

«

»

“

”

“ ”
 _____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ **Языки программирования и методы трансляции**

: 01.03.02

, :

: 3, : 5 6

		5	6
1	()	3	2
2		108	72
3	, .	45	43
4	, .	18	18
5	, .	0	0
6	, .	18	18
7	, .	9	9
8	, .	2	2
9	, .	7	5
10	, .	63	29
11	(, ,)		
12			

(): 01.03.02

228 12.03.2015 ., : 14.04.2015 .

: 1,

(): 01.03.02

, 4 20.06.2017
, 6 20.06.2017

, 6 21.06.2017

:

,

:

,
,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; <i>в части следующих результатов обучения:</i>	
4.	
5.	
Компетенция ФГОС: ПК.3 способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности; <i>в части следующих результатов обучения:</i>	
1.	
2.	
Компетенция ФГОС: ПК.4 способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности; <i>в части следующих результатов обучения:</i>	
1.	
Компетенция ФГОС: ПК.7 способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения; <i>в части следующих результатов обучения:</i>	
1.	
2.	

2.

2.1

	(
--	---	--

.3. 4	
1.знать роль транслятора в системе программного обеспечения	;
2.знать различия в подходах к проектированию трансляторов	;
3.уметь проектировать и разрабатывать транслятор	;
.3. 1	
4.знать особенности нисходящего и восходящего синтаксического анализа	;
5.знать основы автоматизации конструирования синтаксических и лексических анализаторов	;
.7. 2	
6.уметь разработать и реализовать блок генерации кода	;
.4. 1	

1.	segment, proc, assume, end, align,	0	3	11, 12	
2.		0	3	11, 12	
3.	label.	0	2	11, 12	
: ++					
4.	++. ++ / ++.	0	2	11, 13	
:					
5.		0	4	11, 12	
:					
6.	E I GOTO.	0	2	11, 12	
:					
7.		0	2	11, 12	
: 6					

: , ,				
8.	0	2	1, 2, 9	,
9.	1	4	7, 9	,
10.	1	2	1, 2, 4, 5	,
:				
11.	1	4	4	,

12.	<p> . ; . . , , </p>	1	4	4	<p> , , </p>
13.	<p> . . , , . . </p>	0	2	1, 10, 6	<p> , , </p>

3.2

	, .			
: 5				
:				
1.	3	7	11, 12, 15, 17, 18	<p> - , , : " " " </p>
: ++				

2.	++	3	5	11, 12, 13, 16, 17, 18	++ ; , ++, : " ++ "
:					
3.		3	6	11, 12, 16, 17, 18	() , : " "
: 6					
: , ,					
6.		1	4	3, 4, 8	; , ; , ; ; ;
:					

4.		1	4	10, 2, 3, 9	;	,	;	;	;	;
5.	()	1	4	10, 5, 7, 9	;	,	;	;	;	;
7.		2	6	10, 6, 9	;	,	;	;	;	;

4.

: 5				
1		11, 12, 13, 15, 16, 17, 18	47	0
[]: , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782.				
2		11, 12, 13	4	5
[]: , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782.				
3		11, 12, 13	12	2

<p>http://elibrary.nstu.ru/source?bib_id=vtls000234782. - [2017]. -</p>				
: 6				
1		1, 2, 3, 4, 5, 6, 7	5	1
<p>http://elibrary.nstu.ru/source?bib_id=vtls000183565. - [2013]. -</p>				
2		1, 2, 3, 4, 5, 6	20	3
<p>http://elibrary.nstu.ru/source?bib_id=vtls000183565. - [2013]. -</p>				
3		1, 2, 3, 4, 5, 6, 7, 8, 9	4	1
<p>http://elibrary.nstu.ru/source?bib_id=vtls000183565. - [2013]. -</p>				

5.

(. 5.1).

5.1

	-
	e-mail
	e-mail
	e-mail

5.2

1		.3; .3; .4; .7;
<p>Формируемые умения: з1. знать основные тенденции развития компьютерных технологий; з1. знать технологии модульного программирования; з2. знать основные технологии программирования; у1. уметь индивидуально и в команде вести разработку алгоритмических и программных решений в области системного и прикладного программирования; у2. уметь разрабатывать алгоритмы и реализовывать программное обеспечение для решения поставленных задач; у4. владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями</p>		
<p>Краткое описание применения: Беседа студента с преподавателем на предложенные преподавателем темы (контрольные вопросы).</p>		

$(\quad),$

-
15-

ECTS.

. 6.1.

6

6.1

: 5		
Лекция:	0	8
Лабораторная №1: Выполнение	8	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Лабораторная №1: Защита	6	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Лабораторная №2: Выполнение	8	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Лабораторная №2: Защита	6	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Лабораторная №3: Выполнение	8	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Лабораторная №3: Защита	6	12
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
Зачет:	8	20
() " , [2017]. - [] : - / . . . ; , [2017]. - : http://elibrary.nstu.ru/source?bib_id=vtls000234782 - "		
: 6		
Лабораторная №1:	8	16
() " : / - ; [.] - , 2011"		
Лабораторная №2:	8	16
() " : / - ; [.] - , 2011"		
Лабораторная №3:	8	16
() " : / - ; [.] - , 2011"		
Лабораторная №4:	8	16
() " : / - ; [.] - , 2011"		
РГЗ:	8	16
() " [] : - / ; , [2013]. - : http://elibrary.nstu.ru/source?bib_id=vtls000183565 - "		
Зачет:	10	20
-		

		/		
.3	4.	+	+	+
	5.	+		+
.3	1.			+
	2.	+		+
.4	1.	+		+
.7	1.	+		+
	2.	+	+	+

7.

1. Свердлов С. З. Языки программирования и методы трансляции : учебное пособие для вузов / С. З. Свердлов. - СПб., 2007. - 637 с. + 1 CD-ROM.. - Издательская программа 300 лучших учебников для высшей школы.
 2. Юров В. И. Assembler : учебное пособие для вузов по направлению подготовки специалистов " Информатика и вычислительная техника" / В. И. Юров. - СПб. [и др.], 2007. - 636 с. : ил., табл.
 3. Хабибуллин И. . Программирование на языке высокого уровня C/C++ : учебное пособие для вузов по направлению 654600 "Информатика и вычислительная техника" / И. Ш. Хабибуллин. - СПб, 2006. - 485 с. : ил.
 4. Архитектура ЭВМ и вычислительных систем: Учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - 5-е изд., перераб. и доп. - М.: Форум: НИЦ ИНФРА-М, 2015 - 512 с.: ил.; 60x90 1/16. - (Профессиональное образование). (п) ISBN 978-5-91134-742-0 - Режим доступа: <http://znanium.com/catalog.php?bookinfo=492687> - Загл. с экрана.
 5. Карпов Ю. Г. Теория и технология программирования. Основы построения трансляторов : учебное пособие для вузов по направлениям подготовки бакалавров и магистров 553000 - "Системный анализ и управление" и 552800 - "Информатика и вычислительная техника" / Ю. Г. Карпов. - СПб., 2005. - 270 с. : ил., табл.
-
1. Еланцева И. Л. Языки программирования и методы трансляции. Раздел "Методы трансляции" : конспект лекций / И. Л. Еланцева, И. А. Полетаева ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 97, [2] с. : ил., табл.. - Режим доступа: <http://www.library.nstu.ru/fulltext/metodics/2008/elan.rar>
 2. Юров В. И. Assembler : учебное пособие для вузов по направлению подготовки специалистов " Информатика и вычислительная техника" / В. И. Юров. - СПб. [и др.], 2006. - 636 с. : ил., табл.

