

Программу составил(и):

кфмн, Доцент, Бужан Виталий Викторович

Рецензент(ы):

дтн, профессор кафедры информационных систем и программирования КубГТУ, Видовский Л.А.; директор АО «ЮГ-СИСТЕМА ПЛЮС», Глебов О.В.

Рабочая программа дисциплины

Конструирование компиляторов

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.04 Программная инженерия (приказ Минобрнауки России от 19.09.2017 г. № 932)

составлена на основании учебного плана:

09.04.04 Программная инженерия

утвержденного учёным советом вуза от 17.04.2023 протокол № 9.

Рабочая программа одобрена на заседании кафедры

Кафедра математики и вычислительной техники

Протокол от 11.12.2023 г. № 5

Зав. кафедрой Исикова Наталья Павловна

Согласовано с представителями работодателей на заседании НМС, протокол №9 от 17 апреля 2023 г.

Председатель НМС проф. Павелко Н.Н.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	получение базовых знаний в области разработки и применения современных трансляторов и сред для разработки программ, для решения некоторых задач по обеспечению безопасного функционирования программ,
1.2	для решения которых применяются трансляторы.
Задачи: формальные языки и грамматики, конечные автоматы, трансляция языков программирования	

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.О
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Системы автоматизированного проектирования (САПР)
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Выполнение и защита выпускной квалификационной работы
2.2.2	Подготовка к сдаче и сдача государственного экзамена
2.2.3	Производственная практика: Преддипломная практика

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ, ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ и планируемые результаты обучения	
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;	
ОПК-2.1: Знать современные интеллектуальные технологии для решения профессиональных задач	
Знать	
Уровень 1	Минимально допустимый уровень знаний современных интеллектуальных технологий для решения профессиональных задач
Уровень 2	Уровень знаний современных интеллектуальных технологий для решения профессиональных задач в объеме, соответствующем программе подготовки, допущено несколько негрубых ошибок
Уровень 3	Уровень знаний современных интеллектуальных технологий для решения профессиональных задач в объеме, соответствующем программе подготовки без ошибок
ОПК-2.2: Уметь обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач	
Уметь	
Уровень 1	Продемонстрированы основные умения обоснования выбора современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач, решены типовые задачи с негрубыми ошибками, выполнены все задания, но не в полном объеме
Уровень 2	Продемонстрированы все основные умения обоснования выбора современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач, решены все типовые задачи с негрубыми ошибками, выполнены все задания в полном объеме, но некоторые с недочетами
Уровень 3	Продемонстрированы все основные умения обоснования выбора современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме
ОПК-2.3: Иметь навыки разработки оригинальных программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	
Владеть	
Уровень 1	Имеется минимальный набор навыков разработки оригинальных программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач с негрубыми ошибками и некоторыми недочетами
Уровень 2	Продемонстрированы базовые навыки разработки оригинальных программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач с некоторыми недочетами
Уровень 3	Продемонстрированы базовые навыки разработки оригинальных программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач без ошибок и недочетов

ПК-7: Способен проектировать трансляторы и интерпретаторы языков программирования	
ПК-7.1: Знает методы проектирования трансляторов и интерпретаторов языков программирования	
Знать	
Уровень 1	Минимально допустимый уровень знаний методов проектирования трансляторов и интерпретаторов языков программирования

Уровень 2	Уровень знаний методов проектирования трансляторов и интерпретаторов языков программирования в объеме, соответствующем программе подготовки, допущено несколько негрубых ошибок
Уровень 3	Уровень знаний методов проектирования трансляторов и интерпретаторов языков программирования в объеме, соответствующем программе подготовки без ошибок
ПК-7.2: Умеет использовать методы проектирования трансляторов и интерпретаторов языков программирования	
Уметь	
Уровень 1	Продемонстрированы основные умения использования методов проектирования трансляторов и интерпретаторов языков программирования, решены типовые задачи с негрубыми ошибками, выполнены все задания, но не в полном объеме
Уровень 2	Продемонстрированы все основные умения использования методов проектирования трансляторов и интерпретаторов языков программирования, решены все типовые задачи с негрубыми ошибками, выполнены все задания в полном объеме, но некоторые с недочетами
Уровень 3	Продемонстрированы все основные умения использования методов проектирования трансляторов и интерпретаторов языков программирования, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература и эл. ресурсы	Практ. подг.
	Раздел 1. Теоретические основы трансляции					
1.1	Формальные языки и грамматики /Лек/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.2	Формальные языки и грамматики /Пр/	2	6	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.3	Формальные языки и грамматики /Ср/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.4	Автоматные грамматики и языки /Лек/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.5	Автоматные грамматики и языки /Пр/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.6	Автоматные грамматики и языки /Ср/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.7	Контекстно-свободные грамматики и языки /Лек/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.8	Контекстно-свободные грамматики и языки /Пр/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	

1.9	Контекстно-свободные грамматики и языки /Ср/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.10	Трансляция выражений /Лек/	2	6	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.11	Трансляция выражений /Пр/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
1.12	Трансляция выражений /Ср/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
Раздел 2. Трансляция языков программирования						
2.1	Лексический и синтаксический анализаторы /Лек/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.2	Лексический и синтаксический анализаторы /Ср/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.3	Структура компилятора /Лек/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.4	Структура компилятора /Ср/	2	8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.5	Контекстный анализ /Лек/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.6	Контекстный анализ /Ср/	2	4	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.7	Генерация кода /Лек/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	
2.8	Генерация кода /Ср/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5	

2.9	Трансляция процедур /Лек/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5
2.10	Трансляция процедур /Пр/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5
2.11	Трансляция процедур /Ср/	2	2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5
Раздел 3. Промежуточная аттестация					
3.1	Консультация /Консл/	2	1	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5
3.2	Экзамен /КАЭ/	2	0,3	ОПК-2.1 ОПК-2.2 ОПК-2.3 ПК-7.1 ПК-7.2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э3 Э4 Э5

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Контрольные вопросы и задания

Примерный перечень вопросов к экзамену

1. Общие сведения о трансляторах. Основные понятия и определения.
2. Общие особенности языков программирования и трансляторов.
3. Обобщенная структура транслятора.
4. Варианты взаимодействия блоков транслятора. Многопроходная организация взаимодействия блоков транслятора.
5. Варианты взаимодействия блоков транслятора. Однопроходная организация взаимодействия блоков транслятора.
6. Варианты взаимодействия блоков транслятора. Комбинированные взаимодействия блоков транслятора.
7. Способы определения языков. Формальные грамматики. Терминалы и нетерминалы языков программирования.
8. Терминалы и нетерминалы языков программирования. Выделение терминалов и нетерминалов языка программирования. Примеры.
9. Теория формальных грамматик. Грамматики с ограничениями на правила. Примеры.
10. Способы записи синтаксиса языка. Метаязыки Хомского, Хомского-Шутценберже.
11. Способы записи синтаксиса языка. Бэкуса-Наура формы, расширенные Бэкуса-Наура формы.
12. Способы записи синтаксиса языка. Диаграммы Вирта.
13. Распознаватели. Обобщенная структура распознавателя.
14. Конечный автомат как модель дискретных систем.
15. Организация лексического анализа. Назначение и необходимость фазы лексического анализа.
16. Организация лексического анализа. Транслитератор.
17. Грамматики и распознаватели для лексического анализа. Связь между диаграммой Вирта и конечным автоматом.
18. Связь между диаграммами Вирта и праволинейными граммами. Преобразование правой рекурсии в итерацию.
19. Связь между диаграммами Вирта и граммами с левой рекурсией. Преобразование левой рекурсии в итерацию.
20. Методы лексического анализа. Организация непрямого лексического анализатора.
21. Методы лексического анализа. Организация прямого лексического анализатора.
22. Назначение синтаксического разбора. Классификация методов синтаксического разбора.
23. Методы и последовательность синтаксического разбора. Использование просмотра вперед и возвратов.
24. Организация автомата с магазинной памятью. Операции автомата. Пример.
25. Связь между S-грамматикой и автоматом с магазинной памятью. Пример.
26. Обобщенный алгоритм построения нисходящего АМП для S – грамматики. Пример.
27. Построение автомата с магазинной памятью по q-грамматике. Пример

Список заданий по дисциплине

Задание: разработать программу-генератор предложений формального языка, заданного следующими порождающими грамматиками:

- 1 $G:S \rightarrow aS|Sb|\epsilon,$
- 2 $G:S \rightarrow aS|Sb|ab,$
- 3 $G:S \rightarrow aS|Sb|ba,$
- 4 $G:S \rightarrow bS|Sa|cS|\epsilon,$
- 5 $G:S \rightarrow aaSb|aSbb|ab,$
- 6 $G:S \rightarrow SS|aSb|bSa|\epsilon,$
- 7 $G:S \rightarrow aSb|bSc|cSa|\epsilon,$
- 8 $G:S \rightarrow aSS|SSb|a|b,$
- 9 $G:S \rightarrow aSb|SaS|SbS|\epsilon,$
- 10 $G:S \rightarrow SS|aS|Sb|ab$

Пример постановки задачи для практического задания по реализации оптимизирующего преобразования

Постановка задачи

Целью работы является создание оптимизирующего преобразования, позволяющего повысить качество кода, генерируемого учебным компилятором на базе LLVM. Задача оптимизации кода состоит в повышении его быстродействия и/или сокращения его размера.

