

Министерство образования и науки Российской Федерации
Федеральное государственное учреждение высшего образования
Московский государственный университет технологий и управления
(первый казачий университет)
Университетский колледж информационных технологий

Специальность 09.02.03 Программирование в компьютерных системах
ПМ.03 Участие в интеграции программных модулей
МДК.03.02 Инструментальные средства разработки программного обеспечения
Сборник заданий для практических работ

Содержание

1	Аннотация	3
2	Практическая работа №1. Работа с опциями транслятора(2 часа)	3
3	Практическая работа №2. Разработка в среде Linux (12 часов)	6
4	Практическая работа №3. Создание библиотек (4 часа)	9
5	Практическая работа №4. Тестирование (6 часов)	9
6	Практическая работа №5. Компоновка программ, написанных на различных языках программирования (2 часа)	12
7	Практическая работа №6. Автоматизация тестирования GUI (2 часа)	14
8	Практическая работа №7. Средства управления требованиями (2 часа)	14
9	Практическая работа №8. Средство создания документации (2 часа)	14
10	Практическая работа №9 Инструментальные средства проектирования (4 часа)	15
11	Практическая работа №10. Средства автоматизированного программирования (4 часа)	16
12	Практическая работа №11. Средства обратной разработки (2 часа)	18
A	Список используемого программного обеспечения	19
B	Список источников	19

1 Аннотация

Сборник включает практические задания для практических работ по междисциплинарному курсу «МДК03.02 Инструментальные средства разработки программного обеспечения», суммарное время их выполнения – 42 аудиторных часа. Целью практических работ является дальнейшее развитие общих и профессиональных компетенций, усвоение дидактических единиц, соответствующих МДК.

Перечислим общие компетенции, развиваемые практическими работами:

- ОК1 понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес;
- ОК2 организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество;
- ОК3 принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность;
- ОК4 осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;
- ОК5 использовать информационно-коммуникационные технологии в профессиональной деятельности;
- ОК6 работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями;
- ОК7 брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий;
- ОК9 ориентироваться в условиях частой смены технологий в профессиональной деятельности.

При работе осуществляется формирование и дальнейшее развитие следующих профессиональных компетенций:

- ПК3.3 выполнять отладку программного продукта с использованием специализированных программных средств;
- ПК3.4 осуществлять разработку тестовых наборов и тестовых сценариев;
- ПК3.5 производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования;
- ПК3.6 разрабатывать технологическую документацию.

В ходе работы учащийся получает практический опыт участия в выработке требований к программному обеспечению, участия в проектировании программного обеспечения с использованием специализированных программных пакетов, использует методы для получения кода с заданной степенью качества, получает практическое закрепление знаний о модели процесса разработки программного обеспечения, основных принципах процесса разработки программного обеспечения, основных методах и средствах эффективной разработки, основах верификации и аттестации программного обеспечения, реализации программных процессов, принципах построения, структуре и приемах работы с инструментальными средствами, поддерживающими создание программного обеспечения, методах организации работы в коллективах разработчиков программного обеспечения, о стандартах качества программного обеспечения, методах и средствах разработки программной документации.

Редакция 2015 года отличается небольшим изменением содержания практических работ и корректировкой формулировок.

Редакция 2016 года отличается учетом изменения количества времени, отводимого на дисциплину; увеличением времени на некоторые практические работы; удалением практических работ, которые дублируются в других предметах; включением новых изучаемых инструментальных средств. Кроме того, во всех работах количество индивидуальных заданий доведено до 70 так, чтобы задания не повторялись для всех студентов одной специальности.

2 Практическая работа №1. Работа с опциями транслятора(2 часа)

1. Цель работы: практическое закрепление знаний об опциях транслятора.
2. Порядок выполнения:
 - (а) напишите программу в соответствии с вашим вариантом (вычисление суммы ряда до тех пор, пока слагаемое не станет меньше ϵ ; ϵ задается в виде выбираемой вами константы; с целью более точного измерения времени работы программы само вычисление суммы ряда заиклите 1000000 раз);
 - (б) осуществите компиляцию с использованием опций `-O0`, `-O1`, `-O2`, `-O3`, `-Os`, `-Ofast`. Кроме того, всегда указывайте опцию `-s`.
 - (в) оформите таблицу, каждая строка которой соответствует опции, а столбцы соответствуют времени работы программы (измеряется при помощи утилиты `time`; размер исполняемого файла).

Варианты заданий:

В заданиях предполагается, что f_i – это числа Фибоначчи (при этом $f_1 = f_2 = 1$). Программу необходимо написать в наиболее оптимальном по времени выполнения ключе (для вычисления собственно суммы ряда достаточно одного цикла; рекурсия при выполнении заданий не нужна). В этой и последующих работах вариант выбирается следующим образом: первая группа использует варианты 1-35; вторая группа – 36-70. Порядковый номер в остальном определяется номером в журнале студента.