3. Лебедев В. Н. Введение в системы программирования. - М., 1975. - 311 с. : ил.
4. Ахо А. В. Компиляторы : Принципы, технологии, инструменты / А. Ахо, Р. Сети, Д. Ульман. - М., 2001. - 766 с.
5. Хантер Р. Проектирование и конструирование компиляторов / Р. Хантер; пер. с англ. С. М. Круговой под ред. В. М. Савинкова. - М., 1984. - 231, [1] с. : ил.
6. Абель П. Язык Ассемблера для IBM PC и программирования : Пер. с англ.. - М., 1992. - 447 с.
7. Пратт М. Языки программирования: разработка и реализация : Пер с англ. / Под ред. Баяковского Ю. М. - М., 1979. - 574 с.

-

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>
5. :

8.

8.1

1. Лисицин Д. В. Программирование на языке Ассемблера [Электронный ресурс] : учебно-методическое пособие / Д. В. Лисицин, Р. В. Петров, И. А. Полетаева ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://ciu.nstu.ru/fulltext/unofficial/2011/lib_842_1320653325.doc. - Загл. с экрана.
2. Программирование на языке Ассемблера : методические указания к лабораторным работам по курсам "Языки программирования и методы трансляции" (для 3 курса ФПМИ по направлению 010400 - Прикладная математика и информатика) и "Теория вычислительных процессов и структур" (для 3 курса ФПМИ по направлению 010500 - Математическое обеспечение и администрирование информационных систем) / Новосиб. гос. техн. ун-т ; [сост.: Д. В. Лисицин, Р. В. Петров, И. А. Полетаева]. - Новосибирск, 2011. - 34, [2] с. : табл.
3. Лисицин Д. В. Низкоуровневое программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2017]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000234782. - Загл. с экрана.
4. Языки программирования и методы трансляции : методические указания / Новосиб. гос. техн. ун-т ; [сост.: И. Л. Еланцева, И. А. Полетаева]. - Новосибирск, 2011
5. Еланцева И. Л. Языки программирования и методы трансляции [Электронный ресурс] : электронный учебно-методический комплекс / И. Л. Еланцева, М. Э. Рояк, Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000183565. - Загл. с экрана.

8.2

1 Microsoft Visual C++

9. -

1	(- , ,)	

1	(Internet)	

«Новосибирский государственный технический университет»

Кафедра прикладной математики
Кафедра теоретической и прикладной информатики

“УТВЕРЖДАЮ”
ДЕКАН ФПМИ
д.т.н., доцент В.С. Тимофеев
“ ” г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Языки программирования и методы трансляции

Образовательная программа: 01.03.02 Прикладная математика и информатика, профиль: Компьютерное моделирование и информационные технологии

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Языки программирования и методы трансляции приведена в Таблице.

Таблица

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.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	у4. владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями	Генерация кода. Представление промежуточной программы. Понятие постфиксной формы записи, её связь с деревом грамматического разбора, задача преобразования последовательности лексем в постфиксную запись. Модели процесса генерации кода. Системы построения трансляторов. Интерфейс с языком C++ Основные элементы языка ассемблера Программирование сопроцессора Проектирование и реализация таблиц, используемых в трансляторе Разработка и реализация блока синтаксического анализа Синтаксический анализ. Понятие дерева грамматического разбора. Нисходящие и восходящие группы методов синтаксического анализа, их основные идеи. Понятие конечного автомата со стековой памятью в качестве основы синтаксического анализатора Структуры трансляторов и методы их реализации. Цели и задачи трансляции. Основные этапы трансляции	5 семестр: Отчет по лабораторной работе № 1, контрольные вопросы 1–16, Отчет по лабораторной работе № 2, контрольные вопросы 1–11, Отчет по лабораторной работе № 3, контрольные вопросы 1–24 6 семестр: Отчет по лабораторной работе №3, Контрольные вопросы 1-8, Отчет по лабораторной работе №4, Контрольные вопросы 1-7	5 семестр: Зачет, вопросы 1, 3-59. 6 семестр: Зачет, задания теста 13-20
ОПК.3	у5. уметь использовать интегрированные среды разработки программ	Интерфейс с языком C++ Основные элементы языка ассемблера Программирование сопроцессора	5 семестр: Отчет по лабораторной работе № 1, контрольные вопросы 1–16, Отчет по лабораторной работе № 2, контрольные вопросы 1–11, Отчет по лабораторной работе № 3, контрольные вопросы 1–24	5 семестр: Зачет, вопросы 2, 18-24, 60

<p>ПК.3/НИ способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности</p>	<p>31. знать основные тенденции развития компьютерных технологий</p>	<p>Основная идея алгоритма нисходящего синтаксического анализа. Критерии выбора грамматики для реализации этого алгоритма. Понятие S-грамматики для данного языка. Понятие и определение LL(1)-грамматики. Алгоритм определения принадлежности грамматики к классу LL(1). Задача преобразования LL(1)-грамматики в управляющую таблицу нисходящего синтаксического анализатора. Обработка ошибок при нисходящем синтаксическом анализе, методы нейтрализации ошибок. Основные идеи восходящих методов синтаксического анализа. Операции сдвига и свёртки, состояние стека автомата до и после их выполнения. Задача определения множества состояний восходящего анализатора. Понятие конфигурации, связь между подмножествами конфигураций и состоянием автомата. Алгоритм построения подмножеств эквивалентных конфигураций. Разметка грамматики. Структура управляющей таблицы восходящего анализатора. Обработка ошибок при восходящем синтаксическом анализе, методы нейтрализации ошибок Синтаксический анализ. Понятие дерева грамматического разбора. Нисходящие и восходящие группы методов синтаксического анализа, их основные идеи. Понятие конечного автомата со стековой памятью в качестве основы синтаксического анализатора Язык ассемблера, задачи, решаемые с его помощью. Регистры процессора. Процесс разработки программы, структура программы. Понятия команд, директив и операций. Формат записи команд. Директивы segment, proc, assume, end, align, упрощенные директивы сегментации, директивы определения данных. Особенности разработки 16-разрядных программ под MS-DOS.</p>	<p>6 семестр: Отчет по лабораторной работе №3, Контрольные вопросы 1-8, Отчет по лабораторной работе №4, Контрольные вопросы 1-7 РГЗ (темы 1, 6-12, 17-21)</p>	<p>5 семестр: Зачет, вопросы 1-6</p> <p>6 семестр: Зачет, задания теста 13-20</p>
---	--	---	--	---