Предлагается разработать преобразование, которое на вход получает программу в промежуточном представлении LLVM, а на выходе

генерирует ее версию, оптимизированную по размеру и скорости выполнения путем удаления недостижимого и бесполезного кода. Например:

Первоначальная программа:

```
int main (void) {
    printf("First message \n");
    int a = 34+48;
    if (0)
        printf("Unreachable code\n");
    a = a * 2;
    printf("Second message\n");
    return 0;
}
```

Программа после удаления мертвого и недостижимого кода:

```
int main (void) {
    printf("First message \n");
    printf("Second message\n");
    return 0;
}
```

Решение задачи

Теоретические аспекты

Предполагается реализация алгоритмов удаления мертвого и недостижимого кода. Недостижимым кодом называют часть кода программы, которая ни при каких условиях не может быть исполнена, поскольку является недостижимой в графе потока управления.

Мертвый (неиспользуемый, бесполезный) код - команды, вычисляющие никогда не используемые значения.

Практические аспекты

Решения должны быть написаны на языке C++ с использованием контейнеров и алгоритмов стандартной библиотеки, а также средств, предоставляемых LLVM. Оптимизирующие преобразования должны быть выполнены в учебном компиляторе, который представляет собой

модифицированную версию LLVM, не содержащую оптимизирующих преобразований.

Пример сборки LLVM в ОС Linux:

В каталоге с исходными кодами следует создать каталог с названием build и перейти в него:

```
$ cd llvm/
$ mkdir build
$ cd build
```

Затем следует осуществить сборку LLVM с указанием каталога сборки и желаемого типа сборки (Debug/Release). Сборка осуществляется с

помощью утилит configure и make. Ключи configure:

```
--prefix=<путь к каталогу инсталляции>
--disable-assertions/--enable-assertion – включить/выключить проверку утверждений
--enable-optimized/--disable-optimized – выбрать тип сборки: оптимизированная (Release) или без оптимизаций (Debug)
```

Для ускорения компиляции рекомендуется использовать ключ “-jN” утилиты make, где N -- число желаемых потоков сборки (как правило,

равное количеству ядер процессора или количеству ядер процессора + 1).

Пример сборки Debug с выключенной проверкой утверждений и установкой в каталог </home/user/llvm> и сборкой в 8 потоков:

```
$ ../configure --prefix=/home/user/llvm --disable-optimized --disable-assertions
```

```

$ make -j8
$ makeinstall

```

После завершения установки и компиляции требуется добавить путь до каталога с LLVM в переменную окружения PATH:

```

$ export PATH=/home/user/llvm/bin:$PATH

```

Получить промежуточное представление LLVM для программы можно, выполнив команду:

```

$ clang -c -O0 -emit-llvm test.c -o test.bc

```

Перевод бинарного представления в ассемблер LLVM

```

$ llvmdis test.bc -o test.ll

```

Ассемблирование в бинарное представление

```

$ llvmas test.ll -o newTest.bc

```

Запуск преобразования pass_name из динамической библиотеки pass_name.so

```

$ opt -load<путь/до/динамической/библиотеки/>pass_name.so -<pass_name>test.bc -otransformedTest.bc

```

test.bc - файл, содержащий бинарную версию промежуточного представления LLVM.
test.ll – файл, содержащий ассемблер LLVM в читаемом виде.

Написание оптимизирующего прохода LLVM

Для выполнения работы могут потребоваться проходы следующих типов: FunctionPass (по функциям), ModulePass (по модулям), BasicBlockPass (по базовым блокам).

В качестве примера приведен проход по функциям и именем Hello, вызываемым из командной строки с помощью ключа “-hello”:

```

#include "llvm/Pass.h"
#include "llvm/IR/Function.h"
#include "llvm/Support/raw_ostream.h"
using namespace llvm;
namespace {
struct Hello : public FunctionPass {
static char ID;
Hello() : FunctionPass(ID) {}
virtual bool runOnFunction(Function &F) {
errs() << "Hello: ";
errs().write_escaped(F.getName()) << '\n';
return false;
}
};
}
char Hello::ID = 0;
static RegisterPass<Hello> X("hello", "Hello World Pass", false, false);

```

Листинг 1. Пример компиляторного прохода

Данный код выводит на экран сообщения вида “Hello: <function_name>”.

Более детально изучить построение оптимизирующих проходов можно посмотреть в документации на компилятор (страница WritingAnLLVMPass.html в каталоге docs в поставке учебного компилятора).

Тестирование

На личной странице расположена форма загрузки файла, а также информация о результатах тестирования и минимальный набор синтетических тестов.

Загрузка решения. Загружаемый файл должен представлять собой текст программы на языке C++, содержащий компиляторный проход, запускаемый по ключу “-dce”. После загрузки решение будет скомпилировано и запущено с помощью утилиты “opt” на тестовом наборе данных.

Оценивается последнее присланное решение.

Для тестирования на локальной машине во время разработки предлагается осуществлять компиляцию программы с уровнем оптимизации “O0”. Локальное тестирование предлагается осуществлять с помощью программ с открытыми исходными кодами и минимального набора синтетических тестов.

Оценка

Обязательное условие:

Присланное решение должно проходить проверку на корректность на программах SQLite и LZMA.

Оценка эффективности будет производиться на расширенном синтетическом наборе тестов.

Необходимым условием получения оценки «Удовлетворительно» является реализация удаления бесполезного кода на уровне функций.

Необходимым условием получения оценки «Хорошо» является реализация удаления бесполезного и недостижимого кода на уровне функций. Удаление избыточных вызовов “pure” функций (не имеющих побочных эффектов и зависящих только от аргумента), например в выражении double y=sin(x), если y нигде более не используется вызов sin(x) можно удалить.

Необходимым условием получения оценки «Отлично» является реализация удаления бесполезного и недостижимого кода на уровне функций. Реализован алгоритм удаления частично избыточного кода, алгоритм описан в публикации: Jens Knoor, Oliver Rütting, and Bernhard Steffen. 1994. Partial dead code elimination. SIGPLAN Not. 29, 6 (June 1994), 147-158. DOI=10.1145/773473.178256 <http://doi.acm.org/10.1145/773473.178256>.

Итоговая оценка определяется по результатам защиты выполненного задания перед комиссией. На защиту предоставляется письменный отчет, мультимедийная презентация, сопровождаемая устными пояснениями.

5.2. Темы письменных работ

Рефераты по дисциплине

Формой осуществления контроля выполнения самостоятельной работы является подготовки рефератов на актуальные темы, т. е. изучение с помощью научных методов явлений и процессов, анализа влияния на них различных факторов, а также, изучение взаимодействия между явлениями, с целью получения убедительно доказанных и полезных для науки и практики решений с максимальным эффектом.

Цель реферата – определение конкретного объекта и всестороннее, достоверное изучение его структуры, характеристик, связей на основе разработанных в науке принципов и методов познания, а также получение полезных для деятельности человека результатов, внедрение в производство с дальнейшим эффектом.

Основой разработки каждой темы является методология, т. е. совокупность методов, способов, приемов и их определенная последовательность, принятая при разработке научного исследования. В конечном счете, методология – это схема, план решения поставленной научно-исследовательской задачи.

Процесс подготовки реферат состоит из следующих основных этапов:

1. Выбор темы и обоснование ее актуальности.
2. Составление библиографии, ознакомление с законодательными актами, нормативными документами и другими источниками, относящимися к теме проекта (работы).
3. Разработка алгоритма исследования, формирование требований к исходным данным, выбор методов и инструментальных средств анализа.
4. Сбор фактического материала.
5. Обработка и анализ полученной информации с применением современных методов анализа.
6. Формулировка выводов и выработка рекомендаций.
7. Оформление работы в соответствии с установленными требованиями.

Тематики рефератов

1. Современные языки программирования.
2. Обзор современных интерпретаторов
3. Виртуальные машины
4. Многопроходные и однопроходные трансляторы
5. Языки-концепции
6. Языки интернета
7. Сравнительная оценка современных языков программирования

Курсовые работы

Курсовая работа – это самостоятельное научное исследование, выполняемое студентом в соответствии с учебным планом, служащее углубленному познанию избранного предмета и являющееся одной из форм отчетности студента по итогам обучения за соответствующий семестр. Самостоятельность курсовой работы означает, что представленная в ней позиция выражает взгляды студента-автора работы.

