виды и способы реализации. Линейный список как абстрактный тип данных. Односвязный список. Операции включения в список и исключения из списка. Сортировка списка. Очереди. Стеки. Двусвязные списки. Циклические списки. Обобщенные типы в Паскале. Объектно-ориентированное программирование на языке C++. Объектно-ориентированное программирование (C++). Односвязные списки Рекурсивные алгоритмы Рекурсивные определения и алгоритмы. Рекурсия. Рекурсивное определение структур данных. Рекурсивная функция. Доказательство сходимости рекурсивного алгоритма методом математической индукции. Программирование рекурсивных алгоритмов. Использование рекурсивных алгоритмов в задачах поиска. Оценка сложности рекурсивных алгоритмов. Связные списки. Бинарные деревья. Текстовые файлы. Текстовые файлы и динамические структуры данных. Шаблонные функции и шаблоны структур. Шаблонные функции. Шаблоны структур. Шаблонные списковые структуры. Шаблоны структур бинарных деревьев. Перегрузка операций. Шаблоны функций и шаблонные функции.	Контрольная работа «Односвязные списки» Отчет по лабораторной работе №1-8, РГЗ, разделы №1-8	Экзамен, вопросы №1-15
ОПК.5	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Классы и динамическая память. Объектно-ориентированное программирование на языке C++.	РГЗ, разделы №1-8	Экзамен, вопросы №14-15

ОПК.5	у11. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	Бинарные деревья. Обобщенные типы в Паскале. Односвязные списки Рекурсивные алгоритмы Связные списки. Бинарные деревья. Текстовые файлы. Текстовые файлы и динамические структуры данных. Шаблоны функции и шаблоны структур. Шаблоны функций и шаблонные функции.	Отчет по лабораторной работе №2-6, РГЗ, разделы №1-8	Экзамен, вопросы №1-8
ПК.3/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	34. знать основы методологии создания программного обеспечения	Бинарные деревья. Бинарные деревья. Рекурсивное определение узла бинарного дерева. Рекурсивные алгоритмы прямого, симметричного и обратного обхода бинарного дерева. Поиск в бинарном дереве. Записи. Файлы. . Потоки ввода - вывода. Открытие и закрытие потока. Стандартные потоки. Библиотека функций для работы с файлами. Файлы последовательного и произвольного доступа. Создание файла последовательного доступа. Чтение файла последовательного доступа. Создание файла произвольного доступа. Произвольная запись данных в файл произвольного доступа. Последовательное чтение файла произвольного доступа. Линейные списки. Основные виды и способы реализации. Линейный список как абстрактный тип данных. Односвязный список. Операции включения в список и исключения из списка. Сортировка списка. Очереди. Стеки. Двусвязные списки. Циклические списки. Обобщенные типы в Паскале. Рекурсивные алгоритмы Рекурсивные определения и алгоритмы. Рекурсия. Рекурсивное определение структур данных. Рекурсивная функция. Доказательство сходимости рекурсивного алгоритма методом математической индукции. Программирование рекурсивных алгоритмов. Использование рекурсивных алгоритмов в задачах поиска. Оценка сложности рекурсивных алгоритмов. Списки. Бинарные деревья. Текстовые файлы. Шаблоны функции. Шаблоны структур. Шаблоны списковые структуры. Шаблоны структур бинарных деревьев. Перегрузка операций.	Отчет по лабораторной работе №1-8, РГЗ, разделы №1-8	Экзамен, вопросы №1-15
ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"	34. знать основы системного программирования	Бинарные деревья. Бинарные деревья. Рекурсивное определение узла бинарного дерева. Рекурсивные алгоритмы прямого, симметричного и обратного обхода бинарного дерева. Поиск в бинарном дереве. Обобщенные типы в Паскале. Объектно-ориентированное программирование на языке C++. Объектно-ориентированное программирование (C++). Односвязные списки Рекурсивные алгоритмы Связные списки. Бинарные деревья. Текстовые файлы. Текстовые файлы и динамические структуры данных. Шаблоны функции. Шаблоны структур. Шаблоны списковые структуры. Шаблоны структур бинарных деревьев. Перегрузка операций. Шаблоны функций и шаблонные функции. Шаблоны структур.	Отчет по лабораторной работе №2-4,7-8, РГЗ, разделы №1-8	Экзамен, вопросы №1-8

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 2 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК.