ПК.3/НИ	32. знать основные технологии программирования	Директивы определения идентификаторов. Операции. Директива label. Команды языка ассемблера. Команды пересылки данных. Арифметические команды. Режимы адресации. Команды передачи управления и сравнения. Команды манипулирования битами. Команды для работы с двоично-десятичными числами. Основные элементы языка ассемблера Условные директивы. Макросы. Блоки повторения. Директивы EXITM и GOTO. Макрофункции. Макрооперации. Форматы данных сопроцессора. Архитектура сопроцессора. Особенности программирования сопроцессора. Команды сопроцессора. Исключения сопроцессора. Цепочечные команды. Структуры. Объединения. Работа с динамической памятью. Язык ассемблера, задачи, решаемые с его помощью. Регистры процессора. Процесс разработки программы, структура программы. Понятия команд, директив и операций. Формат записи команд. Директивы segment, proc, assume, end, align, упрощенные директивы сегментации, директивы определения данных. Особенности разработки 16-разрядных программ под MS-DOS.	5 семестр: Отчет по лабораторной работе № 1, контрольные вопросы 1–16, Отчет по лабораторной работе № 3, контрольные вопросы 1–24	5 семестр: Зачет, вопросы 1, 3-59.
ПК.4/ППр способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности	у1. уметь индивидуально и в команде вести разработку алгоритмических и программных решений в области системного и прикладного программирования	Основные элементы языка ассемблера	5 семестр: Отчет по лабораторной работе № 1, контрольные вопросы 1–16	5 семестр: Зачет, вопросы 3-17, 25-30
ПК.7/ППр способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	31. знать технологии модульного программирования	Интерфейс с языком C++ Использование функций на языке ассемблера в программах на языке C++. Вызов функций на языке C++ из программ на языке ассемблера. Использование локальных данных. Использование библиотечных функций языка Си в программах/подпрограммах на языке ассемблера. Использование вставок на	5 семестр: Отчет по лабораторной работе № 2, контрольные вопросы 1–11 6 семестр: Отчет по лабораторной работе №1, Контрольные вопросы 1-2,	5 семестр: Зачет, вопросы 18-24 6 семестр: Зачет, задания теста 1-12

		языке ассемблера в программе на языке C++. Понятие формального языка и грамматики, конечного автомата. Задачи лексического анализатора. Диагностика лексических ошибок. Структуры трансляторов и методы их реализации. Цели и задачи трансляции. Основные этапы трансляции	Отчет по лабораторной работе №2, Контрольные вопросы 1-7 РГЗ (темы 5, 13-16)	
ПК.7/ППр	у2. уметь разрабатывать алгоритмы и реализовывать программное обеспечение для решения поставленных задач	Генерация кода. Представление промежуточной программы. Понятие постфиксной формы записи, её связь с деревом грамматического разбора, задача преобразования последовательности лексем в постфиксную запись. Модели процесса генерации кода. Системы построения трансляторов. Интерфейс с языком C++ Программирование сопроцессора Разработка и реализация блока генерации кода	5 семестр: Отчет по лабораторной работе № 2, контрольные вопросы 1–11, Отчет по лабораторной работе № 3, контрольные вопросы 1–24 6 семестр: Отчет по лабораторной работе №3, Контрольные вопросы 1-8, Отчет по лабораторной работе №4, Контрольные вопросы 1-7 РГЗ (темы 2-4, 22-23)	5 семестр: Зачет, вопросы 18-24, 31-43. 6 семестр: Зачет, задания теста 13-20

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 5 семестре - в форме дифференцированного зачета в 6 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ПК.3/НИ, ПК.4/ППр, ПК.7/ППр.

Зачет в 5 семестре проводится в письменной форме, по комплектам заданий, комплекты заданий составляются из вопросов, приведенных в паспорте зачета, позволяющих оценить показатели сформированности соответствующих компетенций

На подготовку к зачету в 5 семестре отводится 14 часов.

На подготовку к зачету в 6 семестре отводится 5 часов.

Зачет в 6 семестре проводится в форме письменного тестирования, пример заданий теста можно посмотреть в паспорте зачета.

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

В 6 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (РГЗ). Требования к выполнению РГЗ, состав и правила оценки сформулированы в паспорте РГЗ.

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

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ПК.3/НИ, ПК.4/ППр, ПК.7/ППр, 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 прикладной математики
Кафедра теоретической и прикладной информатики

Паспорт зачета

по дисциплине «Языки программирования и методы трансляции», 5 семестр

1. Методика оценки

Зачет проводится в письменной форме, по билетам. Билет состоит из двух вопросов и формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-30, второй вопрос из диапазона вопросов 31-60 (список вопросов приведен ниже).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет ФПМИ

Билет № 1

к зачету по дисциплине «Языки программирования и методы трансляции»

1. Директива PROC, команды CALL, RET, JMP
2. Команды передачи данных: FLD, FST(P), FXCH, FILD, FIST(P), FBLD, FBSTP

Утверждаю: зав. кафедрой ____ должность, ФИО
(подпись)
(дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент ответил на вопросы слишком неполно, показал слишком неполное владение теоретическими и практическими аспектами, относящихся к вопросам элементов языка ассемблера, оценка составляет 0-7 баллов
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент лишь частично ответил на вопросы, показал некоторое владение теоретическими и практическими аспектами, относящихся к вопросам элементов языка ассемблера, оценка составляет 8-10 баллов
- Ответ на билет для зачета засчитывается на **базовом** уровне, если студент, отвечая на вопросы, не дал развернутых ответов, показал владение теоретическими и практическими аспектами, относящихся к вопросам элементов языка ассемблера, оценка составляет 11-16 баллов

- Ответ на билет для зачета засчитывается на **продвинутом** уровне, если студент дал развернутые ответы на вопросы, показал свободное владение теоретическими и практическими аспектами, относящихся к вопросам элементов языка ассемблера, оценка составляет 17-20 баллов

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем вопросам билета составляет не менее 8 баллов (из 20 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к зачету по дисциплине «Языки программирования и методы трансляции»

1. Особенности языка ассемблера и задачи, решаемые с его использованием
2. Разработка программы на языке ассемблера в среде Visual C++
3. Регистры микропроцессора
4. Директивы SEGMENT, ASSUME, END, упрощенные директивы сегментации
5. Директивы определения данных, директива ALIGN
6. Команды пересылки данных
7. Команды ADD, ADC, XADD, INC
8. Команды SUB, SBB, DEC, NEG
9. Команды умножения и деления
10. Команды преобразования типа
11. Регистровая, непосредственная и прямая адресация
12. Косвенная регистровая адресация и адресация по базе
13. Прямая адресация с индексированием, адресация по базе с индексированием
14. Директива PROC, команды CALL, RET, JMP
15. Команды условной передачи управления, команда CMP
16. Команды SETx, CMOVx
17. Команды управления циклами
18. Основы взаимодействия языков C++ и ассемблера. Передача управления в подпрограмму и обратно
19. Передача данных из программы на языке C++ в подпрограмму на языке ассемблера: Использование глобальных данных
20. Передача данных из программы на языке C++ в подпрограмму на языке ассемблера: Использование аргументов функции
21. Возвращение данных из подпрограммы на языке ассемблера в программу на языке C++
22. Вызов функций на языке C++ из программ на языке ассемблера, использование локальных данных
23. Использование библиотечных функций языка Си в программах или подпрограммах на языке ассемблера
24. Использование вставок на языке ассемблера в программе на языке C++
25. Команды AND, OR, XOR, NOT, TEST
26. Команды сканирования битов, команды проверки и модификации битов
27. Команды сдвига
28. Команды циклического сдвига, команды сдвига двойной точности
29. Форматы хранения двоично-десятичных чисел
30. Коррекция результатов арифметических команд для двоично-десятичных чисел
31. Форматы и способы задания двоичных, двоично-десятичных и вещественных чисел в сопроцессоре
32. Специальные численные значения сопроцессора