Научность исследования выражается в решении им некоторой познавательной проблемы, соотнесении теоретических положений с фактами, систематичности изложения, оперировании современной специальной терминологией.

Цель курсового проекта – изучение, обобщение, систематизация и апробация методики анализа предмета исследования на материалах объекта исследования и выявление резервов улучшения результатов его деятельности. Предмет исследования определяется темой курсового проекта, а объект исследования – предприятие.

Задачи курсового проекта формулируются в форме этапов, которые нужно пройти на пути к достижению цели курсовой работы.

Курсовой проект должен состоять из следующих структурных элементов:

- Титульный лист.
- Реферат.
- Содержание (перечень всех заголовков в соответствии с планом работы и с указанием начальных страниц).
- Введение.
- Теоретические и методические основы анализа (в соответствии с темой работы).
- Организационно-экономическая характеристика предприятия.
- Разделы и подразделы основной части работы в соответствии с её темой.
- Заключение.
- Список использованных источников.
- Приложения (если требуется).

Методические указания по оформлению и содержанию курсового проекта по дисциплине «Проектирование и архитектура программных систем» предлагаются

5.3. Фонд оценочных средств

Если оператор языка ассемблера отображается при трансляции чаще всего в одну машинную инструкцию, предложения языков более высокого уровня отображаются

Ответ:

- (1) в одну машинную инструкцию

- (2) в несколько машинных инструкций
- (3) в пустую машинную инструкцию
- (4) в произвольную машинную инструкцию

Трансляторы бывают следующих типов:

Ответ:

- (1) compiler
- (2) interpreter
- (3) analysis
- (4) synthesis

Какая часть компилятора разбивает исходную программу на составляющие ее элементы и создает промежуточное представление исходной программы:

Ответ:

- (1) analysis
- (2) synthesis
- (3) interpreter
- (4) begin

Можно сказать, что результатом работы интерпретатора является:

Ответ:

- (1) "программа"
- (2) "код"
- (3) "число"
- (4) "исполняемый файл"

Для интерпретатора верны следующие утверждения:

Ответ:

- (1) анализирует программу на входном языке
- (2) создает промежуточное представление
- (3) не создает никакой новой программы
- (4) выполняет операции, содержащиеся в тексте программы

Цепочка символов, составляющая исходную программу на языке программирования является:

Ответ:

- (1) входом компилятора
- (2) выходом компилятора
- (3) процессом компиляции
- (4) процессом интерпретации

Крайне важной частью процесса трансляции является:

Ответ:

- (1) исправление ошибок, допущенных во входной программе
- (2) игнорирование ошибок, допущенных во входной программе
- (3) точная диагностика ошибок, допущенных во входной программе

Объектная программа может быть:

Ответ:

- (1) последовательностью абсолютных машинных команд
- (2) последовательностью перемещаемых машинных команд
- (3) программой на языке ассемблера
- (4) программой на некотором другом языке

Создание единого перемещаемого объектного сегмента из набора различных сегментов осуществляется программой, которая называется:

Ответ:

- (1) редактором сегментов
- (2) редактором связей
- (3) редактором объектов
- (4) редактором наборов

Подход при котором применяется трансляция программы в ассемблер:

Ответ:

- (1) упрощает конструирование компилятора
- (2) сокращает технологическую цепочку выполнения программы
- (3) удлиняет технологическую цепочку выполнения программы
- (4) усложняет конструирование компилятора

Преимуществами трансляции в ассемблер являются:

Ответ:

- (1) уровень ассемблера выше, чем у машинного кода
- (2) использование ассемблера позволяет отследить целый ряд ошибок
- (3) порождаемый текст на ассемблере значительно читабельней, чем машинный код

Для представления компилятора мы можем использовать так называемые:

Ответ:

- (1) P-диаграммы
- (2) T-диаграммы
- (3) R-диаграммы
- (4) S-диаграммы
- (5) E-диаграммы
- (6) D-диаграммы

Написание компилятора может потребоваться в следующих условиях:

Ответ:

- (1) для различных языков
- (2) для целевых платформ
- (3) при создании новой компьютерной архитектуры

Методиками разработки компиляторов являются следующие:

Ответ:

- (1) метод раскрутки
- (2) метод генерации
- (3) использование кросс-трансляторов
- (4) использование виртуальных машин
- (5) компиляция "на лету"

В каком году Вирт написал с использованием раскрутки транслятор языка Pascal:

Ответ:

- (1) 1969
- (2) 1970
- (3) 1971
- (4) 1972
- (5) 1973

Для того, чтобы справиться с проблемой большой потери времени при написании и отладке компилятора на языке ассемблера был разработан:

Ответ:

- (1) метод раскрутки
- (2) метод упаковки
- (3) метод генерации
- (4) метод распаковки

Под переносимой (portable) программой понимается программа, которая:

Ответ:

- (1) может без перетрансляции выполняться на одной платформе
- (2) не может без перетрансляции выполняться на нескольких платформах
- (3) может без перетрансляции выполняться на нескольких платформах

Компиляторы генерирующие объектную программу на языке более высокого уровня, чем язык ассемблера называют:

Ответ:

- (1) конвертерами
- (2) генераторами
- (3) кросс-компиляторами
- (4) исполнителями

Одна из первых широко известных виртуальных машин была разработана в 70-х годах Н. Виртом:

Ответ:

- (1) при написании компилятора Pascal-P
- (2) при написании компилятора Pascal
- (3) при написании компилятора Simula-P
- (4) при написании компилятора Simula

Сегодня идея виртуальных машин приобрела широкую известность благодаря языку:

Ответ:

- (1) Java

- (2) C++
- (3) C
- (4) Fortran

Компиляторы языка Java генерируют:

Ответ:

- (1) бит-код
- (2) объектный код
- (3) байт-код

Для того, чтобы увеличить скорость работы приложений, была разработана технология:

Ответ:

- (1) compiling
- (2) Just-In-Time compiling
- (3) Just compiling
- (4) Fats compiling

Использование какой связки позволяет заметно повысить скорость выполнения исходной программы:

Ответ:

- (1) "компилятор+интерпретатор+JIT-компилятор"
- (2) "интерпретатор+JIT-компилятор"
- (3) "компилятор+интерпретатор"
- (4) "компилятор+интерпретатор+JIT-компилятор+интепретатор"

Процесс создания компилятора можно свести к решению нескольких задач, которые принято называть:

Ответ:

- (1) compilation steps
- (2) compilation phases
- (3) compilation rounds
- (4) compilation stages

Обычно компилятор состоит из следующих фаз:

Ответ:

- (1) лексический анализ
- (2) синтаксический анализ
- (3) видозависимый анализ
- (4) оптимизация
- (5) генерация кода

В разборе входной цепочки и выделении некоторых более "крупных" единиц, которые удобнее для последующего разбора заключается задача:

Ответ:

- (1) фазы синтаксического анализа
- (2) фазы видозависимого анализа
- (3) фазы генерации кода
- (4) фазы лексического анализа

На этапе лексического анализа обычно выполняются такие действия, как:

Ответ:

- (1) удаление комментариев
- (2) обработка директив условной компиляции
- (3) игнорирование комментариев
- (4) обработка операторов

После синтаксического анализа можно считать, что исходная программа преобразована:

Ответ:

- (1) в некоторое промежуточное представление
- (2) в некоторое промежуточное состояние
- (3) в объектное представление
- (4) в исполняемый файл

В дереве разбора программы внутренние узлы соответствуют:

Ответ:

- (1) операциям
- (2) операндам
- (3) классам
- (4) подклассам

Видозависимый анализ иногда называют:

Ответ:

- (1) semantic analysis
- (2) syntax analysis
- (3) lexical analysis
- (4) code optimization

Обязательность описания переменных может служить примером:

Ответ:

- (1) предварительных условий
- (2) контекстных условий
- (3) дополнительных условий
- (4) временных условий

Наиболее распространенными оптимизациями являются:

Ответ:

- (1) константные вычисления
- (2) уменьшение силы операций
- (3) выделение общих подвыражений
- (4) чистка циклов

На этапе генерации кода необходимо решить множество следующих сопутствующих проблем:

Ответ:

- (1) распределение памяти
- (2) распределение регистров
- (3) распределение блоков
- (4) распределение стеков

Какие фазы иногда объединяют вместе под названием front-end?