Экзаменационный билет содержит:

1. Теоретический вопрос
2. Задача по теме «Динамические структуры данных»
3. Задача по теме «Текстовые файлы»

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 2 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт экзамена

по дисциплине «Языки программирования», 2 семестр

1. Методика оценки

Экзамен проводится в письменной форме, по билетам.

Билет формируется по следующему правилу:

1. вопрос выбирается из диапазона вопросов 1-15 (п. 4);
2. задача по теме «Динамические структуры данных» (п. 4);
3. задача по теме «Текстовые файлы» (п. 4).

В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

В течение семестра дважды проводится промежуточная аттестация по дисциплине: тестирование на 7 и 12 учебных неделях (контрольные недели). Примеры теста №1 и теста №2 приведены в приложениях к ФОС.

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № 10

к экзамену по дисциплине «Языки программирования»

1. Функции с переменным количеством параметров в языке C/C++. Примеры.
2. Определить, входит ли список L1 в L2.
3. Даны два текстовых файла Ф1 и Ф2. Скопировать в файл Ф3 строки в следующем порядке: первая строка из файла Ф1, первая строка из файла Ф2, вторая строка из файла Ф1, вторая строка из файла Ф2 и так далее.

Утверждаю: зав. кафедрой _____ профессор Гриф М.Г.
(подпись) (дата)

2. Критерии оценки

Ответ на экзаменационный билет считается **неудовлетворительным**, если студент не дает ответа на вопрос и не может решить задачу №1.
оценка составляет 24-49 баллов.

Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент полностью справился с теоретическим вопросом и решил задачу №1, ответил на дополнительные вопросы.

оценка составляет 50-66 баллов.

Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент полностью справился с теоретическим вопросом, решил задачи №1 и №2, но допустил некоторое количество ошибок (недочетов), ответил на дополнительные вопросы.

оценка составляет 67-86 баллов.

Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент полностью справился с теоретическим вопросом, решил задачи №1 и №2 без ошибок, ответил на дополнительные вопросы.

оценка составляет 87-100 баллов.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, таблице 1.

Таблица 1

№	Вид учебной деятельности	Мак. балл
1	Лабораторная работа №1. Рекурсивные алгоритмы	10
2	Лабораторная работа №2. Списки: стеки, очереди	5
3	Лабораторная работа №3. Бинарные деревья	10
4	Лабораторная работа №4. Текстовые файлы и динамические структуры данных	10
5	Лабораторная работа №5. Шаблоны функций и шаблонные функции. Шаблоны структур	5
6	Лабораторная работа № 6. Обобщенные типы в Паскале	10
9	РГЗ	10
6	Экзамен	40
	Итого	100

В таблице 2 представлена шкала перехода от 100 - балльной шкалы к системе оценок ECTS и к традиционной шкале оценок.

Таблица 2

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки	
		отлично	зачтено
90-100	A+(97)	отлично	зачтено
	A(94)		
	A-(90)		
80-89	B+(87)	хорошо	
	B(84)		
	B-(80)		
70-79	C+(77)		

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки	
	C(74)		
	C-(70)	удовлетворительно	
60-69	D+(67)	удовлетворительно	зачтено
	D(64)		
	D-(60)		
50-59	E		
25-49	FX	неудовлетворительно	не зачтено
0-24	F		

4. Вопросы и задачи к экзамену по дисциплине «Языки программирования»

1. Рекурсивные функции и рекурсивные структуры данных. Основные особенности.
2. Динамические структуры данных: списки. Виды списков. Основные операции со списками.
3. Рекурсивные структуры данных: Деревья. Бинарные деревья, сбалансированные, деревья поиска.
4. Основные функции работы с бинарными деревьями: включение вершины, печать содержимого дерева, исключение вершины из дерева.
5. Основные функции работы с идеально сбалансированными деревьями: включение вершины, печать содержимого дерева, исключение вершины из дерева.
6. Абстрактные типы данных и классы. Примеры на C / C++ и Паскале.
7. Параметризованные типы данных в C / C++. Примеры.
8. Обобщённые типы данных в Паскале. Примеры.
9. Указатели на функции. Массивы указателей на функции. Указатели на функции как параметры функций. Примеры.
10. Функции с переменным количеством параметров в C / C++. Примеры.
11. Данные на внешних носителях. Алгоритмы поиска и сортировки.
12. Индексированные файлы. Индекс для индекса. Примеры.
13. Шаблоны функций и шаблонные функции. Примеры.
14. Шаблон класса «Односвязный список».
15. Шаблон класса «Бинарное дерево».

