

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования «Казанский национальный исследовательский
технический университет им. А.Н. Туполева-КАИ»
Институт Компьютерных технологий и защиты информации
Кафедра Компьютерных систем

УТВЕРЖДАЮ
Ответственный за ОП
Верш И.С. Вершинин
«31» 09 2017 г.

Регистрационный № 4010-17/Б-192

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения промежуточной аттестации обучающихся по дисциплине

«Трансляция языков программирования»
(наименование дисциплины, практики)

Индекс по учебному плану: Б1.В.ДВ.08.01.

Направление подготовки: 09.03.01 «Информатика и вычислительная техника».

Квалификация: бакалавр.

Профиль подготовки: «Вычислительные машины, комплексы, системы и сети».

Вид(ы) профессиональной деятельности: научно-исследовательская, проектно-конструкторская.

Заведующий кафедрой М.П. Шлеймович

Разработчики: Д.Г. Хохлов

М.В. Медведев

Казань 2017 г.

Фонд оценочных средств для проведения промежуточной аттестации
обучающихся по дисциплине

«Трансляция языков программирования»
(наименование дисциплины, практики)

Содержание фонда оценочных средств (ФОС) соответствует требованиям федерального государственного стандарта высшего образования (ФГОС ВО) по направлению подготовки 09.03.01 «Информатика и вычислительная техника», учебному плану направления 09.03.01 «Информатика и вычислительная техника».

Разработанные ФОС обладают необходимой полнотой и являются актуальными для оценки компетенций, осваиваемых обучающимися при изучении дисциплины. Они полностью соответствуют задачам будущей профессиональной деятельности обучающихся, установленных ФГОС ВО. В составе ФОС имеются оценочные средства в виде тестовых заданий и контрольных вопросов различного уровня сложности, которые позволяют провести оценку порогового, продвинутого и превосходного уровней освоения компетенций по дисциплине.

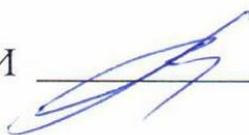
ФОС обладают необходимой степенью приближенности к задачам будущей профессиональной деятельности обучающихся, связанным со способностью применять знания, умения и навыки для решения профессиональных задач, соответствующих компетенциям, реализуемым дисциплиной.

Замечания отсутствуют.

Заключение. Учебно-методическая комиссия делает вывод о том, что представленные материалы соответствуют требованиям ФГОС ВО по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и рекомендуются для использования в учебном процессе.

Рассмотрено на заседании учебно-методической комиссии института КТЗИ от «31» августа 2017 г., протокол № 8.

Председатель УМК института КТЗИ _____ В.В. Родионов



Содержание

ВВЕДЕНИЕ	4
1 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ	5
2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.	5
3 ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ	5
4 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЯ ШКАЛЫ ОЦЕНИВАНИЯ	6
5 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРУ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ	8
6 КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ	10
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	18

Введение

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (ФОС ПА) «Системное программное обеспечение» – это комплект методических и контрольно-измерительных материалов, предназначенных для определения уровня сформированности компетенций, оценивания знаний, умений, владений (навыков) на разных этапах освоения дисциплины для проведения промежуточной аттестации обучающихся по дисциплине.

ФОС ПА является составной частью учебного и методического обеспечения программы бакалавриата по направлению 09.03.01 «Информатика и вычислительная техника».

Задачи ФОС по дисциплине:

– оценка запланированных результатов освоения дисциплины (модуля) или практики обучающимися в процессе изучения дисциплины (модуля) или практики, в соответствии с разработанными и принятыми критериями по каждому виду контроля;

– контроль и управление процессом приобретения обучающимися необходимых знаний, умений, навыков и формирования компетенций, определенных в ФГОС ВО по направлению подготовки

ФОС ПА по дисциплине сформирован на основе следующих основных принципов оценивания:

– пригодности (валидности) (объекты оценки соответствуют поставленным целям обучения);

– надежности (использования единообразных стандартов и критериев для оценивания запланированных результатов);

– эффективности (соответствия результатов деятельности поставленным задачам).