33. Регистровый стек сопроцессора, регистр тегов TWR сопроцессора
34. Регистры состояния (SWR) и управления (CWR) сопроцессора
35. Особенности вычисления выражений в сопроцессоре с использованием постфиксной записи
36. Команды передачи данных: FLD, FST(P), FXCH, FILD, FIST(P), FBLD, FBSTP
37. Команды сравнения данных: FCOM, FCOMP(P), FICOM(P), FTST, FCOMI(P)
38. Арифметические команды: целочисленные (FIADD, FISUB(R), FIMUL, FIDIV(R)) и вещественные (Fx(P), FxR(P))
39. Дополнительные арифметические команды: FSQRT, FABS, FCHS, FXTRACT, FSCALE, FRNDINT, FPREM, FPREM1
40. Команды тригонометрических функций: FCOS, FSIN, FSINCOS, FPTAN, FPATAN
41. Команды вычисления логарифмов и степеней: F2XM1, FYL2X, FYL2XP1
42. Команды управления сопроцессором: FINIT, FINCSTP, FDECSTP, FFREE
43. Исключения сопроцессора и их обработка
44. Директивы определения идентификаторов
45. Операции: арифметические, логические, отношения
46. Операции, возвращающие значения, операции присваивания атрибутов, директива LABEL
47. Условные директивы
48. Сравнение макросов и процедур, определение и вызов макросов
49. Директивы LOCAL, IFB, IFNB
50. Блоки повторения
51. Директивы EXITM и GOTO, макрофункции
52. Макрооперации
53. Общая характеристика цепочечных команд, префиксы повторения
54. Команды пересылки и сравнения цепочек
55. Команды сканирования, загрузки и сохранения цепочек
56. Определение структуры, объявление структурных переменных
57. Обращение к структурным переменным, вложенные структуры
58. Объединения
59. Работа с динамической памятью
60. Средства операционной системы Windows для ввода-вывода информации в консольных приложениях

Паспорт зачета

по дисциплине «Языки программирования и методы трансляции», 6 семестр

1. Методика оценки

Зачет проводится в письменной форме, по тестам. Вариант теста состоит из 20 заданий по темам лекционных и лабораторных занятий. Для правильного ответа на задания теста студент должен иметь знания по следующим вопросам:

- 1) Основные этапы трансляции.
- 2) Организация работы лексического анализатора.
- 3) Формальные языки и грамматики.
- 4) Конечные автоматы.
- 5) Задачи синтаксического анализатора.
- 6) Нисходящий синтаксический разбор. Понятие и свойства LL(1)- грамматики.
- 7) Восходящий синтаксический разбор. Понятие и свойства LR(1)- грамматики.
- 7) Способы записи промежуточного кода.
- 8) Способы построения и использования постфиксной записи.

Пример теста для зачета

1. Выберите неправильный ответ.

К единицам языка относится

- а) Идентификатор;
- б) Зарезервированное слово;
- в) Токен;
- г) Разделитель.

2. Вставьте два пропущенных слова.

Лексема – это _____ и ее атрибуты.

3. Выберите правильный ответ.

Токен – это

- а) единица языка;
- б) тип лексемы и ее адрес в соответствующей таблице;
- в) элемент транслируемой программы.

4. Выберите правильный ответ.

Атрибутами обладают

- а) Токены;
- б) Идентификаторы и константы;
- в) Знаки операций;
- г) Постоянные таблицы.

5. Классифицируйте таблицы, используемые в трансляторах, заполнив форму

Постоянные таблицы транслятора	Переменные таблицы транслятора

- а) Таблица констант;
- б) Таблица разделителей;
- в) Таблица идентификаторов;
- г) Таблица знаков операций;
- д) Таблица зарезервированных слов.

6. Установите соответствие между видом таблицы (левый столбец) и типами операций над ее элементами (правый столбец)

- | | |
|-----------------------|-----------------------|
| 1) Постоянные таблицы | а) Поиск и исключение |
| 2) Переменные таблицы | б) Поиск и добавление |
| | в) Поиск |
| | г) Добавление |

7. Установите правильную последовательность этапов трансляции

- а) Генерация кода.
- б) Синтаксический анализ.
- в) Лексический анализ.
- г) Оптимизация кода.
- д) Генерация промежуточного кода.

8. Выберите правильный ответ.

Грамматика

- а) формирует токен;
- б) задает правила формирования цепочек языка;
- в) задает правила распознавания принадлежности цепочки данному языку;
- г) задает правила работы сканера.

9. Выберите правильный ответ.

Сканер строится на основе

- а) детерминированного конечного автомата;
- б) недетерминированного конечного автомата;
- в) автомата с магазинной памятью.

10. Выберите все неправильные ответы.

К результатам работы сканера относятся

- а) файл сообщений об обнаруженных лексических ошибках;
- б) файл сообщений об обнаруженных синтаксических ошибках;
- в) файл токенов;
- г) диаграмма состояний автомата.

11. Классифицируйте элементы, составляющие автоматную грамматику и конечный автомат, заполнив форму

Автоматная грамматика	Конечный автомат

- а) Множество терминальных символов;
- б) Начальное состояние;
- в) Множество входного алфавита;

- г) Множество конечных состояний;
- д) Множество правил вывода;
- е) Множество нетерминальных символов.

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

Сканер обнаруживает следующие типы ошибок

- а) неверная конструкция языка программирования да/нет
- б) недопустимый символ да/нет
- в) ошибка в последовательности литер, определяющих константу да/нет
- г) деление на ноль да/нет
- д) ошибка при обращении к оперативной памяти да/нет

13. Выберите из представленных грамматик LL(1)-грамматику

$G[A] = \{N = \{A, B, C\}, T = \{b, c, d, z\}\}$

а) $A \rightarrow Bd \mid dC$ $A \rightarrow z$ $B \rightarrow \varepsilon \mid bc$ $C \rightarrow z$	б) $A \rightarrow zA \mid \varepsilon \mid BC$ $B \rightarrow \varepsilon \mid bB$ $C \rightarrow cd \mid \varepsilon$	в) $A \rightarrow zA \mid BC$ $B \rightarrow \varepsilon \mid bB$ $C \rightarrow cC \mid d$
--	--	---

14. Ответьте, используя код:

A	B	C	D
а, б, в	б, в, г	а, в, д	в, г, д

При LL(1)-разборе используются

- а) стек символов;
- б) стек состояний;
- в) файл токенов;
- г) таблица разбора;
- д) очередь состояний.

15. Вставьте пропущенное слово

В таблице LL(1)-разбора в столбце *accept* значение *true* ставится только для состояний, соответствующих _____ символам.

16. Выберите все правильные ответы

Необходимым условием успешного окончания LL(1)-разбора цепочки является условие

- а) дерево разбора двоичное.
- б) входным символом является начальный символ грамматики.
- в) файл токенов прочитан до конца.
- г) стек состояний пуст.
- д) все правила грамматики использованы.

17. Выберите правильный ответ

При LR(1)-разборе в стеке символов сохраняются

- а) только терминальные символы.
- б) только нетерминальные символы.
- в) терминальные и нетерминальные символы.

18. Выберите все правильные ответы.

Генератор кода создает команды выделения памяти для

- а) элементов таблицы идентификаторов.
- б) элементов таблицы ключевых слов.
- в) элементов таблицы LR(1)-разбора.
- г) элементов таблицы LL(1)-разбора.

19. Выберите правильный ответ.

Постфиксной записью, соответствующей выражению $x=(y+z)*(u+w)/v*t$ является

- а) $x\ y\ z\ u\ +\ w\ *\ +\ v\ t\ /\ * =$
- б) $x\ y\ z\ +\ u\ w\ +\ *\ v\ /\ t\ *\ =$
- в) $x\ y\ z\ +\ u\ *\ w\ +\ v\ /\ *\ t\ =$

20. Укажите все правильные ответы, используя код:

A	B	C	D
а, б, в	б, в, г	а, в, д	б, г, д

В состав «четверки» входят следующие элементы

- а) операнды
- б) приоритет операции
- в) операция
- г) количество операндов
- д) результат

2. Критерии оценки

За полный и правильный ответ за каждое задание студент получает один балл.