- 1 $\frac{f_1 \cdot 2^1}{1!} + \frac{f_2 \cdot 2^2}{2!} + \frac{f_3 \cdot 2^3}{3!} + \dots$
- 2 $\frac{f_2 \cdot 3^1 \sqrt{2}}{1!} + \frac{f_3 \cdot 3^3 \sqrt{3}}{2!} + \frac{f_4 \cdot 3^5 \sqrt{4}}{3!} + \dots$
- 3 $\frac{f_2 \cdot 3^1}{1!} + \frac{f_3 \cdot 3^2}{2!} + \frac{f_4 \cdot 3^3}{3!} + \dots$
- 4 $\frac{f_1 \cdot 4^1 \sqrt{2}}{1!} + \frac{f_2 \cdot 4^3 \sqrt{3}}{2!} + \frac{f_3 \cdot 4^5 \sqrt{4}}{3!} + \dots$
- 5 $\frac{f_3 \cdot 4^1}{2!} + \frac{f_4 \cdot 4^2}{3!} + \frac{f_5 \cdot 4^3}{4!} + \dots$
- 6 $\frac{f_1 \cdot 4^2 \sqrt{2}}{2!} + \frac{f_2 \cdot 4^3 \sqrt{3}}{3!} + \frac{f_3 \cdot 4^4 \sqrt{4}}{4!} + \dots$
- 7 $\frac{f_1 \cdot 3^1}{1!} + \frac{f_2 \cdot 3^2}{2!} + \frac{f_3 \cdot 3^3}{3!} + \dots$
- 8 $\frac{f_3 \cdot 4^2 \sqrt{2}}{1!} + \frac{f_4 \cdot 4^3 \sqrt{3}}{2!} + \frac{f_5 \cdot 4^4 \sqrt{4}}{3!} + \dots$
- 9 $\frac{f_1 \cdot 6^2 \sqrt{2}}{1!} + \frac{f_2 \cdot 6^3 \sqrt{3}}{2!} + \frac{f_3 \cdot 6^4 \sqrt{4}}{3!} + \dots$
- 10 $\frac{f_1 \cdot 7^1}{2!} - \frac{f_2 \cdot 7^2}{3!} + \frac{f_3 \cdot 7^3}{4!} - \frac{f_4 \cdot 7^4}{5!} + \dots$
- 11 $\frac{f_2 \cdot 6^2 \cdot 2^3}{1!} + \frac{f_3 \cdot 6^3 \cdot 3^3}{2!} + \frac{f_4 \cdot 6^4 \cdot 4^3}{3!} + \dots$
- 12 $\frac{f_1}{4^2 \cdot 1!} - \frac{f_2}{4^3 \cdot 2!} + \frac{f_3}{4^4 \cdot 3!} - \dots$
- 13 $\frac{f_1}{6^1 \cdot 1!} + \frac{f_2}{6^2 \cdot 2!} + \frac{f_3}{6^3 \cdot 3!} + \dots$
- 14 $\frac{f_1}{5^2 \cdot 1!} + \frac{f_2}{5^3 \cdot 2!} + \frac{f_3}{5^4 \cdot 3!} + \dots$
- 15 $\frac{f_2 \cdot e^2}{1!} + \frac{f_3 \cdot e^3}{2!} + \frac{f_4 \cdot e^4}{3!} + \frac{f_5 \cdot e^5}{3!} + \dots$
- 16 $\frac{f_2 \cdot 2^2}{2!} + \frac{f_3 \cdot 2^3}{3!} + \frac{f_4 \cdot 2^4}{4!} + \dots$
- 17 $\frac{f_3 \cdot 4^1 \sqrt{3}}{2!} + \frac{f_4 \cdot 4^3 \sqrt{4}}{3!} + \frac{f_5 \cdot 4^5 \sqrt{5}}{4!} + \dots$
- 18 $\frac{f_3 \cdot 3^2}{2!} + \frac{f_4 \cdot 3^3}{3!} + \frac{f_5 \cdot 3^4}{4!} + \dots$
- 19 $\frac{f_2 \cdot 4^2 \sqrt{3}}{2!} + \frac{f_3 \cdot 4^4 \sqrt{4}}{3!} + \frac{f_4 \cdot 4^6 \sqrt{5}}{4!} + \dots$
- 20 $\frac{f_4 \cdot 4^2}{3!} + \frac{f_5 \cdot 4^3}{4!} + \frac{f_6 \cdot 4^4}{5!} + \dots$
- 21 $\frac{f_2 \cdot 4^3 \sqrt{3}}{3!} + \frac{f_3 \cdot 4^4 \sqrt{4}}{4!} + \frac{f_4 \cdot 4^5 \sqrt{5}}{5!} + \dots$
- 22 $\frac{f_2 \cdot 3^2}{2!} + \frac{f_3 \cdot 3^3}{3!} + \frac{f_4 \cdot 3^4}{4!} + \dots$
- 23 $\frac{f_4 \cdot 4^3 \sqrt{3}}{2!} + \frac{f_5 \cdot 4^4 \sqrt{4}}{3!} + \frac{f_6 \cdot 4^5 \sqrt{5}}{4!} + \dots$
- 24 $\frac{f_2 \cdot 6^3 \sqrt{3}}{2!} + \frac{f_3 \cdot 6^4 \sqrt{4}}{3!} + \frac{f_4 \cdot 6^5 \sqrt{5}}{4!} + \dots$
- 25 $\frac{f_2 \cdot 7^2}{3!} - \frac{f_3 \cdot 7^3}{4!} + \frac{f_4 \cdot 7^4}{5!} - \frac{f_5 \cdot 7^5}{6!} + \dots$
- 26 $\frac{f_3 \cdot 6^3 \cdot 2^4}{2!} + \frac{f_4 \cdot 6^4 \cdot 3^4}{3!} + \frac{f_5 \cdot 6^5 \cdot 4^4}{4!} + \dots$
- 27 $\frac{f_2}{4^3 \cdot 2!} - \frac{f_3}{4^4 \cdot 3!} + \frac{f_4}{4^5 \cdot 4!} - \dots$
- 28 $\frac{f_2}{6^2 \cdot 2!} + \frac{f_3}{6^3 \cdot 3!} + \frac{f_4}{6^4 \cdot 4!} + \dots$
- 29 $\frac{f_2}{5^3 \cdot 2!} + \frac{f_3}{5^4 \cdot 3!} + \frac{f_4}{5^5 \cdot 4!} + \dots$
- 30 $\frac{f_3 \cdot e^3}{2!} + \frac{f_4 \cdot e^4}{3!} + \frac{f_5 \cdot e^5}{4!} + \frac{f_6 \cdot e^6}{5!} + \dots$
- 31 $\frac{f_2 \cdot 4^2 2^2}{2!} + \frac{f_3 \cdot 4^3 3^2}{3!} + \frac{f_4 \cdot 4^4 4^2}{4!} + \dots$
- 32 $\frac{f_1 \cdot 3^1 \sqrt{2}}{1!} + \frac{f_3 \cdot 3^3 \sqrt{4}}{3!} + \frac{f_5 \cdot 3^5 \sqrt{6}}{5!} + \dots$
- 33 $\frac{1}{2!} + \frac{1}{4!} + \frac{1}{6!} + \dots$
- 34 $\frac{f_1}{1!} + \frac{f_2}{3!} + \frac{f_3}{5!} + \dots$
- 35 $\frac{f_1 \cdot 3^1 \sqrt{2}}{1!} + \frac{f_3 \cdot 3^3 \sqrt{4}}{3!} + \frac{f_5 \cdot 3^5 \sqrt{6}}{5!} + \dots$
- 36 $\frac{1}{1!} + \frac{1}{3!} + \frac{1}{5!} + \dots$
- 37 $\frac{f_2}{1!} + \frac{f_3}{3!} + \frac{f_4}{5!} + \dots$