Ответ:

- (1) лексический анализ
- (2) синтаксический анализ
- (3) видозависимый анализ
- (4) некоторые оптимизации

Процесс обработки всего, возможно, уже преобразованного, текста исходной программы называется:

Ответ:

- (1) passes
- (2) control
- (3) analysis
- (4) audit

Backpatching - это:

Ответ:

- (1) внутренний интерфейс
- (2) техника "заплат"
- (3) внешний интерфейс
- (4) тестирование программы

Одна из первых задач, возникающих в процессе компиляции - это:

Ответ:

- (1) определение рассматриваемого языка программирования
- (2) определение типа необходимого семантического анализа
- (3) определение типа необходимого лексического анализа
- (4) определение типа рассматриваемого языка программирования

Идея создания некоторого обобщенного алгоритма, проверяющего за конечное число шагов принадлежность данной цепочки языку является альтернативой:

Ответ:

- (1) механизму интерпретации
- (2) механизму видозависимого анализа
- (3) механизму грамматик
- (4) механизму анализа на входе

Грамматики представляют собой:

Ответ:

- (1) наименее распространенный класс описаний языков
- (2) наиболее распространенный класс описаний языков

- (3) наиболее распространенный блок описаний языков
- (4) наименее распространенный блок описаний языков

Получение любого предложения языка в грамматике начинается с этого:

Ответ:

- (1) начальный символ
- (2) теорема
- (3) специальный символ
- (4) аксиому

В формальном определении грамматики нетерминалы обозначаются:

Ответ:

- (1) строчными буквами
- (2) пустой строкой
- (3) прописными буквами

Различные грамматики могут порождать:

Ответ:

- (1) один и тот же язык
- (2) эквивалентные языки
- (3) различные языки

Несмотря на эквивалентность определяемых языков, одна грамматика может быть значительно удобнее другой с точки зрения ее использования:

Ответ:

- (1) в интерпретаторе
- (2) в трансляторе
- (3) в виртуальной машине
- (4) в компиляторе

Определение грамматик не накладывает никаких ограничений на количество:

Ответ:

- (1) терминалов в левой части правил
- (2) нетерминалов в левой части правил
- (3) пустых строк в левой части правил

Иерархия Хомского - это классификация грамматик согласно:

Ответ:

- (1) их внешнему виду
- (2) их внутренней структуре
- (3) их времени появления
- (4) их принципов создания

Согласно иерархии Хомского, если любое правило из P имеет вид $A \rightarrow xB$ или $A \rightarrow x$, где A, B - нетерминалы, а x - терминал, то грамматика G называется:

Ответ:

- (1) выровненной вправо
- (2) без ограничений
- (3) бесконтекстной
- (4) праволинейной

Согласно иерархии Хомского, если любое правило из P имеет вид $A \rightarrow a$, где A - нетерминал, а a - нетерминал или терминал то грамматика G называется:

Ответ:

- (1) бесконтекстной
- (2) неукорачивающей
- (3) контекстно-зависимой
- (4) праволинейной
- (5) контекстно-свободной
- (6) выровненной вправо

Обобщенный алгоритм, позволяющий определить некоторое множество и использующий в своей работе следующие компоненты: входную ленту, управляющее устройство с конечной памятью и дополнительную рабочую память - это:

Ответ:

- (1) интерпретатор
- (2) распознаватель
- (3) определитель

В качестве примеров распознавателей можно назвать:

Ответ:

- (1) машину Тьюринга
- (2) конечные автоматы
- (3) бесконечные автоматы
- (4) магазинные автоматы

Путем задания некоторого множества допустимых заключительных состояний распознавателя определяется:

Ответ:

- (1) грамматика
- (2) язык
- (3) лексема
- (4) виртуальная машина

Основная часть конечного автомата - это:

Ответ:

- (1) функция перехода
- (2) функция распознавания
- (3) функция определения
- (4) функция остановки

В конечных автоматах цепочка считается принадлежащей языку, если хотя бы одна из последовательностей шагов:

Ответ:

- (1) завершается в начальном состоянии
- (2) завершается в состоянии перехода
- (3) завершается в состоянии распознавания
- (4) завершается в заключительном состоянии

Язык распознается конечным автоматом, если:

Ответ:

- (1) им распознается каждое слово языка
- (2) им распознается хотя бы одно слово языка
- (3) им не распознается ни одно слово языка
- (4) им распознается ключевое слово языка

Удобная форма записи конечных автоматов – это:

Ответ:

- (1) графы переходов
- (2) диаграммы переходов
- (3) дерево переходов
- (4) графики переходов

Следующий набросок программы: $q = q_0$;

```
c = GetChar();
```

```
while (c != eof) {
```

```
    q = move (q, c);
```

```
    c = GetChar();
```

```
}
```

```
if (q is in F) return "yes";
```

```
else return "no";
```

демонстрирует (предполагается, что входная лента заканчивается символом `end_of_file`):

Ответ:

- (1) моделирование конечного автомата
- (2) моделирование бесконечного автомата
- (3) моделирование магазинного автомата
- (4) моделирование машины Тьюринга

Два детерминированных автомата называются эквивалентными, если они:

Ответ:

- (1) распознают один и тот же язык
- (2) распознают один язык
- (3) распознают определенное множество языков
- (4) распознают специальные языки

Если мы предположим, что начальные состояния конечных автоматов эквивалентны, то мы можем получить:

Ответ:

- (1) и другие пары начальных состояний
- (2) и другие пары эквивалентных состояний

- (3) и другие пары конечных состояний
- (4) и другие пары промежуточных состояний

Следующий алгоритм: удаление всех недостижимых состояний, разбиение множества всех достижимых состояний на классы эквивалентности неразличимых состояний, из каждого класса эквивалентности берется только по одному представителю - это:

Ответ:

- (1) алгоритм минимизации
- (2) алгоритм отладки
- (3) алгоритм выделения представителей
- (4) алгоритм достижимых и недостижимых состояний

Классы языков, определяемых праволинейной грамматикой являются:

Ответ:

- (1) эквивалентными
- (2) не эквивалентными
- (3) формализмами

Класс языков, задаваемых праволинейными грамматиками, очень удобен в задачах:

Ответ:

- (1) трансляции
- (2) интерпретации
- (3) компиляции

Если существует, по крайней мере, одна выводимая в грамматике цепочка, для которой существует более одного вывода, то такая грамматика является:

Ответ:

- (1) однозначной
- (2) естественной
- (3) произвольной
- (4) неоднозначной

Любая КС-грамматика может быть приведена к нормальному виду Хомского, в котором все правила имеют один из следующих видов:

Ответ:

- (1) $A \rightarrow BC$, где A, B и C - нетерминалы
- (2) $A \rightarrow a$, где a - терминал
- (3) $AB \rightarrow a$, где a - терминал
- (4) $BC \rightarrow A$, где A, B и C - нетерминалы

В нормальной форме Грейбах все правые части правил начинаются:

Ответ:

- (1) с терминалов
- (2) с нетерминалов
- (3) с формул
- (4) с инструкций

Магазинные автоматы, известны также как:

Ответ:

- (1) автоматы с магазинной памятью
- (2) МА-автоматы
- (3) МП-автоматы

На каждом шаге работы МП-автомат может либо:

Ответ:

- (1) занести что-то в магазин
- (2) снять какие-то значения с его вершины
- (3) удалить вершину
- (4) удалить магазин

МП-автоматы обладают одним существенным недостатком:

Ответ:

- (1) они недетерминированны по своей природе
- (2) они детерминированны по своей природе
- (3) они имеют сложную структуру
- (4) они имеют простую структуру

Детерминированные МП-автоматы описывают только подмножество всего класса КС-языков - это подмножество

называется:

Ответ:

- (1) детерминированными КС-языками
- (2) недетерминированными КС-языками
- (3) детерминированными языками
- (4) недетерминированными языками

Форма Бэкуса-Наура был разработана для описания:

Ответ:

- (1) Фортрана
- (2) Паскаля
- (3) Алгола-60
- (4) Кобола

При определении синтаксиса языков Pascal и Modula-2 Вирт использовал расширенную форму Бэкуса-Наура (EBNF):

Ответ:

- (1) нетерминалы записываются как отдельные слова
- (2) символ равенства используется вместо символа ::=
- (3) символ точка используется для обозначения конца правила
- (4) комментарии заключаются между символами (* ... *)

Следующее правило: REF to MODE NEST assignation:

REF to MODE NEST destination, becomes token,

MODE NEST source.