- Зачет считается сдан на **неудовлетворительном** уровне, если студент дал правильные ответы менее чем на 10 вопросов, оценка составляет от 0 до 9 баллов
- Зачет считается сдан на **пороговом** уровне, если студент дал полный и правильный ответ не менее чем на 10 вопросов, оценка составляет от 10 до 14 баллов
- Зачет считается сдан на **базовом** уровне, если студент дал полный и правильный ответ не менее чем на 15 вопросов, оценка составляет от 15 до 18 баллов
- Зачет считается сдан на **продвинутом** уровне, если студент дал полный и правильный ответ не менее чем на 19 вопросов, оценка составляет от 19 до 20 баллов

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям составляет не менее 10 баллов (из 20 возможных).

Паспорт лабораторных работ

по дисциплине «Языки программирования и методы трансляции», 5 семестр

1. Методика оценки

ЛАБОРАТОРНАЯ РАБОТА №1

Задание 1. Изучить основные команды языка Ассемблера, функции операционной системы, осуществляющие ввод-вывод, работу символьными данными.

Задание 2. По предложенному преподавателем варианту разработать программу на языке Ассемблера, решающую поставленную задачу:

- 1) ввод с клавиатуры 2-х чисел в заданной системе счисления;
- 2) выполнение арифметической операции над этими числами (в предположении, что размер чисел не вызывает переполнения регистров);
- 3) вывод результата в заданной системе счисления.

Все промежуточные данные должны сохраняться в памяти. При выводе результата не использовать функцию `wsprintfA`.

Задание 3. Отладить программу, убедиться в правильности ее работы на тестовых примерах.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, алгоритм, текст разработанной программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Функция `GetStdHandle`: назначение, параметры, примеры использования.
2. Функция `WriteConsoleA`: назначение, параметры, примеры использования.
3. Функция `CharToOem`: назначение, параметры, примеры использования.
4. Функция `ReadConsoleA`: назначение, параметры, примеры использования.
5. Функция `wsprintfA`: назначение, параметры, примеры использования.
6. Команда `mov`: назначение, операнды, примеры использования.
7. Стек и команды работы со стеком `push`, `pop`: назначение, операнды, примеры использования.
8. Команды сохранения и извлечения флагов и регистров из стека.
9. Команды сложения.
10. Команды вычитания.
11. Команды деления.
12. Команды умножения.
13. Директивы определения данных.
14. Регистровая, непосредственная и прямая адресация.
15. Косвенная регистровая адресация и адресация по базе.
16. Прямая адресация с индексированием и адресация по базе с индексированием.

ЛАБОРАТОРНАЯ РАБОТА № 2

Задание 1. Изучить способы вызова процедуры на языке Ассемблера, возврата из процедуры.

Задание 2. Изучить условия взаимодействия функции на языке C++ с процедурой на языке Ассемблера. Изучить правила передачи управления в процедуру и обратно.

Изучить способы обмена данными между вызывающей функцией на языке C++ и процедурой на языке Ассемблера.

Задание 3. Написать процедуру на языке Ассемблера, реализующую функцию заданного варианта. Написать вызывающую функцию на языке C++, осуществляющую ввод исходных данных и вывод результатов.

Задание 4. Отладить программу, убедиться в правильности ее работы на тестовых примерах.

Задание 5. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, алгоритм, текст разработанной программы и результаты тестирования.

Задание 6. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Определение процедуры.
2. Команда вызова процедуры CALL.
3. Команда возврата из процедуры.
4. Косвенный вызов процедуры.
5. Основы взаимодействия языков C++ и Ассемблера.
6. Использование глобальных данных для передачи данных процедуре на языке Ассемблера.
7. Использование аргументов функции для передачи данных процедуре на языке Ассемблера.
8. Возвращение значений из процедуры на языке Ассемблера.
9. Вызов функций на языке C++ из процедур на языке Ассемблера.
10. Использование локальных данных.
11. Использование вставок на языке Ассемблера в программе на языке C++.

ЛАБОРАТОРНАЯ РАБОТА № 3

Задание 1. Изучить архитектуру и средства программирования сопроцессора на языке ассемблера. Изучить численный метод решения нелинейного уравнения, заданный в варианте.

Задание 2. Написать программу, реализующую метод заданного варианта.

2.1. Программа должна состоять из модулей на C++ и ассемблере, причем в модуле на C++ осуществляется ввод-вывод, а все вычисления – в модуле на ассемблере.

2.2. Следует оформить отдельными процедурами на ассемблере:

- вычисление функции, задающей левую часть уравнения;
- вычисление (если требуется) производных функции;
- вычисления, связанные с методом решения.

2.3. Входными данными является точность решения. Выходными данными являются решение и число итераций.

2.4. Для функции и (если требуется) ее производных необходимо использовать алгоритм вычисления выражений в постфиксной записи (любые отклонения от него должны быть обоснованы повышением эффективности). Необходимо обращать особое внимание на грамотное использование стека сопроцессора.

Задание 3. Отладить программу, убедиться в правильности ее работы на тестовых примерах.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, график функции с указанием начального приближения и искомого корня, алгоритм, текст разработанной программы (на этапах вычислений в комментариях указывать состояние стека сопроцессора) и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Форматы и способы задания целых и двоично-десятичных чисел.
2. Форматы и способы задания вещественных чисел.
3. Денормализованные вещественные числа, нуль, бесконечность.

4. Нечисла (SNAN, QNAN).
5. Регистровый стек: принципы работы.
6. Регистр состояния SWR: назначение, основные поля.
7. Регистр управления CWR: назначение, основные поля.
8. Регистр тегов TWR: назначение, принцип использования.
9. Особенности вычисления выражений с использованием постфиксной записи.
10. Инициализация сопроцессора: команда FINIT.
11. Команды передачи данных в вещественном формате: FLD, FST(P), команда обмена FXCH.
12. Команды передачи данных в целочисленном и двоично-десятичном формате: FILD, FIST(P), FBLD, FBSTP.
13. Команды сравнения данных: FCOM, FCOMP(P), FICOM(P), FTST, FCOMI(P).
14. Целочисленные арифметические команды: FIADD, FISUB(R), FIMUL, FIDIV(R).
15. Вещественные арифметические команды: Fx(P), FxR(P).
16. Дополнительные арифметические команды: FSQRT, FABS, FCHS, FXTRACT, FSCALE, FRNDINT.
17. Дополнительные арифметические команды: FPREM, FPREM1
18. Команды тригонометрических функций: FCOS, FSIN, FSINCOS, FPTAN, FPATAN.
19. Команды вычисления логарифмов и степеней: F2XM1, FYL2X, FYL2XP1.
20. Команды управления сопроцессором: FINCSTP, FDECSTP, FFREE.
21. Исключения сопроцессора и их обработка: недействительная операция.
22. Исключения сопроцессора и их обработка: деление на ноль, денормализация операнда.
23. Исключения сопроцессора и их обработка: переполнение и антипереполнение, неточный результат.
24. Исключения сопроцессора и их обработка: приоритет особых случаев.