$$38 \frac{f_1 \cdot 2^1 \ln 2}{1!} + \frac{f_3 \cdot 2^3 \ln 4}{3!} + \frac{f_5 \cdot 2^5 \ln 6}{5!} + \dots$$

$$39 \frac{1}{1!} + \frac{1}{4!} + \frac{1}{7!} + \dots$$

$$40 \frac{f_1}{2!} + \frac{f_2}{4!} + \frac{f_3}{6!} + \dots$$

$$41 \frac{f_2 \cdot 2^1 \ln 2}{2!} + \frac{f_3 \cdot 2^3 \ln 4}{4!} + \frac{f_4 \cdot 2^5 \ln 6}{6!} + \dots$$

$$42 \frac{1}{2!} + \frac{1}{5!} + \frac{1}{8!} + \dots$$

$$43 \frac{f_2}{1!} + \frac{f_4}{3!} + \frac{f_6}{5!} + \dots$$

$$44 \frac{f_1 \cdot 3^1 \ln 2}{1!} + \frac{f_3 \cdot 3^3 \ln 4}{3!} + \frac{f_5 \cdot 3^5 \ln 6}{5!} + \dots$$

$$45 \frac{1}{2!!} + \frac{1}{4!!} + \frac{1}{6!!} + \dots$$

$$46 \frac{f_4}{1!} + \frac{f_5}{2!} + \frac{f_6}{3!} + \dots$$

$$47 \frac{f_1 \cdot 4^1}{2!} + \frac{f_1 \cdot 4^2}{3!} + \frac{f_2 \cdot 4^3}{4!} + \frac{f_2 \cdot 4^4}{5!} + \dots$$

$$48 \frac{f_1 \cdot 2^1 \ln 2}{1!} + \frac{f_3 \cdot 2^3 \ln 4}{4!} + \frac{f_5 \cdot 2^5 \ln 6}{7!} + \dots$$

$$49 \frac{1}{1!!} + \frac{1}{3!!} + \frac{1}{5!!} + \dots$$

$$50 \frac{f_3}{1!} + \frac{f_5}{2!} + \frac{f_7}{3!} + \dots$$

$$51 \frac{f_2 \cdot 4^1}{2!} + \frac{f_2 \cdot 4^2}{3!} + \frac{f_3 \cdot 4^3}{4!} + \frac{f_3 \cdot 4^4}{5!} + \dots$$

$$52 \frac{f_1 \cdot 2^1 \ln 2}{1!} - \frac{f_3 \cdot 2^3 \ln 4}{4!} + \frac{f_5 \cdot 2^5 \ln 6}{7!} - \dots$$

$$53 \frac{1}{1!!} + \frac{1}{2!!} + \frac{1}{3!!} + \dots$$

$$54 \frac{f_2}{1!} + \frac{f_5}{2!} + \frac{f_8}{3!} + \dots$$

$$55 \frac{f_1 \cdot 2^1 \ln 2}{1!!} - \frac{f_3 \cdot 2^3 \ln 4}{3!!} + \frac{f_5 \cdot 2^5 \ln 6}{5!!} - \dots$$