определяет:

Ответ:

- (1) массив
- (2) цикл с постусловием
- (3) цикл с предусловием
- (4) присваивание

Синтаксические диаграммы или синтаксические схемы имеют форму:

Ответ:

- (1) блок-схем
- (2) блоков
- (3) прямоугольников
- (4) квадратов

В задачу анализа потока управления входит определение свойств:

Ответ:

- (1) передачи управления между операторами программы
- (2) передачи управления между блоками программы
- (3) передачи управления между классами программы
- (4) передачи управления между объектами программы

Анализ потока управления производится над:

Ответ:

- (1) секцией потока управления
- (2) блоком потока управления
- (3) подграфом потока управления
- (4) графом потока управления

Основное употребление анализа потока управления в оптимизации - это:

Ответ:

- (1) преобразование
- (2) проверка контекстных условий
- (3) фрагментация
- (4) проверка логических условий

Основным способом представления потока управления программы является:

Ответ:

- (1) подграф потока управления
- (2) граф потока управления
- (3) блок потока управления

Последовательность вершин, такая, что между каждой последующей и предыдущей вершиной в графе существует ребро - это:

Ответ:

- (1) дуга в графе
- (2) путь в графе
- (3) блок вершин
- (4) блок секций

Любая вершина обязательно предшествует:

Ответ:

- (1) конечной вершине
- (2) предыдущей вершине
- (3) себе самой
- (4) последующей вершине

Отношение обязательного предшествования обозначается символом:

Ответ:

- (1) '>'
- (2) '#'
- (3) '\$'
- (4) '<'

Отношение обязательного предшествования:

Ответ:

- (1) рефлексивно
- (2) симметрично
- (3) транзитивно

Отношение непосредственного предшествования - это:

Ответ:

- (1) дуга
- (2) вершина
- (3) подграф
- (4) дерево

Задачи проверки сводимости и построения статической формы единственного присваивания могут быть решены с помощью:

Ответ:

- (1) непосредственного предшествования
- (2) отношения не обязательного предшествования
- (3) отношения обязательного предшествования
- (4) простого предшествования

Взаимно однозначное отображение множества вершин графа на отрезок натурального ряда $[1..|V|]$ называется:

Ответ:

- (1) нумерация
- (2) подграф
- (3) дерево
- (4) обратная дуга

Дерево, содержащее все вершины графа и некоторые его дуги:

Ответ:

- (1) глубинное остовное дерево
- (2) глубинное дерево
- (3) остовное дерево
- (4) простое дерево

Глубинное остовное дерево - это:

Ответ:

- (1) дерево, полученное при обходе в глубину
- (2) остовное дерево, полученное при обходе в глубину
- (3) простое дерево, полученное при обходе в глубину
- (4) сложное дерево, полученное при обходе в глубину

Существует следующие типы дуг графа по отношению к данному глубинному остовному дереву:

Ответ:

- (1) деревянные
- (2) прямые
- (3) обратные
- (4) поперечные

Необходимо различать:

Ответ:

- (1) просто обратные дуги в графе
- (2) обратные дуги по отношению к глубинному остовному дереву
- (3) сложные обратные дуги в графе
- (4) обратные дуги по отношению к глубинному дереву

Дуги, которые входят в состав остовного дерева - это:

Ответ:

- (1) деревянные
- (2) прямые
- (3) обратные
- (4) поперечные

Дуги, чей конец достижим из начала в остовном дереве называются:

Ответ:

- (1) поперечные
- (2) деревянные
- (3) обратные
- (4) прямые

Дуги, чье начало достижимо из конца в остовном дереве называются:

Ответ:

- (1) прямые
- (2) поперечные
- (3) обратные
- (4) деревянные

Состояние вершины, когда вершина еще не рассматривалась алгоритмом обозначается:

Ответ:

- (1) InProcess
- (2) Done
- (3) Init
- (4) Start

Состояние вершины, когда вершина еще рассматривается алгоритмом обозначается:

Ответ:

- (1) InProcess
- (2) Init
- (3) Start
- (4) Done

Для определения типа дуги используется:

Ответ:

- (1) состояние конечной вершины
- (2) нумерация Post
- (3) нумерация Pre
- (4) состояние начальной вершины

Каждая обратная дуга определяет в графе потока управления:

Ответ:

- (1) контур
- (2) фон
- (3) круг
- (4) вершину

Произвольный подграф графа управления называется:

Ответ:

- (1) элемент
- (2) контур
- (3) ядро
- (4) фрагмент

Альтом называется:

Ответ:

- (1) фрагмент, имеющий две начальных вершину
- (2) фрагмент, имеющий одну начальную вершину
- (3) фрагмент, не имеющий начальных вершин

Алгоритм выделения максимального альта, для которого данная вершина p является начальной, включает следующие шаги:

Ответ:

- (1) вначале все вершины графа помечаются как "черные"
- (2) вначале все вершины графа помечаются как "белые"
- (3) все вершины, достижимые из p , помечаются как "серые"
- (4) все вершины, достижимые из p , помечаются как "красные"

Свойства альтов дают возможность использовать их для определения:

Ответ:

- (1) отношения предшествования
- (2) отношения не обязательного предшествования
- (3) отношения простого предшествования
- (4) отношения обязательного предшествования

Луч - это:

Ответ:

- (1) линейная последовательность вершин
- (2) не линейная последовательность вершин
- (3) прямая последовательность вершин
- (4) обратная последовательность вершин

Множество максимальных лучей образует:

Ответ:

- (1) разбиение множества вершин контура
- (2) разметку множества вершин графа
- (3) разбиение множества вершин графа
- (4) разметку множества вершин контура

Сильно связный подграф - это:

Ответ:

- (1) фрагмент, состоящий из не достижимых вершин
- (2) фрагмент, состоящий из достижимых вершин
- (3) фрагмент, состоящий из взаимно не достижимых вершин
- (4) фрагмент, состоящий из взаимно достижимых вершин

Множества входных и начальных вершин для компонент сильной связности:

Ответ:

- (1) совпадают
- (2) не совпадают
- (3) эквивалентны
- (4) не эквивалентны

Для выделения сильно связных подграфов, достаточно научиться:

Ответ:

- (1) выделять фрагменты
- (2) выделять вершины
- (3) выделять дуги
- (4) выделять области

Компонента сильной связности, являющееся областью своей вершины, имеющей минимальный номер в нумерации Post среди всех остальных вершин этой компоненты называется:

Ответ:

- (1) моновершиной
- (2) минивершиной
- (3) максивершиной
- (4) бивершиной

Набор областей всех вершин при нумерации Post является:

Ответ:

- (1) иерархией вложенных фрагментов
- (2) иерархией вложенных зон
- (3) иерархией вложенных контуров
- (4) иерархией вложенных областей

Иерархия вложенных зон - это один из способов:

Ответ:

- (1) описать структуру программы

- (2) описать динамическую структуру программы
 (3) описать циклическую структуру программы
 (4) описать вложенную структуру программы

Стягивание линейных компонент переводит граф:

Ответ:

- (1) в луч
 (2) в дугу
 (3) в область
 (4) в контур

Признаком чего является то, что вершина со следующим номером - во-первых, является бивершиной, а во-вторых, ее номер - максимальный среди номеров всех потомков вершин текущей линейной компоненты:

Ответ:

- (1) старта линейной компоненты
 (2) продления линейной компоненты
 (3) завершения линейной компоненты
 (4) завершаемости линейной компоненты

5.4. Перечень видов оценочных средств

Задания со свободно конструируемым ответом (СКО) предполагает составление развернутого ответа на теоретический вопрос. Задание с выбором варианта ответа (ОВ, в задании данного типа предлагается несколько вариантов ответа, среди которых верный. Задания со свободно конструируемым ответом (СКО) предполагает составление развернутого ответа, включающего полное решение задачи с пояснениями.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	Партька Т. Л., Попов И.И.	Операционные системы, среды и оболочки: Учебное пособие	Москва: Издательство "ФОРУМ", 2021, URL: https://znanium.com/catalog/document?id=364475
Л1.2	Голицына О. Л., Партька Т. Л.	Языки программирования: Учебное пособие	Москва: Издательство "ФОРУМ", 2021, URL: http://znanium.com/catalog/document?id=367055
Л1.3	Миков А. И.	Обобщенные графы и грамматики: Учебное пособие	Москва: ООО "Научно-издательский центр ИНФРА-М", 2021, URL: https://znanium.com/catalog/document?id=377436

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	Малявко А.А.	Формальные языки и компиляторы: Учебник	Новосибирск: Новосибирский государственный технический университет (НГТУ), 2014, URL: http://znanium.com/catalog/document?id=74628
Л2.2	Алымова Е.В., Деундяк В.М.	Конечные автоматы и формальные языки: Учебник	Ростов-на-Дону: Издательство Южного федерального университета (ЮФУ), 2018, URL: http://znanium.com/catalog/document?id=339524