2. Критерии оценки

- Лабораторная работа считается **не выполненной**, если студент выполнил задания частично, отчет оформил с недостатками, показал недостаточное владение теоретическими и практическими навыками, оценка составляет 0-13 баллов
- Лабораторная работа считается выполненной **на пороговом** уровне, если студент выполнил задания не полностью, отчет оформил с небольшими недостатками, показал некоторое владение теоретическими и практическими навыками, касающимися изучаемых элементов языка ассемблера, оценка составляет 14-18 баллов
- Лабораторная работа считается выполненной **на базовом** уровне, если студент выполнил задания полностью, но показал неполное владение теоретическими и практическими навыками, касающимися изучаемых элементов языка ассемблера, отчет оформил без недостатков, оценка составляет 19-21 баллов
- Лабораторная работа считается выполненной **на продвинутом** уровне, если студент выполнил задания полностью, отчет оформил без недостатков, показал свободное владение теоретическими и практическими навыками, касающимися изучаемых элементов языка ассемблера, оценка составляет 22-24 баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за лабораторные работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Перечни вариантов

ЛАБОРАТОРНАЯ РАБОТА №1

1. Сложение чисел, ввод в восьмеричной системе счисления (не менее 3-х знаков каждое число), вывод в десятичной системе счисления.
2. Сложение чисел, ввод в шестнадцатеричной системе счисления (не менее 3-х знаков каждое число), вывод в десятичной системе счисления.
3. Сложение чисел, ввод в десятичной системе счисления (не менее 4-х знаков каждое число), вывод в шестнадцатеричной системе счисления.
4. Вычитание чисел, ввод в восьмеричной системе счисления (не менее 3-х знаков каждое число), вывод в шестнадцатеричной системе счисления.
5. Вычитание чисел, ввод в шестнадцатеричной системе счисления (не менее 3-х знаков каждое число), вывод в десятичной системе счисления.
6. Вычитание чисел, ввод в десятичной системе счисления (не менее 3-х знаков каждое число), вывод в шестнадцатеричной системе счисления.
7. Умножение чисел, ввод в восьмеричной системе счисления (не менее 3-х знаков каждое число), вывод в десятичной системе счисления.
8. Умножение чисел, ввод в шестнадцатеричной системе счисления (не менее 3-х знаков каждое число), вывод в десятичной системе счисления.
9. Умножение чисел, ввод в десятичной системе счисления (не менее 4-х знаков каждое число), вывод в шестнадцатеричной системе счисления.
10. Сложение чисел, ввод в двоичной системе счисления (не менее 16-ти знаков каждое число), вывод в десятичной системе счисления.
11. Умножение чисел, ввод в двоичной системе счисления (не менее 16-ти знаков каждое число), вывод в десятичной системе счисления.
12. Вычитание чисел, ввод в двоичной системе счисления (не менее 16-ти знаков каждое число), вывод в шестнадцатеричной системе счисления.

ЛАБОРАТОРНАЯ РАБОТА № 2

1. Написать программу формирования сжатой строки символов. Сжатие заключается в удалении точек из исходной строки при просмотре ее слева направо.
2. Написать программу формирования сжатой строки символов. Сжатие заключается в удалении точек из исходной строки при просмотре ее справа налево.
3. Написать программу выделения из исходной строки подстроки символов заданной длины с указанного номера позиции.
4. Написать программу, определяющую номер позиции, с которой начинается первое слева вхождение заданной конфигурации символов в исходную строку.
5. Написать программу формирования строки из исходной путем заданного числа повторений исходной строки.
6. Написать программу, выполняющую следующую функцию. Заданы две строки. Проверить вхождение каждого символа первой строки во вторую строку. Если какой-либо (первый слева) символ первой строки не представлен во второй строке, то фиксируется номер позиции этого символа в первой строке.
7. Написать программу, которая по исходной строке формирует инвертированную строку и проверяет, совпадают ли обе строки.
8. Написать программу, находящую максимальный и минимальный символы в исходной строке.
9. Написать программу, заменяющую в исходной строке все десятичные цифры на заданный символ.
10. Написать программу, удаляющую из исходной строки повторные вхождения заданного символа.
11. Написать программу, удаляющую точки в конце исходной строки.
12. Написать программу, удаляющую из исходной строки заданную конфигурацию символов.

ЛАБОРАТОРНАЯ РАБОТА № 3

1. Метод половинного деления
2. Метод хорд
3. Метод Ньютона
4. Метод секущих
5. Метод Стеффенсена
6. Метод Хейли
7. Метод Ньютона–половинного деления
8. Метод Ньютона с подвижным полюсом
9. Однопараметрический полюсный метод Ньютона
10. Метод простых итераций
11. Δ^2 -процесс Эйткена
12. Метод Вегстейна

Паспорт расчетно-графического задания

по дисциплине «Языки программирования и методы трансляции», 6 семестр

1. Методика оценки

В рамках расчетно-графического задания по дисциплине студенты должны самостоятельно изучить этап или метод трансляции в соответствии с темой РГЗ.

При выполнении расчетно-графического задания студенты должны:

- используя конспект лекций и рекомендуемую научную литературу, изучить материал по заданной теме;
- написать реферат;
- реализовать исследованные методы в виде подпрограмм, включив их в спроектированный транслятор;
- защитить работу.

Обязательные структурные части РГЗ: Реферат о современном состоянии вопроса, описание реализованных алгоритмов, текст программы на языке C++, тестовые примеры.

2. Критерии оценки

- Работа считается **не выполненной**, если реферат не соответствует современному состоянию вопроса, студент не ориентируется в содержании реферата, оценка составляет менее 8 баллов.
- Работа считается выполненной **на пороговом** уровне, если представлен только реферат, без программной реализации алгоритмов, студент хорошо разбирается в содержании реферата, оценка составляет 8-9 баллов.
- Работа считается выполненной **на базовом** уровне, если представлен содержательный реферат, имеется описание реализованных программно методов и проведенных тестов, студент хорошо разбирается в содержании реферата, оценка составляет 10-14 баллов.
- Работа считается выполненной **на продвинутом** уровне, если РГЗ выполнена и защищена в срок, представлен содержательный реферат, имеется описание и обоснование реализованных программно методов и проведенных тестов, студент свободно разбирается в теме РГЗ, оценка составляет 15-16 баллов.

3. Шкала оценки

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

4. Перечень тем РГЗ

1. Прямые методы трансляции. Особенности реализации конструкций языков программирования с использованием прямых методов.
2. Методы оптимизации кода. Оптимизация вычислений с константами.
3. Методы оптимизации кода. Оптимизация выражений.
4. Методы оптимизации кода. Оптимизация циклов.
5. Методы диагностики и исправления ошибок. Лексические ошибки.
6. Методы диагностики и исправления ошибок. Синтаксические ошибки.

7. Методы диагностики и исправления ошибок. Ошибки в употреблении скобок.
8. Методы диагностики и исправления ошибок. Контекстно-зависимые ошибки.
9. Методы диагностики и исправления ошибок. Ошибки, связанные с употреблением различных типов данных.
10. Методы диагностики и исправления ошибок. ошибки выполнения: нахождение индекса массива вне области действия, целочисленное переполнение, попытка чтения за пределами файла.
11. Методы диагностики и исправления ошибок: ошибки, связанные с нарушением ограничений на размер программ, число элементов в таблице символов, размер стека разбора и пр.
12. Семантические действия.
13. Создание надежных компиляторов. Использование формального определения.
14. Создание надежных компиляторов. Модульное проектирование.
15. Создание надежных компиляторов. Проверка компилятора.
16. Анализатор, проверяющий принадлежность грамматики к LL(1)-типу.
17. Программа для построения таблиц синтаксического LL-разбора.
18. Программа для построения таблиц синтаксического LR-разбора.
19. Синтаксический анализатор на основе грамматики простого предшествования.
20. Синтаксический анализатор на основе грамматики расширенного предшествования.
21. Синтаксический анализатор на основе грамматики операторного предшествования.
22. Схемы управления памятью: статическое управление памятью.
23. Схемы управления памятью: стековое управление памятью.