$$56 \frac{1}{2!!} + \frac{1}{3!!} + \frac{1}{4!!} + \dots$$

$$57 \frac{f_4}{1!} + \frac{f_6}{2!} + \frac{f_8}{3!} + \dots$$

$$58 \frac{f_2 \cdot \sin 2}{2!} + \frac{f_2 \cdot \sin 3}{3!} + \frac{f_3 \cdot \sin 4}{4!} + \frac{f_3 \cdot \sin 5}{5!} + \dots$$

$$59 \frac{f_1 \cdot 2^1 \ln 2}{1!!} - \frac{f_3 \cdot 2^3 \ln 4}{3!!} + \frac{f_5 \cdot 2^5 \ln 6}{5!!} - \dots$$

$$60 \frac{1}{2!!} + \frac{1}{4!!} + \frac{1}{6!!} + \dots$$

$$61 \frac{f_4}{1!} + \frac{f_6}{3!} + \frac{f_8}{5!} + \dots$$

$$62 \frac{f_2 \cdot \cos 2}{1!} + \frac{f_2 \cdot \cos 3}{2!} + \frac{f_3 \cdot \cos 4}{3!} + \frac{f_3 \cdot \cos 5}{4!} + \dots$$

$$63 \frac{f_1 \cdot 5^1 \ln 2}{2!!} - \frac{f_3 \cdot 5^3 \ln 4}{4!!} + \frac{f_5 \cdot 5^5 \ln 6}{6!!} - \dots$$

$$64 \frac{1}{3!!} + \frac{1}{5!!} + \frac{1}{7!!} + \dots$$

$$65 \frac{f_1}{1!} - \frac{f_3}{3!} + \frac{f_5}{5!} + \dots$$

$$66 \frac{f_1 \cdot \cos 2}{1!} + \frac{f_1 \cdot \cos 3}{3!} + \frac{f_3 \cdot \cos 4}{5!} + \frac{f_3 \cdot \cos 5}{7!} + \dots$$

$$67 \frac{f_2}{5^1 \cdot 1!} + \frac{f_4}{5^3 \cdot 2!} + \frac{f_6}{5^5 \cdot 3!} + \dots$$

$$68 \frac{f_1 \cdot 5^1}{\ln 2 \cdot 2!!} - \frac{f_3 \cdot 5^3}{\ln 4 \cdot 4!!} + \frac{f_5 \cdot 5^5}{\ln 6 \cdot 6!!} - \dots$$

$$69 \frac{1}{3!!} + \frac{1}{4!!} + \frac{1}{5!!} + \dots$$

$$70 \frac{f_1}{2!} + \frac{f_3}{4!} + \frac{f_5}{6!} + \dots$$

3. Содержание отчета:

- (a) цель работы;
- (b) исходный текст программы;
- (c) таблица, описанная в задании;
- (d) ответ на контрольные вопросы.

4. Контрольные вопросы:

- (a) Опишите опции, использованные в задании
- (b) Приведите объяснение тех данных, что были получены в вашей работе

3 Практическая работа №2. Разработка в среде Linux (12 часов)

1. Цель работы: практическое закрепление знаний об инструментальных средствах, используемых при разработке программ в среде Linux
2. Порядок выполнения:
 - (a) в процессе разработки используйте средство контроля версий git, средство отслеживания ошибок (bitbucket.org);
 - (b) напишите программу в соответствии с вашим вариантом, разбив ее на различные исходные файлы (не менее трех исходных файлов);
 - (c) осуществите компиляцию с использованием транслятора и редактора связей gcc, а также программы автоматизации сборки make (Makefile необходимо написать грамотно, чтобы перекомпиляция исходных файлов осуществлялась только в той части, что изменилась)
 - (d) осуществите статический анализ кода с использованием средства cppcheck (используйте опцию `--enable=all`). Исправьте найденные недочеты в вашей программе или подготовьте объяснение того, почему исправление некоторых недочетов не требуется. Включите в отчет скриншоты результатов статического анализа до и после исправлений. Если изначально не было найдено замечаний, то второй скриншот не требуется.
 - (e) осуществите UNIT-тестирование нескольких подпрограмм с помощью [check]<https://libcheck.github.io/check/> или аналогичного средства. Тестовый набор должен содержать минимум 10 отдельных тестов для минимум трех отдельных подпрограмм. Для запуска UNIT-теста создайте соответствующую цель в Makefile.
 - (f) осуществите подготовку профиля работы вашей программы (время выполнения и частота вызова каждой подпрограммы) с использованием valgrind и какого-либо средства для демонстрации результатов его работы (например, kcachegrind).
 - (g) осуществите подготовку распространяемого пакета с использованием autotools (см. <http://blablablancode.ru/linux/500>). В результате у вас должны работать все обычные цели make, включая dist.
 - (h) подготовьте документацию вашего продукта с использованием wiki, встроенном в bitbucket.

Напишите файловую базу данных, управляемую в консольном интерактивном режиме (квазиоконный интерфейс делать не нужно) с возможностями: добавление, удаление, изменение информации соответственно вашему варианту (на языке C). Необходимо осуществить проверку всех возможных ошибок в части некорректных действий пользователя и при работе с файлами.