Экзаменационные задачи по теме «Динамические структуры данных»

1. Найти среднее арифметическое значение элементов списка.
2. Перенести в начало списка его последний элемент.
3. Перенести в конец списка его последний элемент.
4. Поменять местами первый и последний элементы списка.
5. Определить количество слов в списке, которые начинаются и заканчиваются на одну букву.
6. Проверить, что каждое следующее слово в списке начинается с последней буквы предыдущего.
7. Определить количество слов в списке, которые совпадают с первым (последним) словом списка.
8. Проверить равенство списков L1 и L2.
9. Проверить, упорядочены ли элементы списка по алфавиту.

10. Определить, входит ли список L1 в L2.
11. Перевернуть список наоборот.
12. Найти максимальный элемент списка (вернуть номер).
13. Определить, есть ли в списке хотя бы два одинаковых значения (вывести его).
14. Поменять местами максимальный и минимальный элементы дерева T, все элементы дерева различны.
15. Напечатать элементы из всех листьев дерева T.

Экзаменационные задачи по теме «Текстовые файлы»

1. Дан текстовый файл Ф1, скопировать из него в файл Ф2 строки, содержащие один и более символов цифр.
2. Дан текстовый файл Ф1, скопировать из него в файл Ф2 строки, длина которых больше заданной.
3. Дан текстовый файл Ф1, скопировать из него в файл Ф2 строки, имеющие четную длину.
4. Дан текстовый файл Ф1, скопировать из него в файл Ф2 строки, содержащие только буквы.
5. Дан текстовый файл Ф1, продублировать (записать в файл Ф2 дважды) строки, длина которых меньше заданной. Остальные строки переписать без изменений.
6. Дан текстовый файл Ф1. Каждую строку нечетной длины перевернуть и записать в файл Ф2.
7. Дан текстовый файл Ф1. Скопировать в файл Ф2 строки с нечетными номерами, а в файл Ф3 строки с четными номерами.
8. Дан текстовый файл Ф1. Скопировать в файл Ф2 строки, содержащие цифры, а в файл Ф3 – все остальные строки, сохраняя исходную последовательность строк.
9. Дан текстовый файл Ф1. Скопировать в файл Ф2 строки, длина которых меньше заданной, а в файл Ф3 строки, длина которых больше или равна заданной.
10. Даны два текстовых файла Ф1 и Ф2. Скопировать в файл Ф3 строки в следующем порядке: первая строка из файла Ф1, первая строка из файла Ф2, вторая строка из файла Ф1, вторая строка из файла Ф2 и так далее.
11. Даны два текстовых файла Ф1 и Ф2. Скопировать в файл Ф3 сначала все строки четной длины из файла Ф1, а затем все строки нечетной длины из файла Ф2.
12. Даны два текстовых файла Ф1 и Ф2. Скопировать в файл Ф3 сначала все строки, содержащие только буквы из файла Ф1, а затем строки, содержащие только буквы из файла Ф2.
13. Переписать непустые строки из файла f1 в файл f2.
14. Элементы файлов f1 и f2 упорядочены по не убыванию. Объединить эти элементы в один файл f3, также упорядоченный по не убыванию.
15. Считая, что непустой текстовый файл разбит на строки, длина каждой из которых не превосходит LEN. Написать функцию, которая дополняет короткие строки пробелами слева, формируя файл, в котором все строки имеют длину LEN.

Паспорт расчетно-графического задания (работы)

по дисциплине «Языки программирования», 2 семестр

1. Методика оценки

В рамках расчетно-графического задания по дисциплине студент:

- выбирает структуры данных;
- проектирует набор функций;
- разрабатывает пользовательский интерфейс.