ФОС ПА по дисциплине разработан в соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений требованиям поэтапного формирования соответствующих составляющих компетенций и включает контрольные вопросы (или тесты) и типовые задания, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций.

1 Формы промежуточной аттестации по дисциплине

Дисциплина «Трансляция языков программирования» изучается в 6 семестре на третьем курсе при очной форме обучения и завершается промежуточной аттестацией в форме экзамена.

2 Оценочные средства для промежуточной аттестации

Оценочные средства для промежуточной аттестации по дисциплине.

Таблица 1

Оценочные средств для промежуточной аттестации
(очная форма обучения)

№ п/п	Семестр	Форма промежуточной аттестации	Оценочные средства
1.	6	экзамен	ФОС ПА

3 Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Перечень компетенций и их составляющих, которые должны быть сформированы при изучении темы соответствующего раздела дисциплины, представлен в таблице 2.

Таблица 2

Перечень компетенций и этапы их формирования
в процессе освоения дисциплины

№ п/п	Этап формирования (семестр)	Наименование раздела	Код формируемой компетенции (составляющей компетенции)		Форма промежуточной аттестации
1.	6	Учебная система программирования С0	ПК-1	ПК-1 З, ПК-1 У, ПК-1 В	экзамен
2.	6	Системы программирования	ПК-1	ПК-1 З, ПК-1 У, ПК-1 В	экзамен

4 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкалы оценивания

Показатели и критерии оценивания сформированности компетенций приведены в таблице 3.

Показатели и критерии оценивания сформированности компетенций

№ п/п	Этап формирования (семестр)	Код формируемой компетенции (составляющей компетенции)		Критерии оценивания	Показатели оценивания (планируемые результаты обучения)		
					Пороговый уровень	Продвинутый уровень	Превосходный уровень
1.	6	ПК-1	ПК-1 З	Теоретические знания	Знание основных понятий языка программирования, этапов трансляции и критериев оценки трансляторов	Знание основных характеристик распространенных языков программирования и трансляторов.	Иметь представление о современных языках программирования, их области применения и особенностях реализации.
2.	6	ПК-1	ПК-1 З ПК-1 У ПК-1 В	Практические навыки	Умение разрабатывать и отлаживать программы в учебной системе программирования на языке С0.	Умение составлять алгоритмы и программы фрагментов учебного транслятора	Умение составлять алгоритмы и программы для реализации структур данных учебного транслятора.
					Владение навыками разработки и отладки программ в учебной системе программирования на языке С0.	Владение навыками грамотного оформления документации к программным продуктам на примере транслятора С0.	Владение навыками составления, обоснования и оценки алгоритмов и программ для реализации структур данных учебного транслятора С0.

Формирование оценки при промежуточной аттестации по итогам освоения дисциплины зависит от уровня освоения компетенций, которые обучающийся должен освоить по данной дисциплине. Связь между итоговой оценкой и уровнем освоения компетенций (шкала оценивания) представлена в таблице 5.

Таблица 5

Описание шкалы оценивания

Описание оценки в требованиях к уровню и объему компетенций	Выражение в баллах	Словесное выражение
Освоен превосходный уровень усвоения компетенций	от 86 до 100	Зачтено (отлично)
Освоен продвинутый уровень усвоения компетенций	от 71 до 85	Зачтено (хорошо)
Освоен пороговый уровень усвоения компетенций	от 51 до 70	Зачтено (удовлетворительно)
Не освоен пороговый уровень усвоения компетенций	до 51	Не зачтено (не удовлетворительно)

5 Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формирование оценки по результатам текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины приведено в таблице 6.