- 1 База данных студентов группы. Поля: фамилия, имя, отчество, пол, возраст.
- 2 База данных расходов семьи. Поля: товар, стоимость, количество, дата.
- 3 База данных загрузки аудиторий. Поля: дата и время начала, дата и время конца, аудитория, преподаватель.
- 4 База данных учета доходов и расходов предпринимателя. Поля: дата, тип операции (доход/расход), объем операции, описание, корреспондент.
- 5 База данных велоклуба. Поля: ФИО, тип велосипеда (МТВ и др.), стаж участия в велоклубе.
- 6 База данных рейсов авиакомпании. Поля: дата и время вылета, аэропорт вылета, аэропорт прилета, дата и время прилета, марка самолета.
- 7 База данных автобусных маршрутов. Поля: номер маршрута, номер парка, времена начала и окончания движения, длина маршрута в км.
- 8 База данных электричек. Поля: вокзал, номер поезда, количество вагонов, тип (экспресс/обычный/спутник), станция назначения.
- 9 База данных товаров Интернет-магазина. Поля: название товара, категория, цена товара, описание товара.
- 10 База заказов Интернет-магазина. Поля: ФИО заказчика, стоимость заказа, скидка (в процентах), адрес доставки.
- 11 База данных выборов. Поля: участок, кандидат, количество голосов.
- 12 База данных практических работ. Поля: практическая работа, студент, номер варианта, номер уровня, дата сдачи, оценка.
- 13 База данных операторов и телеканалов. Поля: Название, тип (спутник, кабель, Интернет), охват (кол-во миллионов домохозяйств), минимальная стоимость подписки.
- 14 База данных тарифных планов оператора. Поля: название, тип вещания (обычный/HD), флаг общедоступности.
- 15 База данных незаконно огороженных берегов. Поля: водный объект, регион, GPS-координаты, длина недоступного участка берега, дата фиксации нарушения.
- 16 База данных временного прекращения движения в метро. Поля: дата и время начала прекращения движения, дата и время окончания прекращения движения, станция, станция (от какой до какой станции прекращено движение).
- 17 База данных проката фильмов. Поля: дата, время, кинотеатр, фильм, номер зала, тип сеанса (3D/Imax/обычный).

- 18 База данных эвакуированных автомобилей. Поля: улица, автостоянка, GPS-координаты, тип нарушения (стоянка на проезжей части в месте запрета, стоянка на тротуаре, стоянка на газоне), номер автомобиля, тип автомобиля (легковой/грузовой малой тонажности/грузовой большой тонажности).
- 19 База данных средних специальных учебных учреждений. Поля: название, адрес, тип подчинения (федеральный/региональный), год основания, номер лицензии, номер аккредитации, дата окончания действия аккредитации.
- 20 База данных поселков. Поля: название, девелопер, площадь, количество жителей.
- 21 База данных сухопутной военной техники. Поля: название, модель, разработчик, предприятие, стоимость, тип.
- 22 База данных деревьев в городе. Поля: GPS-координаты, вид дерева, округ, год посадки.
- 23 База данных футбольных матчей. Поля: дата, команда, команда, счет, место проведения.
- 24 База данных обращений жителей. Поля: дата, время, объект, заявитель, содержание обращения (до 255 символов), дата ответа, ответ на обращение (до 255 символов).
- 25 База данных студентов колледжа. Поля: ФИО, группа, признак бюджетности, стипендия (нет/обычная/повышенная) флаг наличия социальной стипендии, дата рождения.
- 26 База данных паевых инвестиционных фондов (ПИФ). Поля: название, тип (интервальный, открытый, закрытый), стоимость пая на момент открытия, среднее изменение стоимости пая за год, управляющая компания.
- 27 База данных посылок для таможенной службы. Поля: дата и время таможенного оформления посылки, отправитель, получатель, стоимость груза, вес посылки.
- 28 База данных занятости врачей в поликлинике. Поля: дата, время начала интервала, время окончания интервала, тип приема (первичный/повторный), пациент, врач.
- 29 База данных запросов по исправлению ошибок в программе. Поля: название, описание, дата регистрации запроса, дата завершения обработки запроса, ответственный программист, отправитель запроса.
- 30 База данных митингов. Поля: дата, время, адрес, количество участников заявленное, количество участников по факту (может не указываться), список заявителей, флаг разрешения.
- 31 База данных новостей. Поля: название, содержание, дата и время появления, агентство.
- 32 База данных арендаторов здания. Поля: название, дата начала аренды, дата окончания аренды, номер помещения.
- 33 База данных звезд. Поля: название, восхождение, склонение, видимая звездная величина.
- 34 База данных барельефов. Поля: описание, адрес здания, высота размещения (самой низкой точки барельефа), год размещения.
- 35 База данных арбузов, продаваемых в магазине. Поля: сорт, вес, дата поставки, стоимость.
- 36 База данных санитарных рубок в Московской области. Поля: название района, номер квартала, примерные GPS-координаты, площадь, дата начала, дата окончания.
- 37 База данных бракосочетаний. Поля: ФИО брачующегося, ФИО брачующейся, дата, возраст брачующегося, возраст брачующейся.
- 38 База данных естественных языков. Поля: название, примерное число носителей, примерный год появления, название географической области наибольшего распространения.
- 39 База данных буровых установок. Поля: GPS-координаты, наименование владельца, дата начала сооружения, наименование извлекаемого вещества.
- 40 База данных вагонов трамвайного депо. Поля: модель, дата приобретения, дата последнего ТО-1, дата последнего ТО-2.
- 41 База данных деловых встреч. Поля: ФИО встречаемого, дата и время встречи, место встречи, длительность встречи.
- 42 База данных заявок клиентов. Поля: ФИО клиента, дата и время приезда, описание проблемы, примерное время обслуживания, адрес, телефон.
- 43 База данных рейсов самолетов. Поля: номер рейса, дата и время вылета, дата и время прилета, аэропорт вылета, аэропорт прилета.
- 44 База данных online-курсов. Поля: название, URL, дата начала, длительность в днях, стоимость.
- 45 База данных полей. Поля: GPS-координаты поля, площадь поля, название засеянной культуры, номер месяца и номер года.
- 46 База данных химических элементов. Поля: название, количество протонов, количество нейтронов, количество электронов, год открытия.
- 47 База данных гидов. Поля: ФИО, дата рождения, специализация (текстовое поле), адрес проживания.
- 48 База данных врачей. Поля: ФИО, дата рождения, номер кабинета, флаг наличия кандидатской степени.
- 49 База данных горелок. Поля: Марка, тип соединения (текст), вес, мощность, наличие пьезоподжига.