При выполнении расчетно-графического задания студент:

- *формулирует* постановку задачи;
- *определяет*: схему иерархии программных модулей, функциональное описание каждого модуля, входные/выходные данные;
- *выбирает* структуры данных, форматы данных;
- *строит* блок-схемы обобщенных алгоритмов;
- *оценивает* степень сцепления и связности модулей;
- кодирует, отлаживает и тестирует программу;
- *анализирует* результаты проектирования, кодирования, отладки и тестирования;
- *составляет* расчетно-пояснительную записку объемом 30-40 страниц.

Деятельность обучающегося и критерии оценки ее качества на основных этапах выполнения РГЗ приведены в таблице 2.

Таблица 1

Этап	Деятельность обучающегося	Балл	Критерии качества
1	Выбор задания. Описание постановки задачи.	5	своевременность выбора, полнота системного описания
2	Определение и утверждение схемы иерархии программных модулей	5	своевременность – 2, правильность - 3
3	Функциональное описание каждого модуля, определение входной и выходной информации	15	своевременность – 5, правильность, универсальность –10
4	Описание структур данных. Построение блок-схем алгоритмов. Оценка сцепления и связности модулей.	20	своевременность – 5, универсальность – 15
5	Разработка интерфейса, программная реализация	15	своевременность – 5, качество интерфейса – 10
6	Отладка, тестирование программы.	15	• Качество тестирования – 5;

	Анализ процесса проектирования, кодирования, тестирования программы.		<ul style="list-style-type: none"> • Своевременность–5; • Универсальность- 5
7	Оформление пояснительной записки	15	логичность, связность, полнота описания работы
8	Защита РГЗ	10	качество презентации – 5, качество доклада - 5
	Итого	100	

В зависимости от количества набранных баллов за выполнение соответствующих этапов РГЗ оценивается следующим образом: 85-100 баллов – **отлично (10 баллов** в общей оценке по дисциплине), 70-84 баллов – **хорошо (8 баллов** в общей оценке по дисциплине), 45-69 – **удовлетворительно (5 баллов** в общей оценке по дисциплине), < 45 баллов или отсутствие работающего программного продукта – **неудовлетворительно (2 балла** в общей оценке по дисциплине).

1. Критерии оценки

- Работа считается **не выполненной**, если выполнены не все части РГЗ, отсутствуют этапы выполнения РГЗ со 2 по 8, оценка составляет **2 балла**.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ выполнены не в полном объеме: отсутствуют 4 и 6 этапы выполнения РГЗ, оценка составляет **5 баллов**.
- Работа считается выполненной **на базовом** уровне, если выполнены все этапы, но 4 и 5 этапы выполнены неэффективно, оценка составляет **8 баллов**.
- Работа считается выполненной **на продвинутом** уровне, если все этапы выполнены в полном объеме и защита РГЗ прошла успешно, оценка составляет **10 баллов**.

2. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в таблице 2.

Таблица 2

№	Вид учебной деятельности	Максимальный балл
1	Лабораторные работы №1-8.	50
3	РГЗ	10
4	Экзамен	40
	Итого	100

Таблица соответствия баллов, традиционной оценки и буквенной оценки ECTS приведена в рабочей программе дисциплины.

3. Примерный перечень тем РГЗ(Р)

1. Числа хранятся в файлах. Необходимо скопировать числа в ОП, отсортировать и записать в файл-результат.
 - Числа целые/вещественные – **перегрузка функций**;
 - Количество исходных файлов заранее неизвестно – **функции с переменным количеством параметров**;
 - Метод сортировки выбираем из трех предложенных (сортировка выбором, обменом или вставками) – **указатели на функции**.
2. Числа хранятся в файлах. Необходимо скопировать их в файл-результат, упорядочивая по возрастанию/убыванию (в ОП числа не загружать).