Формирование оценки по итогам освоения дисциплины (модуля) или практики

Наименование контрольного мероприятия	Рейтинговые показатели				
	I атте- ста- ция	II атте- ста- ция	III атте- ста- ция	по ре- зульта- там текуще- го кон- троля	по итогам промежу- точной аттестации
Раздел 1	20			20	
Тест текущего контроля по разделу	4			4	
Защита лабораторных работ	16			16	
Раздел 2		30		30	
Тест текущего контроля по разделу		6		6	
Защита лабораторных работ		24		24	
Промежуточная аттестация (экзамен):					50
– тест промежуточной аттестации по дисциплине					10
– в письменной форме по билетам					40

6 Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины

6. Тестовые задания промежуточной аттестации

6.1. Тестовые задания

К динамическому связыванию относятся

- +выполнение команды компьютера
- выделение памяти глобальным переменным программы
- +выделение памяти локальным переменным программы
- выбор типа переменной
- выбор компилятора
- +загрузка DLL

 К статическому связыванию относятся

- выделение памяти переменным при входе в блок, где они описаны
- +выбор имени переменной
- присваивание значения локальной переменной
- +выбор языка программирования
- выполнение команды компьютера
- +выделение памяти для загрузки программы

Инфиксное выражение $12 / (5 * (X - 2)) + 7$ в постфиксной записи (без изменения порядка операндов) имеет вид %% $12\ 5\ X\ 2\ -\ *\ / 7\ +$

Инфиксное выражение $12 / (5 * (X - 2)) + 7$ в префиксной записи (без изменения порядка операндов) имеет вид %% $+ / 12\ *\ 5\ -\ X\ 2\ 7$

Постфиксное выражение $A\ 6\ *\ B\ -\ 9\ C\ + /$ в инфиксной записи (без изменения порядка операндов) имеет вид %% $(A\ *\ 6\ -\ B) / (9\ +\ C)$

Постфиксное выражение $A\ 6\ *\ B\ -\ 9\ C\ + /$ в префиксной записи (без изменения порядка операндов) имеет вид %% $/\ -\ *\ A\ 6\ B\ +\ 9\ C$

Префиксное выражение $+ *\ 4\ -\ P\ 3 / T\ 2$ в инфиксной записи (без изменения порядка операндов) имеет вид %% $4\ *\ (P\ -\ 3) + T / 2$

Префиксное выражение $+ *\ 4\ -\ P\ 3 / T\ 2$ в постфиксной записи (без изменения порядка операндов) имеет вид %% $4\ P\ 3\ -\ *\ T\ 2\ + /$

Н. Хомский классифицировал порождающие грамматики
по сложности грамматического разбора
*по виду правил
по сложности порождения предложений

К лексическому анализу относится
–удаление из текста всех пробелов
+контроль правильности лексем

- упорядочение лексем
- +удаление всех комментариев
- +преобразование текста в последовательность лексем

К лексемам относятся

- +целочисленные константы
- выражения
- +служебные слова
- +идентификаторы
- комментарии

Результатом синтаксического анализа является

- *дерево разбора
- объектный код программы
- таблица предложений
- список операторов

Система программирования включает

- язык(и) управления заданиями
- +язык(и) программирования
- систему управления базами данных
- +редактор связей
- +интерпретатор(ы)

К этапам жизненного цикла языка программирования относится

- отладка языка
- +стандартизация языка
- изучение языка
- +реализация языка

Типы трансляторов

- +компоновщик
- коммутатор
- +конвертор
- сопроцессор
- +макропроцессор
- системный загрузчик

Преимуществом интерпретатора перед компилятором является

- быстрое выполнение программы
- простота разработки программы
- +мобильность программы
- экономия памяти при выполнении программы
- +простота разработки транслятора

Преимуществом компилятора перед интерпретатором является

- +быстрое выполнение программы
- простота разработки транслятора
- мобильность транслятора
- +экономия времени на трансляцию
- простота разработки программы

Заданы правила грамматики:

$A ::= y B \mid x$

$B ::= y \mid x A$

терминальными символами являются $\% \% x y \mid y x$

 Заданы правила грамматики:

$N ::= d \mid Nd \mid NF$

$F ::= .d \mid Fd$

нетерминальными символами являются $\% \% N F \mid F N$

Заданы правила грамматики:

$B ::= y \mid A x$

$A ::= y B \mid x$

Грамматика относится к типу

0

1

*2

3

 Заданы правила грамматики:

$F ::= .d \mid Fd$

$N ::= d \mid Nd \mid F$

Грамматика относится к типу

0

1

2

*3

Укажите соответствие трансляторов и свойств
интерпретатор ~ простая разработка транслятора
компилятор ~ сложная разработка транслятора