- 50 База данных больниц. Поля: название, адрес, дата открытия, количество коек.
- 51 База данных депозитов. Поля: название, название банка, минимальная сумма, валюта, процент годовых, максимальная сумма, длительность в днях.
- 52 База данных зафиксированных нарушений ПДД. Поля: номер машины, GPS-координаты, дата и время, номер статьи и пункта.
- 53 База данных акций со скидками. Поля: название акции, название сети магазинов, максимальный процент скидки, описание акции, дата начала, дата окончания.
- 54 База данных диссертаций. Поля: название, тип (кандидатская, докторская), место защиты, дата защиты, ФИО защищающегося.
- 55 База данных философов. Поля: ФИО, дата рождения, дата смерти, основной труд, страна рождения.
- 56 База данных договоров страхования квартиры. Поля: ФИО, адрес, дата начала действия, срок действия в днях, страховая сумма, страховая премия.
- 57 База данных должителей. Поля: ФИО, страна проживания, дата рождения, комментарий.
- 58 База данных мест, где распространены грибы. Поля: GPS-координаты, площадь, преимущественный вид грибов, интервал (число, месяц – число, месяц), когда актуален сбор грибов в данном месте.
- 59 База данных спортивных площадок. Поля: Адрес, GPS-координаты, площадь, виды спорта.
- 60 База данных стадионов. Поля: Адрес, площадь, количество мест для зрителей, время начала и окончания работы (график работы).
- 61 База данных иностранных агентов. Поля: Адрес, телефон, количество вовлеченных лиц, сумма иностранных инвестиций за последний год в рублях, дата внесения в реестр иностранных агентов.
- 62 База данных жалоб на нарушения расписания движения общественного транспорта. Поля: тип (автобус, троллейбус, трамвай), номер, дата и время, название остановки, время ожидания.
- 63 База данных поездов дальнего следования. Поля: Станция отправления, станция назначения, дата и время отправления, дата и время прибытия, тип поезда (обычный/скорый/высокоскоростной).
- 64
- 65
- 66
- 67
- 68
- 69
- 70

3. Содержание отчета:

- (a) цель работы;
- (b) Исходный текст программы
- (c) Разработанный Makefile
- (d) Протокол git
- (e) Протокол системы отслеживания ошибок
- (f) Скриншоты wiki
- (g) Скриншоты результатов работы cppcheck
- (h) Исходные тексты UNIT-тестов
- (i) Самостоятельно написанные файлы для использования в autotools
- (j) Ответ на контрольные вопросы

4. Контрольные вопросы:

- (a) Как осуществляется компиляция и сборка программ, разбитых на несколько исходных файлов с использованием gcc?
- (b) Изложите методику работы с программой make.
- (c) Каковы достоинства использования программы make и разбиения программы на отдельные исходные файлы?
- (d) В чем сущность UNIT-тестирования? Как она осуществляется с использованием check?
- (e) Как осуществляется работа с wiki?
- (f) Опишите принцип работы системы отслеживания ошибок
- (g) Опишите назначение статического анализатора кода и метод работы с cppcheck.
- (h) Опишите методику работы с autotools.
- (i) Опишите назначение и методику работы с git.