6.2. Электронные учебные издания и электронные образовательные ресурсы

Э1	Интернет университет информационных технологий ИНТУИТ . - Режим доступа: https://www.intuit.ru/studies/courses		
Э2	Электронная библиотечная система Znanium. - Режим доступа: http://znanium.com/		
Э3	Электронные ресурсы Академии ИМСИТ. - Режим доступа: http://eios.imsit.ru/		
Э4	Электронная библиотечная система Ibooks. - Режим доступа: http://www.ibooks.ru/		
Э5	Электронная библиотечная система BOOK.ru. - Режим доступа: http://www.book.ru		

6.3.1. Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства

6.3.1.1	7-Zip Архиватор 7-Zip Программное обеспечение по лицензии GNU GPL
6.3.1.2	Яндекс Браузер Браузер Яндекс Браузер Лицензионное соглашение на использование программ Яндекс Браузер https://yandex.ru/legal/browser_agreement/
6.3.1.3	Mozilla Firefox Браузер Mozilla Firefox Программное обеспечение по лицензии GNU GPL

6.3.1.4	LibreOffice Офисный пакет LibreOffice Программное обеспечение по лицензии GNU GPL
6.3.1.5	Oracle VM VirtualBox VM VirtualBox — программный продукт виртуализации для операционных систем Программное обеспечение по лицензии GNU GPL
6.3.1.6	Windows 10 Pro RUS Операционная система – Windows 10 Pro RUS Подписка Microsoft Imagine Premium – Order №143659 от 12.07.2021
6.3.1.7	Notepad++. Текстовый редактор Notepad++. Программное обеспечение по лицензии GNU GPL
6.3.1.8	MS Visio Pro 2016 Интегрированная среда разработки Microsoft Visio профессиональный 2016 Подписка Microsoft Imagine Premium – Order №143659 от 12.07.2021
6.3.1.9	MS Visual Studio Community Edition Среда разработки Microsoft Visual Studio 2022 Программное обеспечение по лицензии GNU GPL
6.3.2. Перечень профессиональных баз данных и информационных справочных систем	
6.3.2.1	ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION https://www.omg.org/spec/UML
6.3.2.2	Кодекс – Профессиональные справочные системы https://kodeks.ru
6.3.2.3	ИСО Международная организация по стандартизации https://www.iso.org/ru/home.html
6.3.2.4	РОССТАНДАРТ Федеральное агентство по техническому регулированию и метрологии https://www.gost.ru/portal/gost/
6.3.2.5	Консультант Плюс http://www.consultant.ru

7. МТО (оборудование и технические средства обучения)

Ауд	Наименование	ПО	Оснащение
114	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	LibreOffice Inkscape MS Visual Studio Community Edition Blender Gimp IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC MAC OS Big Sure JetBrains PyCharm Community JetBrains DataGrip	20 посадочных мест, рабочее место преподавателя, 15 моноблоков Apple iMac 21,1/Apple M1/RAM 8Гб/Apple SSD AP0256Q/GPU Apple M1/Ethernet 1000BaseT/AirPort Extreme 5 моноблоков Apple iMac 21,1/Apple M1/RAM 16Гб/Apple SSD AP0512Q/GPU Apple M1/Ethernet 1000BaseT/AirPort Extreme 1 сетевой неуправляемый коммутатор DES-1024G 1 Интерактивная панель EliteBoard LR-75UT40i7 1 Ноутбук 15.6 HP 15-ra105ur 1 МФУ Brother DCP-1612WR 1 HP Color LaserJet CP5225
114а	Кабинет информатики. Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox PostgreSQL IntelliJ IDEA PyCharm Community Edition	16 посадочных мест, рабочее место преподавателя 16 компьютеров GA-870A-USB3/AMD-Phenom(tm)-II-X4-945/DDR3-1333-4Гб/SSD Flexis 120Gb/WD5000AAKX/Radeon HD-5800/Realtek PCIe GBE 16 мониторов AOC e2243Fw 21,5” 16 комплектов клавиатура+мышь 1 Коммутатор LincSys SR224G 1 Проектор ViewSonic PJD5232 1 Проекторный экран Luma 1 Шкаф телекоммуникационный 1 ИБП SMART UPS 2000 3 Коммутатор Cisco Catalyst 2960 1 Концентратор AlterPath 16 port 4 Маршрутизатор Cisco-2800 2 Маршрутизатор Cisco-2811 6 Модуль 2-port 2 Панель коммутационная 12 Шнур V.35 Cable Витая пара, Коннектор RJ-45 2 Инструмент для зачистки кабеля UTP 1 Протяжка кабельная, d=3,5 мм 10 м 1 Тестер МЕГЕОН 40060/Шт. 5 Инструмент для обжима витой пары 5 Тестер кабельный 3 Инструмент для заделки кабеля витая пара тип Krone с крючками

		Eclips Adobe Reader DC Arduino Software (IDE) NetBeans IDE ZEAL Klite Mega Codec Pack MS Office Standart 2010 Ramus Educational Micro-Cap Evaluation	3 Р телефон GrandStream GXPI610 2 Комплект для монтажа СКС (патч-панель 1U kat.5e UTP 24 порта-1 шт., инструмент обжимной для RJ-45 1 шт., инструмент для зачистки кабеля 1 шт., инструмент для разделки контактов - 1 шт., LAN тестер 1 шт.) 2 Роутер Wi-Fi роутер Keenetic 1 Сервер GA-870A-USB3/AMD-Phenom(tm)-II-X4-945/ DDR3-1333-4Гб/SSD Flexis 120Gb/WD5000AAKX/Radeon HD-5800/Realtek PCIe GBE
115	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Arduino Software (IDE) NetBeans IDE ZEAL Klite Mega Codec Pack	20 посадочных мест, рабочее место преподавателя 20 компьютеров GA-870A-USB3/AMD-Phenom(tm)-II-X4-945/DDR3-1333-4Гб/ SSD Flexis 120Gb/WD5000AAK/Radeon HD-5800/Atheros AR9287 Wireless 19 мониторов AOC e2243Fw 21,5” 1 монитор Acer V226HQL 21,5” 20 комплектов клавиатура+мышь 1 беспроводная точка доступа TP-Link TL-WA801ND
118	Кафедра математики и вычислительной техники. Специальное помещение для хранения и профилактического обслуживания учебного оборудования	7-Zip Mozilla Firefox LibreOffice Kaspersky Endpoint Security Adobe Reader DC Klite Mega Codec Pack Java 8 PDF24 Creator Etxt Antiplagiat Microsoft Windows 10 PRO x64 DSP OEM MS Office Professional Plus 2007	Системный блок H310CM-DVS P 1.30\Intel(R) Pentium(R) Gold G5400 CPU 3.70GHz\DDR4-4Gb\SSD 240Gb Монитор Принтер HP LaserJet 1018 МФУ Brother DCP-L2540DNR
119	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019	20 посадочных мест, рабочее место преподавателя 20 компьютеров H110M-S2-C/INTEL Pentium G4400/DDR4-2133-4Гб/TOSHIBA HDWD105/Intel HD-510/Atheros AR9287 Wireless 20 мониторов 20 комплектов клавиатура+мышь 1 беспроводная точка доступа TP-Link TL-WA801ND

	контроля и промежуточной аттестации, самостоятельной работы.	MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL ARIS Express AnyLogic Klite Mega Codec Pack MS Office Standart 2007	
120	Кабинет информатики, технологий и методов программирования. Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC NetBeans IDE ZEAL SMath Studio Klite Mega Codec Pack	20 посадочных мест, рабочее место преподавателя 20 компьютеров A320M-H-CF/AMD Ryzen 5 2600X/DDR4-2933 16Гб/SSD XPG GAMMIX S11 Pro 512Гб/NVIDIA GeForce GTX 1050 Ti/Realtek PCIe GbE Family Controller 40 мониторов Samsung S24R350FHI 23.8" 20 ИБП CyberPower UT650EG 20 комплектов клавиатура+мышь 20 гарнитур Defenfer G-320 1 неуправляемый коммутатор TP-LINK TL-SG1024D 1 Интерактивная панель EliteBoard LR-75UT40i7
121	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security	17 посадочных мест, рабочее место преподавателя 17 компьютеров P8H67/INTEL i5-2300/DDR3-1333-4Гб/SSD Flexis 120Gb/ WD5000AAKX/Radeon HD 6700/Realtek PCIe GBE 17 мониторов AOC e2243Fw 21,5" 17 комплектов клавиатура+мышь 1 коммутатор неуправляемый DES-1024D

	индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC SMath Studio ПО ЛИНКО v8.2 демо-версия Klite Mega Codec Pack	
122	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL Комплекс КРЕДО - Землеустройство и кадастры Klite Mega Codec Pack MS Office Standart 2007	20 посадочных мест, рабочее место преподавателя 20 компьютеров H310M S2P/Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz/DDR4-2400-16Гб/TS240GMTS820S/ Radeon RX 550 Series/Realtek Gaming GbE Family Controlle 20 мониторов Acer G246HYL 24” 20 комплектов клавиатура+мышь 1 управляемый коммутатор TP-LINK TL-SG1024D 1 МФУ Brother DCP-1612WR
123	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++.	19 посадочных мест, рабочее место преподавателя 19 компьютеров H110M-S2-C/INTEL Pentium G4400/DDR4 -2133-4Гб/TOSHIBA HDWD105/Intel HD-510/Atheros AR9287 Wireless 19 мониторов LG Flatron 1718s 19 комплектов клавиатура+мышь 1 управляемый коммутатор TP-LINK TL-SG1024D

	проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Blender Gimp Maxima Oracle VM VirtualBox PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL Klite Mega Codec Pack	
125	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL SMath Studio Klite Mega Codec Pack	17 посадочных мест, рабочее место преподавателя 17 компьютеров H110M-S2-C/INTEL Pentium G4400/DDR4-2133-8Гб/TOSHIBA HDWD105/Intel HD-510/Atheros AR9287 Wireless 17 мониторов Samsung SyncMaster 920N 17 комплектов клавиатура+мышь 1 коммутатор неуправляемый DES-1024G
126	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов),	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект	20 посадочных мест, рабочее место преподавателя 11 компьютеров типа «Моноблок» Lenovo IdeaCentre-/ Intel Pentium CPU 4415U 2.30GHz/DDR4-2133-4Гб/ WDC WD10EZEX-08WN4A0 1000Гб/ Intel(R) HD Graphics 610 / Realtek PCIe GbE Family Controller/ Qualcomm Atheros QCA9377 Wireless Network Adapter 5 компьютеров типа «Моноблок» Lenovo IdeaCentre IAO 300-23SU /INTEL Pentium 4405U/DDR4-2400-8Гб/ST1000DM003/Intel HD-510/Intel(R) Dual Band Wireless

	групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы.	Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclipse Adobe Reader DC Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL Klite Mega Codec Pack	-AC 3165 4 Компьютера типа "Моноблок" Lenovo /Intel Pentium Silver J5040 CPU 2.00GHz/DDR4-2400 8Гб/SSD WDC PC SN530 SDBPMPZ-512G-1001/Intel(R) UHD Graphics 605/ Realtek PCIe GbE Family Controller/ Realtek 8821CE Wireless LAN 802.11ac PCI-E NIC 20 комплектов клавиатура+мышь 1 беспроводная точка доступа DWL-3200AP
208	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++. 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclipse Adobe Reader DC Diptrace Embarcadero RAD Studio XE8 Arduino Software (IDE) NetBeans IDE ZEAL Klite Mega Codec Pack MS Office Standart 2007 NI LabVIEW Full	19 посадочных мест, рабочее место преподавателя, 10 компьютеров H97-PLU/INTEL i5-4460/DDR3-1333-16Гб/SD7SB6S-128G+ST500DM002/Radeon R7 200/Realtek PCIe GBE 9 компьютеров A320M-H-CF/AMD Ryzen 5 2600/DDR4-2666-16Гб/Apacer AS2280P4-256Gb, Toshiba HDWD110 1Tb/Nvidia GT-710/Realtek PCI-E GBE 1 компьютер P8Z77-V-LX2/INTEL I5-3570K/DDR3-1600-8Гб/ SSD SSDPR-CX400-128G2, WDC WS15EARS/AMD HD-5700 Realtek PCIe GBE 10 мониторов Philips 274E5QSB 27" 1 монитор Samsung SyncMaster E1720 11 комплектов клавиатура+мышь 1 принтер HP LaserJet 1018 1 коммутатор неуправляемый TL-SG1024D Междисциплинарная лабораторная станция NI ELVIS II и ПО Circuit Design Suit Лаборатория схемотехники (необходимо наличие лаб. станции ELVIS) Практикум по цифровым элементам вычислительной и информационно-измерительной техники (необходимо наличие лабораторной станции ELVIS) Лаборатория проектирование цифровых устройств и программирования ПЛИС (необходимо наличие лабораторной станции ELVIS) Комплект аксессуаров NI myRIO Starter Accessory Kit (опционально) Комплект аксессуаров NI myRIO Mechatronics Accessory Kit Комплект аксессуаров NI myRIO Embedded Systems Accessory Kit Лаборатория программирования встраиваемых систем Локальные вычислительные сети (необходимо наличие лабораторной станции ELVIS) Промышленные интерфейсы и протоколы (программная версия) Академическая лицензия NI LabVIEW на неограниченное кол-во рабочих мест в пределах кафедры. Arduino Robot.
120	Лаборатория «Программная инженерия и разработка программного обеспечения».	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD	20 посадочных мест, рабочее место преподавателя 20 компьютеров A320M-H-CF/AMD Ryzen 5 2600X/DDR4-2933 16Гб/SSD XPG GAMMIX S11 Pro 512Гб/NVIDIA GeForce GTX 1050 Ti/Realtek PCIe GbE Family Controller 40 мониторов Samsung S24R350FHI 23.8" 20 ИБП CyberPower UT650EG

	<p>Полигон кибер-спорт». Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы. Кабинет информатики, технологий и методов программирования.</p>	<p>Inkscape Notepad++. I:\Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016 MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox StarUML V1 PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC NetBeans IDE ZEAL SMath Studio Klite Mega Codec Pack 10-Strike File search pro УМКК "Объектно-ориентированные технологии» УМКК "Основы алгоритмизации и программирования»</p>	<p>20 комплектов клавиатура+мышь 20 гарнитур Defenfer G-320 1 неуправляемый коммутатор TP-LINK TL-SG1024D 1 Интерактивная панель EliteBoard LR-75UT40i7</p>
124	<p>Кластерная лаборатория Серверный центр Специальное помещение для хранения и профилактического обслуживания учебного оборудования</p>	<p>7-Zip Mozilla Firefox PostgreSQL Java 8 Kaspersky Endpoint Security 11 Windows Server 2003 R2 Standart Open SuSe Linux MySQL Server Community Windows Server 2016 Standard Сервер администрирования Kaspersky Security Center УМКК «Телекоммуникации и сети» УМКК «Коммутаторы локальных сетей» УМКК «Электротехника и электроника» УМКК «Информационные системы в экономике» УМКК «Корпоративные информационные системы» УМКК "Моделирование данных» УМКК "Объектно-ориентированные технологии» УМКК «Информационные технологии» УМКК «Управление базами данных» УМКК «Сетевые информационные технологии» УМКК «Теоретические основы информатики» УМКК "Основы алгоритмизации и программирования»</p>	<p>Стойка серверная Управляющий узел кластера I500PX-S5380\ Xeon E5345\ DDR-2-667-8192Mb\WD5001ABYS 1 шт. Рабочий узел кластера I500PX-S5380\ Xeon E5345\ DDR-2-667-8192Mb\WD800JD\ - 16 шт Серверный узел Spectrus I500PX-S5380\ Xeon E5345\ DDR-2-667-8192Mb Серверный узел DEXUS II I500PX-S5380\ Xeon E5345\ DDR-2-667-8192Mb\ Коммутатор DLink Коммутатор DLink Серверный узел SuperMicro 1U6019PMT\Xeon silver 4108 \8xDDR4 8Gd\ - 2 шт Сетевое хранилище данных Synology DS-418 1 шт. Монитор Acer V193 1 шт. Шкаф 2-х дверный архивный металл. - 2шт Сплит система AirWell 1 шт. Сплит-система Lessar 1 шт. Система контроля доступа СКАТ 1200 И7 1 шт</p>