Правила аттестации студентов по учебной дисциплине Языки программирования и методы трансляции

Текущая аттестация студента в пятом семестре проводится по результатам выполнения им лабораторных работ, а также по результатам работы на лекциях.

За работу на лекциях студент получает от 0 до 8 баллов.

Баллы за каждую из трех лабораторных работ выставляются по правилам, приведенным в таблице 6.1.

Таблица 6.1.

Вид деятельности	Минимальное количество баллов	Максимальное количество баллов
Выполнение лабораторной работы (оценивается полнота выполнения задания, работоспособность разработанного программного обеспечения, полнота набора тестов, полнота отчета, его соответствие требованиям оформления, своевременность* выполнения лабораторной работы)	8	12
Защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике, своевременность* защиты)	6	12
Всего за лабораторную работу	14	24

* график выполнения и защиты лабораторных работ приведен ниже

Минимальное допустимое количество баллов за лабораторную работу получает студент, выполнивший задание не полно, оформивший отчет с недостатками и показавший на защите некоторое владение теоретическими и практическими навыками. Студент, не выполнивший и не защитивший лабораторную работу, не допускается к выполнению следующей лабораторной работы.

График выполнения и защиты лабораторных работ:

- 1 – 7 неделя – 1-я лабораторная работа;
- 8 – 12 неделя – 2-я лабораторная работа;
- 13 – 18 неделя – 3-я лабораторная работа.

Оценка за контрольные недели выставляется в соответствии с таблицей 6.2.

Таблица 6.2.

Номер недели	7 контрольная неделя			12 контрольная неделя		
Оценка за контрольную неделю	0	1	2	0	1	2
Текущий рейтинг студента	Менее 8	От 8 до 12	14 и более	Менее 14	От 22 до 36	28 и более
Количество зачтенных лабораторных заданий	0	1*	1	1	2*	2

* лабораторная работа выполнена, но не защищена

Зачет в пятом семестре сдается в виде письменных ответов на два теоретических вопроса. К зачету допускаются студенты, выполнившие и защитившие все 3 лабораторные работы и набравшие не менее 42 баллов. Максимальное количество баллов за теоретический зачет – 20 баллов, минимальное – 8. Оценивается правильность ответа, последовательность и полнота ответа, владение профессиональной терминологией. Минимальное количество баллов за теоретический зачет получает студент, частично ответивший на один из вопросов.

Текущая аттестация студента в шестом семестре проводится по результатам выполнения им лабораторных работ и расчетно-графического задания.

Баллы за каждую из четырех лабораторных работ выставляются по правилам, приведенным в таблице 6.3.

Таблица 6.3.

Вид деятельности	Минимальное количество баллов	Максимальное количество баллов
Выполнение лабораторной работы (оценивается полнота выполнения задания, работоспособность разработанного программного обеспечения, своевременность* выполнения лабораторной работы)	3	6
Оформление отчета (оценивается полнота набора тестов, полнота отчета, его соответствие требованиям оформления, своевременность* выполнения отчета)	2	4
Защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике, своевременность* защиты)	3	6
Всего за лабораторную работу	8	16

* график выполнения и защиты лабораторных работ приведен ниже

Минимальное допустимое количество баллов за лабораторную работу получает студент, выполнивший задание не полно, оформивший отчет с недостатками и показавший на защите некоторое владение теоретическими и практическими навыками. Студент, не выполнивший и не защитивший лабораторную работу, не допускается к выполнению следующей лабораторной работы.

График выполнения и защиты лабораторных работ:

- 1 – 4 неделя – 1-я лабораторная работа;
- 5 – 8 неделя – 2-я лабораторная работа;
- 9 – 14 неделя – 3-я лабораторная работа.
- 15 – 18 неделя – 4-я лабораторная работа.

Оценка за контрольные недели выставляется в соответствии с таблицей 6.4.

Таблица 6.4.

Номер недели	7 контрольная неделя			12 контрольная неделя		
Оценка за контрольную неделю	0	1	2	0	1	2
Текущий рейтинг студента	Менее 8	От 8 до 12	14 и более	Менее 22	От 22 до 32	28 и более
Количество зачтенных лабораторных заданий	0	1	2	1	2	3*

* лабораторная работа выполнена, но не защищена

Баллы за каждую расчетно-графическую работу выставляются по правилам, приведенным в таблице 6.5.

Таблица 6.5.

Вид деятельности	Минимальное количество баллов	Максимальное количество баллов
Реферат (оценивается полнота раскрытия темы и современного состояния проблемы) и защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике)	4	8
Практическая часть (оценивается качество разработанного программного обеспечения, полнота тестов)	4	8
Всего за расчетно-графическую работу	8	16

Зачет в шестом семестре сдается в виде письменных ответов тест. К зачету допускаются студенты, выполнившие и защитившие все 4 лабораторные работы РГР и набравшие не менее 40 баллов. Максимальное количество баллов за теоретический зачет – 20 баллов, минимальное – 10. Оценивается количество полных и правильных ответов на задания теста. Минимальное количество баллов за теоретический зачет получает студент, ответивший правильно на 10 заданий теста.

Таблица соответствия разных шкал оценивания

Характеристика работы студента	Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки	
«Отлично» – работа высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые	98-100	A+	отлично	зачтено
	93-97	A		

практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	90-92	A-	хорошо	
«Очень хорошо» – работа хорошая, уровень выполнения отвечает большинству требований, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	87-89	B+		
	83-86	B		
	80-82	B-		
«Хорошо» – уровень выполнения работы 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	77-79	C+		
	73-76	C		
	70-72	C-	удовлетворительно	
«Удовлетворительно» – уровень выполнения работы отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном	67-69	D+	удовлетворительно	зачтено
	63-66	D		
	60-62	D-		

сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками				
«Посредственно» – работа слабая, уровень выполнения не отвечает большинству требований, теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	50-59	E		
«Неудовлетворительно» (с возможностью пересдачи) – теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	25-49	FX	неудовлетворительно	незачтено
«Неудовлетворительно» (без возможности пересдачи) – теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	0-24	F		

Паспорт лабораторных работ

по дисциплине «Языки программирования и методы трансляции», 6 семестр

1. Методика оценки

Лабораторные работы выполняются по варианту, назначенному преподавателем. При выполнении лабораторной работы студент должен ознакомиться с методикой выполнения лабораторной работы, выполнить все пункты работы, подготовить отчет и защитить работу.

Защита происходит в устной форме в виде ответа на один контрольный вопрос.

Варианты лабораторной работы, пункты выполнения лабораторной работы и контрольные вопросы приведены в методических указаниях к лабораторным работам.