4 Практическая работа №3. Создание библиотек (4 часа)

1. Цель работы: знакомство с созданием библиотек в среде Linux
2. Порядок выполнения:
 - (a) один из исходных файлов программы, созданную в практической работе №2, реализуйте в виде статической библиотеки; пошаговое руководство по созданию библиотек приведено по адресу <https://www.opennet.ru/docs/RUS/zlp/003.html>
 - (b) другой исходный файл – в виде динамической библиотеки
 - (c) осуществите соответствующие изменения в Makefile, результат зафиксируйте в git, проведите UNIT-тестирование, в случае необходимости используйте систему отслеживания ошибок
3. Содержание отчета:
 - (a) цель работы;
 - (b) исходный текст, что были изменены в ходе работы (включая Makefile);
 - (c) протокол git;
 - (d) протокол системы отслеживания ошибок (при необходимости);
 - (e) ответ на контрольные вопросы.
4. Контрольные вопросы:
 - (a) Опишите назначение и способ создания статической библиотеки
 - (b) Опишите назначение и способ создания динамической библиотеки

5 Практическая работа №4. Тестирование (6 часов)

1. Цель работы: практическое овладение инструментом поиска утечек памяти `valgrind`, получение опыта разработки тестовых наборов и анализа покрытия кода.
2. Порядок выполнения:
 - (a) напишите программу в соответствии с вашим вариантом;
 - (b) разработайте тестовый набор для проверки работы программы в различных условиях (когда не введены элементы, когда все элементы удаляются, когда элементы вставляются и т. д.) /результат – это таблица с четырьмя столбцами, первый столбец – номер теста, второй столбец – вводимые данные, третий столбец – ожидаемый результат, четвертый столбец – скриншот результата тестирования/;
 - (c) проверьте отсутствие утечек памяти для разработанного тестового набора и, в случае необходимости, осуществите отладку;
 - (d) с помощью `gcov` осуществите проверку полноты покрытия кода (в части операторов и в части веток ветвлений), в случае необходимости дополните тестовый набор и проведите дополнительное тестирование
3. Содержание отчета:
 - (a) цель работы;
 - (b) исходный текст программы;
 - (c) копия отчета, выведенного программой `valgrind`;
 - (d) копия отчета, выведенного программой `gcov`;
 - (e) ответы на контрольный вопрос.
4. Контрольные вопросы:
 - (a) опишите методику работы с программой `valgrind`, назначение инструментов поиска утечек памяти
 - (b) опишите назначение и способ использования программы `gcov`

Индивидуальные варианты задания

В программе во время обработки список должен просматриваться один раз (с 23-его варианта – два раза). Программа состоит из трех частей: формирование списка (ввод с клавиатуры); обработка; вывод списка.

- 1 Нахождение суммы всех элементов исходного списка, удаление всех элементов, равных 0, добавление квадрата элемента после каждого элемента списка, полученного после завершения предыдущих преобразований.
- 2 Нахождение произведения всех элементов исходного списка, удаление всех элементов, меньших 0, добавление единицы после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
- 3 Нахождение максимума всех элементов исходного списка, удаление всех четных элементов, добавление копии элемента после каждого элемента списка, пропорционального 3.
- 4 Нахождение минимума всех элементов исходного списка, удаление всех нечетных элементов, добавление после каждого элемента исходного списка его номера (каким он был в исходном списке).
- 5 Нахождение суммы всех четных элементов исходного списка, удаление всех чисел, оканчивающихся на 0, из списка, добавление после каждого элемента списка квадрата этого элемента.
- 6 Нахождение максимума всех четных элементов исходного списка, удаление всех нечетных элементов и добавление после каждого элемента списка, являющегося квадратом целого числа, значения -1 .
- 7 Нахождение минимума всех элементов, больших нуля; удаление всех отрицательных элементов и добавление после каждого элемента списка числа, на единицу большего.
- 8 Нахождение суммы всех элементов; удаление всех элементов, пропорциональных двум, добавление после каждого элемента списка куба этого элемента.
- 9 Нахождение суммы всех элементов, меньших нуля; удаление всех элементов, последняя цифра которых равна 5, добавление после каждого элемента списка числа, равного последней цифре данного элемента.
- 10 Нахождение наименьшего четного числа; удаление всех элементов, больших 10, добавление после каждого элемента списка квадрата данного числа.
- 11 Нахождение наименьшего числа, кратного трем; удаление всех элементов – двузначных натуральных чисел, добавление после каждого элемента, пропорционального двум, копии этого числа.
- 12 Нахождение наибольшего номера элемента списка (в исходном списке) числа, кратного трем; удаление всех элементов, в которых последние две цифры совпадают, добавление после каждого элемента числа, на единицу меньшего.
- 13 Нахождение суммы чисел, кратных трем; удаление всех элементов, оканчивающихся на 5, добавление после каждого положительного элемента, копии этого числа.
- 14 Нахождение произведения однозначных натуральных чисел; удаление всех элементов, меньших по модулю 5, добавление после каждого отрицательного элемента, копии этого числа.
- 15 Нахождение наибольшего номера элемента списка (в исходном списке) двузначного числа; удаление всех элементов, в которых последняя цифра больше 5, добавление после каждого элемента числа, на единицу большего.
- 16 Нахождение суммы всех чисел, кратных 5; удаление элементов, являющихся квадратом натурального числа, добавление после каждого положительного элемента его копии.
- 17 Нахождение произведения всех чисел, кратных 5; удаление элементов, стоящих на позиции, номер которой (в исходном списке) совпадает с самим элементом, добавление после каждого элемента его копии, уменьшенной в 10 раз (целой части полученного частного).
- 18 Нахождение наименьшего числа, кратного десяти; удаление всех двузначных натуральных чисел, добавление после трехзначных натуральных чисел копии.
- 19 Нахождение наибольшего числа, кратного десяти; удаление всех чисел, в которых разряд десятков и единиц совпадает, добавление после каждого числа, пропорционального последней цифре частного от деления числа на последнюю цифру.
- 20 Нахождение количества чисел, у которых последняя цифра – 5; удаление всех чисел, стоящих после числа, на единицу большего (в исходном списке), добавление после положительных чисел элемента списка, равного единице.
- 21 Нахождение произведения однозначных чисел списка; удаление чисел, меньших предыдущего (в исходном списке), добавление после каждого элемента остатка от деления его на 3.
- 22 Нахождение суммы всех четных чисел списка; удаление чисел, в которых разряд сотен равен разряду единиц, вставление после каждого элемента списка суммы его и первого элемента списка.