- Числа целые/вещественные – *перегрузка функций*;
 - Количество исходных файлов заранее неизвестно – *функции с переменным количеством параметров*;
 - Сортировка по возрастанию/убыванию – *указатели на функции*;
 - Файлы текстовые/бинарные – *перегрузка функций*.
3. Слова хранятся в файлах. Необходимо скопировать их в файл-результат, упорядочивая по алфавиту (в ОП слова не загружать).
- Количество исходных файлов заранее неизвестно – *функции с переменным количеством параметров*;
 - Сортировка по алфавиту А-Я/Я-А – *указатели на функции*;
 - Файлы текстовые/бинарные (в текстовом файле: слово – это строка, в бинарном: перед словом записана его длина, затем само слово) – *перегрузка функций*.
4. Дроби хранятся в бинарных файлах. Необходимо скопировать дроби в ОП, отсортировать и записать в файл-результат.
- Дробь в файле записана как пара целых чисел (числитель знаменатель). Дроби могут быть правильными и неправильными, хранятся только положительные дроби;
 - Количество исходных файлов заранее неизвестно – *функции с переменным количеством параметров*;
 - Метод сортировки выбираем из двух предложенных (сортировка выбором, обменом) – *указатели на функции*.
5. Дроби хранятся в бинарных файлах. Необходимо скопировать их в файл-результат, упорядочивая по возрастанию/убыванию (в ОП числа не загружать).
- Дробь в файле записана как пара целых чисел (числитель знаменатель). Дроби могут быть правильными и неправильными, хранятся только положительные дроби;
 - Количество исходных файлов заранее неизвестно – *функции с переменным количеством параметров*;
 - Сортировка по возрастанию/убыванию – *указатели на функции*.
6. Слова хранятся в файлах. Необходимо скопировать их в файл-результат, упорядочивая по «весу» (в ОП слова не загружать).
- Количество исходных файлов заранее неизвестно – *функции с переменным количеством параметров*;
 - Сравнение слов осуществляем по «весу» - сумме кодов символов слова (например, слово “words” имеет «вес» $559=119+111+114+100+115$, а слово “return” «весит» $672=114+101+116+117+114+110$, значит слово “return” «больше» слова “words”);
 - Файлы текстовые/бинарные (в текстовом файле: слово – это строка, в бинарном: перед словом записана его длина, затем само слово) – *перегрузка функций*.

8. Контролирующие материалы для аттестации студентов по дисциплине

Итоговый тест 2 семестр (время проведения - 50 мин.)

Дополните определения (в заданиях 1 - 7)

- _____ – другое имя уже существующего объекта.
- Имя массива – _____ на его начало.
- Оперативное управление выделяемой памятью в соответствии с требованиями, предъявляемыми данными, называется _____ памяти.
- Динамические переменные не имеют имени, доступ к ним осуществляется через _____.
- Динамические переменные создаются и уничтожаются _____.
- Функции выделения динамической памяти возвращают указатель на _____ или NULL.
- Для освобождения динамически выделенной памяти используется операция _____.

8. Установите соответствие (укажите буквы)

Структуры данных Особенности

1) Массив

A. Элементы всегда располагаются в последовательных ячейках памяти.

2) Список

B. Операции удаления и вставки осуществляются в любом месте без перемещения элементов в памяти.

C. Порядок элементов поддерживается логически.

D. Структура с произвольным доступом к элементам.

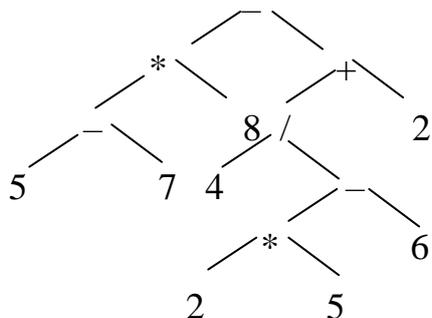
E. Не накладывает ограничений на размер структуры.

Ответ: 1) __, __ 2) __, __, __.

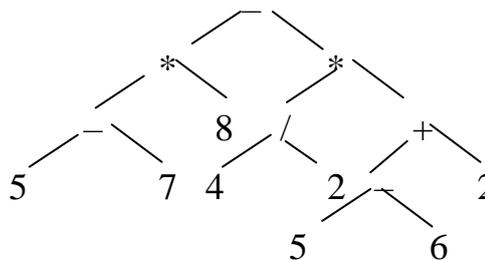
9. Установите соответствие (укажите буквы)

Деревья

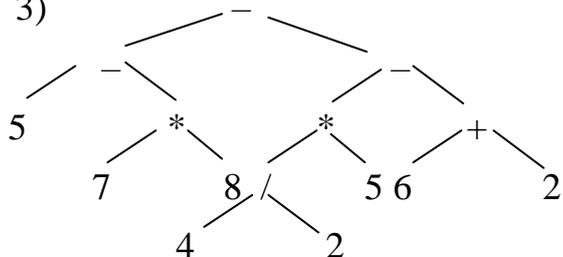
1)