2. Критерии оценки

- Лабораторная работа считается выполнена **неудовлетворительно**, если написанная студентом программа не выполняет действия, обусловленные вариантом задания, студент при ответе на вопросы не дает определений основных понятий, оценка составляет *0-7 баллов*.
- Лабораторная работа считается выполнена на **пороговом** уровне, если написанная студентом программа выполняет основные действия, обусловленные вариантом задания, студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, оценка составляет *8-10 баллов*.
- Лабораторная работа считается выполнена на **базовом** уровне, если написанная студентом программа выполняет все действия, обусловленные вариантом задания, студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, оценка составляет *11-13 баллов*.
- Лабораторная работа считается выполнена на **продвинутом** уровне, если написанная студентом программа выполняет все действия, обусловленные вариантом задания, студент реализовал также дополнительные возможности программы по интерфейсу и/или диагностике и исправлению ошибок, студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет *14-16 баллов*.

3. Шкала оценки

Лабораторная работа считается выполнена, если оценена не менее чем на 8 баллов (из 16 возможных).

4. Задания на лабораторные работы по дисциплине «Языки программирования и методы трансляции»

Задание на лабораторную работу состоит в разработке и реализации транслятора программы, написанной на подмножестве языка Си++ (вид подмножества указан в варианте), в программу на языке Ассемблер.

ВАРИАНТЫ К ЛАБОРАТОРНЫМ РАБОТАМ

- 1) Подмножество языка C++ включает:
 - данные типа **int**;
 - инструкции описания переменных;
 - операторы присваивания, **if, if- else** любой вложенности и в любой последовательности;
 - операции $+, -, *, =, !=, <$.
- 2) Подмножество языка C++ включает:
 - данные типа **int**;
 - инструкции описания переменных;
 - операторы присваивания, **while** любой вложенности и в любой последовательности;
 - операции $+, -, <=, >=, <, >$.
- 3) Подмножество языка C++ включает:
 - данные типа **int**;
 - инструкции описания переменных;
 - операторы присваивания, **do-while** любой вложенности и в любой последовательности;
 - операции $+=, -=, +, -, ==, !=$.
- 4) Подмножество языка C++ включает:
 - данные типа **int**;
 - инструкции описания переменных;
 - операторы присваивания, **for** любой вложенности и в любой последовательности;
 - операции $+, -, *, \&\&, \parallel$.
- 5) Подмножество языка C++ включает:
 - данные типа **int**;
 - инструкции описания переменных;
 - операторы присваивания, **switch** любой вложенности и в любой последовательности;
 - операции $+, -, *, =, !=, <$.
- 6) Подмножество языка C++ включает:
 - данные типа **int, float, char**;
 - инструкции описания переменных;
 - операторы присваивания в любой последовательности;
 - операции $+, -, *, =, !=, <, >$.
- 7) Подмножество языка C++ включает:
 - данные типа **int, float, массивы** из элементов указанных типов;
 - инструкции описания переменных;
 - операторы присваивания в любой последовательности;
 - операции $+, -, *, =, !=, <, >$.
- 8) Подмножество языка C++ включает:
 - данные типа **int, float, struct** из элементов указанных типов;
 - инструкции описания переменных;
 - операторы присваивания в любой последовательности;
 - операции $+, -, *, =, !=, <, >$.

9) Подмножество языка C++ включает:

- данные типа **int, char**;
- инструкции описания переменных;
- операторы присваивания в любой последовательности;
- операции **+, −, <, >**, побитовые операции **<<, >>, &, |**.

10) Подмножество языка C++ включает:

- данные типа **int**;
- инструкции описания переменных;
- операторы присваивания в любой последовательности;
- полный набор арифметических, логических операций и операций сравнения.

11) Подмножество языка C++ включает:

- данные типа **int**;
- инструкции описания переменных и функций;
- несколько функций с параметрами и возвращаемым значением;
- операторы присваивания в любой последовательности;
- операции **+, −, =, !=, <, >**.

5. **Этапы выполнения** лабораторных работ по дисциплине «Языки программирования и методы трансляции»

Этапы выполнения лабораторной работы №1 «Проектирование и реализация таблиц, используемых в трансляторе»:

- 1) разработать структуру постоянных таблиц для хранения алфавита языка, зарезервированных слов, знаков операций, разделителей и пр.; реализовать для постоянных таблиц алгоритм поиска элемента в упорядоченной таблице;
- 2) разработать структуру переменных таблиц с вычисляемым входом для хранения идентификаторов и констант (вид хеш-функции и метод рехеширования задает разработчик);
- 3) реализовать для переменных таблиц алгоритмы поиска/добавления лексемы, поиска/добавления атрибутов лексемы;
- 4) разработать программу для тестирования и демонстрации работы программ пп.1-3.

Этапы выполнения лабораторной работы №2 «Разработка и реализация блока лексического анализа»:

- 1) разработать соответствующий варианту детерминированный конечный автомат.
- 2) реализовать лексический анализатор на основе детерминированного конечного автомата. Исходными данными для сканера является программа на языке C++ и постоянные таблицы, реализованные в лабораторной работе №1. Результатом работы сканера является создание файла токенов, переменных таблиц (таблицы символов и таблицы констант) и файла сообщений об ошибках.

Этапы выполнения лабораторной работы №3 «Разработка и реализация блока лексического анализа»:

- 1) Сконструировать КС-грамматику в соответствии с вариантом задания.
- 2) В случае несоответствия построенной грамматики требованиям выбранного табличного метода разбора следует провести эквивалентные преобразования грамматики либо выбрать другой метод разбора.
- 3) Построить таблицу разбора и запрограммировать драйвер, реализующий работу с этой таблицей реализовать возможности табличного метода по диагностике и исправлению синтаксических ошибок в исходной программе.

Этапы выполнения лабораторной работы №4 «Разработка и реализация блока генерации кода»:

- 1) В соответствии с выбранным вариантом реализовать генератор кода. Исходными данными являются: синтаксическое дерево или постфиксная запись, построенные в лабораторной работе №3; таблицы лексем. Результатом выполнения лабораторной работы является программа на языке Ассемблер.
- 2) В режиме отладки продемонстрировать работоспособность генератора кода и транслятора в целом.

6. Вопросы к защите лабораторных работ по дисциплине «Языки программирования и методы трансляции»

Контрольные вопросы к лабораторной работе №1

Проектирование и реализация таблиц, используемых в трансляторе

1. Постоянные таблицы, используемые в трансляторах.
2. Переменные таблицы, используемые в трансляторах.
3. Операции с таблицами.
4. Способы организации таблиц.
5. Хеш-функции.
6. Методы рехеширования.
7. Таблицы с вычисляемым входом.

Контрольные вопросы к лабораторной работе №2

Разработка и реализация блока лексического анализа

1. Интерпретаторы и компиляторы.
2. Основные фазы трансляции.
3. Регулярные выражения и конечные автоматы.
4. Детерминированные конечные автоматы.
5. Методы приведения недетерминированного конечного автомата к детерминированному.
6. Автоматная грамматика.
7. Диаграмма состояний сканера.

Контрольные вопросы к лабораторной работе №3

Разработка и реализация блока синтаксического анализа

1. Нисходящая стратегия разбора.
2. Восходящая стратегия разбора.
3. Синтаксические деревья.
4. Классификация языков по Хомскому.
5. *LL*-грамматика.
6. *LL*-таблица разбора.
7. *LR*-грамматика.
8. *LR*-таблица разбора.

Контрольные вопросы к лабораторной работе №4

Разработка и реализация блока генерации кода

1. Внутренние формы представления программы.
2. Постфиксная запись.
3. Представление генерируемого кода в форме четверок.
4. Организация генератора кода.
5. Среда выполнения.
6. Методы управления памятью.
7. Фазы управления памятью.