Заданы правила грамматики с аксиомой A:

$A ::= y B \mid x$

$B ::= y \mid x A$

Предложением соответствующего языка является

xx

xA

*yy

Заданы правила грамматики с аксиомой B:

$B ::= y \mid A x$

$A ::= y B \mid x$

Предложением соответствующего языка является

*xx

xA

yx

 Заданы правила грамматики с аксиомой N:

$N ::= d \mid Nd \mid NF$

$F ::= .d \mid Fd$

Предложением соответствующего языка является

*dd
Nd
.d

 Заданы правила грамматики с аксиомой F:

F ::= .d | Fd

N ::= d | Nd | F

Предложением соответствующего языка является

d

Fd

*.dd

Грамматика учебного языка C0 относится к типу

0

1

*2

3

К критериям оценки языка программирования относятся

–простота отладки

+надежность

–эффективность по памяти

+ортогональность понятий

–функциональность

+стоимость использования языка

Редактор связей предназначен

*для объединения объектных модулей в модуль загрузки

для связи программы с операционной системой

для отладки связей между модулями программы

Методом стека с приоритетами можно

+интерпретировать инфиксное выражение

–компилировать постфиксное выражение

–вычислить префиксное выражение

+транслировать всю C0-программу

Метаязык служит для

–описания метаданных

+описания языка программирования

–описания объектного языка транслятора

+описания командного языка операционной системы

Метод рекурсивного спуска служит

для разработки рекурсивных программ

*для разработки программ обработки текстов

для разработки программ «сверху вниз»

Укажите соответствие языка и свойств программ

язык ассемблера ~ быстрое выполнение программы

язык высокого уровня ~ быстрая разработка программы

В учебном трансляторе С0 лексическим анализом занимаются

+подпрограмма chleks

–подпрограмма pozic

+подпрограмма chsim

–подпрограмма virag

В учебном трансляторе С0 синтаксическим анализом занимаются

+подпрограмма orfun

–подпрограмма gen_kom

+подпрограмма virag

–подпрограмма vkluch

 В учебном трансляторе С0 семантическим анализом занимаются

+подпрограмма operatr

–подпрограмма vkluch

+подпрограмма operac

–подпрограмма chsim

В учебном трансляторе С0 генерацией кода занимаются

+подпрограмма operatr

–подпрограмма pozic

+подпрограмма kodret

–подпрограмма chleks

В учебном трансляторе С0 метод рекурсивного спуска используется

+в подпрограмме operatr

–в подпрограмме pozic

+в подпрограмме orfun

–в подпрограмме virag

В учебном трансляторе С0 диагностикой ошибок занимаются

+подпрограмма operatr

–подпрограмма pozic

–подпрограмма kodret

+подпрограмма chleks

 Учебный транслятор С0

является компилятором

является интерпретатором

*является компилятором и частично интерпретатором

6.2. Примеры заданий для промежуточной аттестации

1. Составить макроопределение, заменяющее макрокоманду с заданным списком чисел вида

dw_m <10,7, ...,23>

на последовательность команд вида

dw 10

dw 10*10

dw 7

dw 7*7

...

```
dw      23
dw      23*23
```

2. Составить определение макроса `abs(X)`, заменяемого на `X` или `-X` в зависимости от знака параметра `X`.

3. Задан фрагмент программы для ANSI-символов и строк, например:

```
#include <string.h>
char sim, text1[80], text2[80];
...
strcpy(text1, text2);
text[10]=sim;
```

Переписать этот фрагмент так, чтобы можно было при необходимости путем перетрансляции получать UNICODE-версию этой же программы.

4. Указать терминальные и нетерминальные символы и аксиому грамматики с заданными правилами.

5. Привести пример предложения длиной не менее пяти символов для языка, определенного заданной грамматикой. Привести вывод этих предложений и изобразить соответствующее дерево разбора.

6. Описать в виде формальной грамматики или на метаязыке МБНФ (БНФ, синтаксических диаграмм) синтаксис входного текста заданного вида.

а) Целое десятичное число, перед которым может быть знак `+` или `-`.

б) Строка из нулей и единиц, содержащая нечетное число единиц.