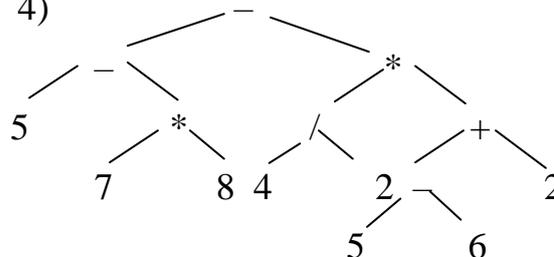
2)



3)



4)



Арифметические выражения

A. $(5 - 7 * 8) - (4 / 2 * 5 - (6 + 2))$

B. $(5 - 7) * 8 - (4 / (2 * 5 - 6) + 2)$

C. $5 - 7 * 8 - (4 / 2 * (5 - 6 + 2))$

D. $(5 - 7) * 8 - 4 / 2 * (5 - 6 + 2)$

Постфиксная запись

a. $5 7 8 * - 4 2 / 5 6 - 2 + * -$

b. $5 7 8 * - 4 2 / 5 * 6 2 + - -$

c. $5 7 - 8 * 4 2 / 5 6 - 2 + * -$

d. $5 7 - 8 * 4 2 5 * 6 - / 2 + -$

Ответ: 1) __, __ 2) __, __ 3) __, __ 4) __, __.

Дополните операторы (в заданиях 10, 11) именами функций:

calloc(), malloc(), realloc(), strdup()

```
#define N 80
#define M 10
struct list
{ long number; struct list *next;};
...
list *head, *qnew, *ptr; char *str, **text;
```

10. ...
str = (char*) _____ (N);
text = (char**) _____ (M, sizeof(char*));
for (i = 0; i < M; i++)
{ gets (str);
text [i] = _____ (str);
if(!text [i]) break; }
...
text = (char**) _____ (text , (2*M)*sizeof (char*));

11. ...
head = (struct list*) _____ (1 , sizeof (struct list));
head->number = 1; head->next = NULL;
qnew = (struct list*) _____ (1 * sizeof (struct list));
ptr = head; ptr -> next = qnew;
qnew->number = 5; ptr = ptr -> next;
...

12. Выберите все верные утверждения (укажите буквы)

- A. Список допускает только последовательный просмотр элементов.
- B. Список дает возможность для быстрого (бинарного) поиска при соответствующей упорядоченности элементов.
- C. Доступ к элементу массива возможен, как через имя, так и через указатель.
- D. Массив является линейным набором элементов с самоадресацией.
- E. Тип и имя одинаковы для всех элементов массива.

Ответ __, __, __.

13. Выберите все верные варианты ответов: «да» или «нет»

Двоичное дерево – это ориентированное дерево, в котором:

- 1) Имеется ровно одна вершина, в которую не входит ни одного ребра – да / нет
это корень дерева;
- 2) В каждую вершину, включая корень, входит одно ребро да / нет
- 3) Из каждой вершины, включая корень, выходит не более трех ребер да / нет
- 4) Значение в любом узле его левого поддерева меньше, а правого да / нет
поддерева – больше значения в его родительском узле
- 5) Используя включения элемента в бинарное дерево, можно да / нет
отсортировать входную последовательность данных по выбранному ключу

Ответ __, __, __, __, __.

14. Выберите все *неверные* утверждения (укажите буквы)

```
struct list
{ int value;          struct list *next; };
list *head, // заголовок списка
      *ptr, // ссылка на текущий элемент
      *pred; // ссылка на предыдущий элемент
```

- A. ptr = head; // текущая ссылка на первый элемент
- B. ptr -> next -> next... // удаление элемента из списка
- C. for (ptr = head ; ptr != NULL; ptr = ptr -> next;) // просмотр списка элементов
- D. ptr = ptr -> next; // переход к следующему элементу
- E. if (ptr -> next == NULL) // проверка, первый элемент в списке?
- F. if (ptr != NULL) // проверка на конец списка
- G. pred -> next = ptr -> next; // ссылка на следующий элемент после текущего

Ответ __, __, __.