		JetBrains License Service Autodesk Network License Manager AppWave Enterprise License Center Windows Server 2008 R2 Standart Traffic inspector Special Unlimited Эшэлон II “Кредо-диалог” Система управления хранилищем документов “Кредо-диалог” Центр управления ПО Кредо MS SQL Server 2016 Apache HTTP Server	
118	Кафедра математики и вычислительной техники. Специальное помещение для хранения и профилактического обслуживания учебного оборудования	7-Zip Mozilla Firefox LibreOffice Kaspersky Endpoint Security Adobe Reader DC Klite Mega Codec Pack Java 8 PDF24 Creator Etxt Antiplagiat Microsoft Windows 10 PRO x64 DSP OEM MS Office Professional Plus 2007	Системный блок H310CM-DVS P 1.30\Intel(R) Pentium(R) Gold G5400 CPU 3.70GHz\DDR4-4Gb\SSD 240Gb Монитор Принтер HP LaserJet 1018 МФУ Brother DCP-L2540DNR
123а	Специальное помещение для хранения и профилактического обслуживания учебного оборудования	7-Zip Яндекс Браузер LibreOffice Notepad++ Oracle VM VirtualBox Adobe Reader DC ZEAL Klite Mega Codec Pack Windows 7 Pro CDBurnerXP Java 8 PDF24 Creator CCleaner Консоль Kaspersky Security Center Kaspersky Endpoint Security 11 ПАРУС-Бюджет 8.5.6.1 Microsoft Office 2007 Professional Plus 10-Strike File search pro 10-Страйк Сканирование Сети 10-Страйк Инвентаризация Компьютеров	Системный блок AMD FX-8120 1шт Системный блок Intel Core 2 CPU 4400 1шт. Монитор “LG L1718S” 1 шт. Монитор “BENQ CL2240” 1шт. Монитор “SAMSUNG 740m” 1шт. Набор инструментов 1 шт. Паяльная станция Lukey 902 1 шт Принтер SAMSUNG ML-1665 1 шт. Принтер SAMSUNG ML-1615 1 шт. Коммутатор D-Link DES-1005D 1 шт. Роутер Keenetic Lite (KN-3110)1 шт. Паяльник 40 Вт дер/ручка 1 шт. Лампа настольная 1 шт. Стол 1-тумбовый 1 шт. Стол 2 тумбовый 1 шт. Стол офисный компьютерный 1 шт. Столик компьютерный 1 шт. Стол 1-тубовый с верхней приставкой 1шт. Стулья тканевые на металлокаркасе 2шт Стул деревянный 1шт Пылесос “SUPRA 1800W” 1 шт. Шуруповерт “Hitachi ds12dvf3” 1 шт. Веб-камера Logitech HD WebCam C525 1280*720 MicUSB - 4 шт Перфоратор Град-М 1 шт. Микрофон Yanmai R933 – 2 шт Ноутбук Asus X541U – 1 шт Проектор Cactus CS-PRO.02B.WXGA-W – 1 шт. Проектор Acer QNX1310 – 2 шт
113	Помещение для проведения занятий лекционного типа, семинарского типа, курсовых работ (курсовых проектов), групповых и индивидуальных консультаций,	Windows 10 Pro RUS 7-Zip Яндекс Браузер Mozilla Firefox LibreOffice LibreCAD Inkscape Notepad++ 1С:Предприятие 8. Комплект Kaspersky Endpoint Security MS Access 2016 MS Project Pro 2016	20 посадочных мест, рабочее место преподавателя 20 компьютеров P55-UD3/INTEL-i5-750/DDR3-1333-8Гб/SSD Flexis 120Gb /WD3200AAKS/Radeon HD-4600/DWL-G520 Wireles 20 мониторов Acer V193W-19” 20 комплектов клавиатура+мышь 1 коммутатор неуправляемый DES-1024D 1 беспроводная точка доступа DWL-3200AP 3 Комплект оборудования Arduino 5 учебных комплектов SDK 1.1s 1 МФУ HP LJ M1212nf MFP 12 Инструмент для сборки ПК (отвертка ph-1, плоскогубцы

текущего контроля и промежуточной аттестации, самостоятельной работы.	MS SQL Server 2019 MS SQL Server Management Studio 18.8 MS Visio Pro 2016 MS Visual Studio Community Edition Visual Studio Code Blender Gimp Maxima Oracle VM VirtualBox PostgreSQL IntelliJ IDEA PyCharm Community Edition Eclips Adobe Reader DC Diptrace Ramus Educational Micro-Cap Evaluation	150 мм, термопаста 2гр., Антистатический браслет, стяжки 150 мм)
---	--	--

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

В соответствии с требованиями ФГОС ВО по направлению подготовки реализация компетентностного подхода предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных задач, проведение блиц-опросов, исследовательские работы) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Лекционные занятия дополняются ПЗ и различными формами СРС с учебной и научной литературой. В процессе такой работы студенты приобретают навыки «глубокого чтения» - анализа и интерпретации текстов по методологии и методике дисциплины.

Учебный материал по дисциплине «Конструирование компиляторов» разделен на логически завершенные части (модули), после изучения, которых предусматривается аттестация в форме письменных тестов, контрольных работ.

Работы оцениваются в баллах, сумма которых дает рейтинг каждого обучающегося. В баллах оцениваются не только знания и навыки обучающихся, но и их творческие возможности: активность, неординарность решений поставленных проблем. Каждый модуль учебной дисциплины включает обязательные виды работ – лекции, ПЗ, различные виды СРС (выполнение домашних заданий по решению задач, подготовка к лекциям и практическим занятиям).

Форма текущего контроля знаний – работа студента на практическом занятии, опрос. Форма промежуточных аттестаций – контрольная работа в аудитории, домашняя работа. Итоговая форма контроля знаний по модулям – контрольная работа с задачами по материалу модуля.

Методические указания по выполнению всех видов учебной работы размещены в электронной образовательной среде академии.

Методические указания и материалы по видам учебных занятий по дисциплине:

Вид учебных занятий, работ - Организация деятельности обучающегося

Лекция - Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения, отмечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, попытаться найти ответ в рекомендуемой литературе, если самостоятельно не удаётся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.

Практические занятия - Конспектирование источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы, работа с текстом. Выполнение практических задач в инструментальных средах. Выполнение проектов. Решение расчётно-графических заданий, решение задач по алгоритму и др.

Самостоятельная работа - Знакомство с основной и дополнительной литературой, включая справочные издания, зарубежные источники, конспект основных положений, терминов, сведений, требующихся для запоминания и являющихся основополагающими в этой теме. Составление аннотаций к прочитанным литературным источникам и др.

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа студентов в ходе семестра является важной составной частью учебного процесса и необходима для закрепления и углубления знаний, полученных в период сессии на лекциях, практических и интерактивных занятиях, а также для индивидуального изучения дисциплины «Объектно-ориентированное программирование» в соответствии с программой и рекомендованной литературой.

Самостоятельная работа выполняется в виде подготовки домашнего задания или сообщения по отдельным вопросам, написания и защита научно-исследовательского проекта.

Контроль качества выполнения самостоятельной (домашней) работы может осуществляться с помощью устного опроса на лекциях или практических занятиях, обсуждения подготовленных научно-исследовательских проектов, проведения тестирования.

Устные формы контроля помогут оценить владение студентами жанрами научной речи (дискуссия, диспут, сообщение,

доклад и др.), в которых раскрывается умение студентов передать нужную информацию, грамотно использовать языковые средства, а также ораторские приемы для контакта с аудиторией.

Письменные работы позволяют оценить владение источниками, научным стилем изложения, для которого характерны: логичность, точность терминологии, обобщенность и отвлеченность, насыщенность фактической информацией.

Выбор варианта задания может быть сделан из предложенного ниже списка:

разработать программу для конечного автомата (КА) в соответствии с вариантом работы (вариант выбирается по последней цифре зачёта).

Вариант 0. Арифметическое простое арифметическое выражение. Операндами выражения являются вещественные и целые числа, а операциями – сложение, умножение, деление и вычитание.

Вариант 1. Прямая польская запись арифметических выражений. Операнды – вещественные и целые числа, операции – сложение, умножение, деление и вычитание.

Вариант 2. Обратная польская запись арифметических выражений. Операнды – вещественные и целые числа, операции – сложение, умножение, деление и вычитание.

Вариант 3. Квадратное уравнение с целыми и вещественными коэффициентами.

Вариант 4. Линейное алгебраическое уравнение.

Вариант 5. Сумма обыкновенных дробей.

Вариант 6. Комплексное число (с целочисленными значениями действительной и мнимой частей).

Вариант 7. Линейное однородное дифференциальное уравнение с постоянными целочисленными коэффициентами.

Вариант 8. Неравенство вида:

Операнд1 \square Операнд2,

где Операнд1 и Операнд2 – целые числа;

\square – знак отношения ($>$, $<$, $=$, $<>$, $>=$, $<=$).

Вариант 9. Двуместный логический предикат. Операнды – числа 1 (истина) или 0 (ложь). Операции – AND (логическое умножение) и OR (логическое сложение).

Разработать синтаксическую диаграмму и, с её помощью, реализовать программно КА

Вариант 0. Сумма – последовательность натуральных чисел и имен, разделенных знаками плюс и минус. Возможен и знак перед первым слагаемым.

Вариант 1. Сумма вещественных чисел в форме с фиксированной точкой.

Вариант 2. Произведение вещественных чисел с фиксированной точкой.

Вариант 3. Сумма вещественных чисел с плавающей точкой.

Вариант 4. Произведение вещественных чисел с плавающей точкой.

Вариант 5. Квадратное уравнение с целыми коэффициентами.

Вариант 6. Сумма обыкновенных дробей.

Вариант 7. Произведение обыкновенных дробей.

Вариант 8. Разность обыкновенных дробей.

Вариант 9. Комплексное число (с целочисленными значениями действительной и мнимой частей).