в) Строка из правильно расставленных открывающих и закрывающих скобок, например: `(())`.

7. Составить в виде формальной грамматики или на заданном метаязыке (МБНФ, БНФ, синтаксических диаграмм) правила грамматики, эквивалентной грамматике, заданной на другом из перечисленных выше метаязыков.

8. Определить, к какому типу по классификации Н. Хомского относится заданная грамматика и определяемый ею язык. Если можно построить эквивалентную грамматiku более простого типа, то язык следует отнести к этому более простому типу.

9. Построить автомат-распознаватель для языка с заданной грамматикой.

10. Составить грамматику, описывающую язык, распознаваемый заданным автоматом.

11. Составить С0-программы для решения следующих задач.

1) Подсчитать количества повторений во входном тексте сочетаний "РФ" и "РТ".
Указание: использовать переменную "символ, предшествующий текущему".

2) Вывести входной текст на экран с нумерацией строк.

3) Вывести входной текст на экран, заменяя строчные латинские буквы заглавными.

Указание: использовать тот факт, что в коде ASCII коды строчных латинских букв возрастают по алфавиту через 1, и таким же свойством обладают коды заглавных букв.

4) Входной текст состоит из слов, разделенных пробелами. Подсчитать количество слов и их среднюю длину.

Указание: использовать переменную "символ, предшествующий текущему".

5) Входной текст состоит из слов, разделенных пробелами. Найти максимальную длину слова.

6) Входной текст состоит из слов, разделенных пробелами. Подсчитать количество слов, начинающихся с буквы 'a'.

Указание: использовать переменную "символ, предшествующий текущему".

7) Входной текст состоит из слов, разделенных пробелами. Подсчитать количество слов, кончающихся буквой 'a'.

Указание: использовать переменную "символ, предшествующий текущему".

8) Вывести таблицу кодов символов, лежащих в диапазоне, задаваемом двумя вводимыми символами.

9) Подсчитать во входном тексте количество слов, у которых первый символ совпадает с последним.

Указание: использовать переменную "символ, предшествующий текущему".

10) Составить функцию ввода целого числа со знаком `getn()`, которая пропускает пробелы и символы новой строки перед числом, читает число и выдает его значение в качестве значения функции. Пример использования этой функции: `x=getn()`;

Указание: использовать в качестве аналога функцию `chislo()` из программы вычисления выражения (программа 4.1 из раздела 4.3.4).

11) Дано натуральное число N и N целых чисел. Найти сумму заданных целых чисел. Для ввода чисел использовать функцию `getn()` из задачи 6, вариант 10.

12. Получить объектный код на языке ассемблера для заданного оператора языка C0.

1) **while** (100>a) a = (b-6)*a + c;

где a, b – параметры; c – глобальная переменная.

2) **if** (f(c+d) < 3*d) s = 20;

где f – функция; c, s – глобальные переменные; d – локальная переменная.

3) **return** a*(a>b+4) – 8*(a<=b+4);

где a – глобальная переменная; b - локальная переменная.

4) p(5-x, a+10*(b-c/8));

где p – функция; a – глобальная переменная; x – параметр; b, c – локальные переменные.

13. Составить трассировочную таблицу компиляции на язык ассемблера оператора языка C0: `pr (b+4, (a+7*b)/c)`;

где pr – функция; a – глобальная переменная; b – параметр; c – локальная переменная.

14. Построить дерево синтаксического анализа (грамматического разбора) заданного оператора языка C0:

1) **while** (x!=5) x = x - y/(z+10);

2) **if** (x==y+1) putchar(c+v/8);

15. Представить в двух других формах записи

1) инфиксное выражение $9 - 8 / (a + 1) * 3$

2) префиксное выражение $- / a 2 * 3 + b 8$

3) постфиксное выражение $x 1 y + 4 / 5 - *$

Лист регистрации изменений и дополнений

№ п/п	№ стра- ницы вне- сения изме- не- ний	Дата вне- сения изме- не- ний	Содержание изменений	«Согласова- но» зав. каф., реализую- щей дисци- плину	«Согла- совано» председа- тель УМК ИКТЗИ