15. Выберите все *верные* варианты ответов: «да» или «нет»

- 1) Структуры могут содержать данные только одного типа. да / нет
- 2) Элементы разных структур могут иметь совпадающие имена. да / нет
- 3) Доступ к элементам структуры осуществляется с помощью индекса. да / нет
- 4) Структуры всегда передаются в функцию по ссылке. да / нет
- 5) Структуры могут содержать в качестве элементов массивы, и массивы могут состоять из структур. да / нет

Ответ __, __, __, __, __.

16. Установите соответствие (укажите цифры)

Фрагменты программ

1) for (p = head; p->next != head; p = p->next) if (pnew->number == p->number) return; p ->next = pnew; pnew -> next = head; head = pnew;	2) for (p = head; p->next != head; p = p->next) if (pnew->number > p ->next->number) break; pnew -> next = p -> next; p-> next = pnew;
3) for (p = head; p ->next! = head; p = p ->next) if (pnew->number == p ->number) return; pnew -> next = head; p -> next = pnew;	4) for (p = head; p ->next != head; p = p ->next) if (pnew->number == p ->next->number) break; p -> next = p ->next -> next;
5) for (p = head; p ->next != head; p = p ->next); p ->next = pnew; pnew -> next = head;	6) for (p = head; p ->next != head; p = p ->next); p ->next = p ->next -> next; head = head -> next;

Действия с элементом списка:

- A. Включение в начало
- B. Включение в конец
- C. Включение в середину
- D. Исключение из списка

Ответ: A) __ B) __, __ C) __ D) __, __.

17. Установите соответствие (укажите цифры)

*struct list { char letter; struct list *next; };*

Фрагменты программ

Способы формирования структур данных

- 1) `struct list array[N], *head = array;`
`for(i = 0; i < N-1; i++);`
`array[i].next = array+i+1;`
`array[i].next = NULL;`
- 2) `void Ins(struct list *ptr, char ch)`
`{ struct list *pnew = new list;`
`pnew -> letter = ch; pnew -> next = NULL;`
`ptr -> next = pnew; ptr = pnew;`
`}`
- 3) `struct list el1={'A', NULL},`
`el2={'D', &el1},`
`el3={'M', &el2},`
`*head=&el3;`
- 4) `struct list *plist;`
`plist = new struct list[N];`
`for(i = 0; i < N-1; i++)`
`plist[i] -> next = plist[i+1];`
`plist[i] -> next = NULL;`

- А. Элементы и связи формируются компилятором
- В. Элементы - динамические переменные, связи устанавливаются программно
- С. Элементы - статические структуры данных, связи устанавливаются программно

Ответ: А) __ В) __, __ С) __.

Выберите все неверные утверждения (в заданиях 18, 19) (укажите буквы)

18. Переменная вида *int *pnt;* в соответствии с контекстным определением может быть:

- А. Указателем на массив элементов целого типа.
- В. Указателем на указатель на переменную типа `int`.
- С. Указателем на область памяти, отведенную под элементы целочисленной матрицы.
- Д. Указателем на скалярную переменную типа `int`.
- Е. Константным указателем на переменную целого типа.

Ответ __, __.

19. Динамическое распределение памяти используется, когда:

- А. Размер и количество используемых структур данных определяется на этапе компиляции программы.
- В. Используются рекурсивные структуры данных.
- С. Заранее неизвестна размерность обрабатываемых данных.
- Д. Типы и состав элементов используемых структур данных заранее неизвестны.
- Е. Необходимо изменять требуемый объем памяти в соответствии с потребностями программы.

Ответ __, __.

20. Выберите все операции, не применимые к переменным `pp` и `pnt` (укажите буквы):

Переменные

Операции

- 1) `char **pp;`
- 2) `char *pnt [N];`

- А. Инициализация указателя.
- В. Разыменование указателя(*).
- С. Получение адреса указателя(&).
- Д. Инкремент, декремент.
- Е. Сложение указателей.
- Ф. Вычитание указателей.
- Г. Сложение (вычитание) указателя и целого числа.
- Н. Использование операции `sizeof` (указатель) для получения размера памяти, необходимого для указателя.

Ответ: 1) __ 2) __, __, __, __, __.