- 23 Удаление элементов равных номеру элемента в исходном списке, добавление после каждого элемента числа его единиц, нахождение минимума из оставшихся элементов списка.
- 24 Удаление элементов, пропорциональных последней цифры, добавление после каждого элемента его номера (каким бы он был после удаления элементов), нахождение максимума из оставшихся элементов списка.
- 25 Удаление элементов – однозначных чисел, добавление после каждого элемента его последней цифры, нахождение суммы оставшихся элементов списка.
- 26 Нахождение суммы всех положительных элементов исходного списка, удаление всех элементов, меньших 0, добавление куба элемента после каждого элемента списка, полученного после завершения предыдущих преобразований.
- 27 Нахождение произведения всех положительных элементов исходного списка, удаление всех элементов, больших 0, добавление на единицу большего элемента после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
- 28 Нахождение максимума всех отрицательных элементов исходного списка, удаление всех нечетных элементов, добавление числа, на единицу меньшего, чем элемент списка, после каждого элемента списка.
- 29 Нахождение минимума всех положительных элементов исходного списка, удаление всех четных элементов, добавление после каждого элемента исходного списка его номера (каким он был в исходном списке).
- 30 Нахождение максимума всех нечетных элементов исходного списка, удаление всех четных элементов и добавление после каждого элемента списка, являющегося кубом целого числа, значения -1 .
- 31 Нахождение суммы всех элементов исходного списка, удаление всех элементов, равных 0, добавление квадрата элемента перед каждым элементом списка, полученного после завершения предыдущих преобразований.
- 32 Нахождение произведения всех элементов исходного списка, удаление всех элементов, меньших 0, добавление единицы перед каждым четным элементом списка, полученного после завершения предыдущих преобразований.
- 33 Нахождение максимума всех элементов исходного списка, удаление всех четных элементов, добавление копии элемента перед каждым элементом списка, пропорциональным 3.
- 34 Нахождение минимума всех элементов исходного списка, удаление всех нечетных элементов, добавление перед каждым элементом исходного списка его номера (каким он был в исходном списке).
- 35 Нахождение суммы всех четных элементов исходного списка, удаление всех чисел, оканчивающихся на 0, из списка, добавление перед каждым элементом списка квадрата этого элемента.
- 36 Нахождение максимума всех четных элементов исходного списка, удаление всех нечетных элементов и добавление перед каждым элементом списка, являющимся квадратом целого числа, значения -1 .
- 37 Нахождение минимума всех элементов, больших нуля; удаление всех отрицательных элементов и добавление перед каждым элементом списка числа, на единицу большего.
- 38 Нахождение суммы всех элементов; удаление всех элементов, пропорциональных двум, добавление перед каждым элементом списка куба этого элемента.
- 39 Нахождение суммы всех элементов, меньших нуля; удаление всех элементов, последняя цифра которых равна 5, добавление перед каждым элементом списка числа, равного последней цифре данного элемента.
- 40 Нахождение наименьшего четного числа; удаление всех элементов, больших 10, добавление перед каждым элементом списка квадрата данного числа.
- 41 Нахождение наименьшего числа, кратного трем; удаление всех элементов – двузначных натуральных чисел, добавление перед каждым элементом, пропорционального двум, копии этого числа.
- 42 Нахождение наибольшего номера элемента списка (в исходном списке) числа, кратного трем; удаление всех элементов, в которых последние две цифры совпадают, добавление перед каждым элементом числа, на единицу меньшего.
- 43 Нахождение суммы чисел, кратных трем; удаление всех элементов, оканчивающихся на 5, добавление перед каждого положительного элемента, копии этого числа.
- 44 Нахождение произведения однозначных натуральных чисел; удаление всех элементов, меньших по модулю 5, добавление перед каждым отрицательным элементом, копии этого числа.
- 45 Нахождение наибольшего номера элемента списка (в исходном списке) двузначного числа; удаление всех элементов, в которых последняя цифра больше 5, добавление перед каждым элементом числа, на единицу большего.
- 46 Нахождение суммы всех чисел, кратных 5; удаление элементов, являющихся квадратом натурального числа, добавление перед каждым положительным элементом его копии.

- 47 Нахождение произведения всех чисел, кратных 5; удаление элементов, стоящих на позиции, номер которой (в исходном списке) совпадает с самим элементом, добавление перед каждым элементом его копии, уменьшенной в 10 раз (целой части полученного частного).
- 48 Нахождение наименьшего числа, кратного десяти; удаление всех двузначных натуральных чисел, добавление перед трехзначными натуральными числами копии.
- 49 Нахождение наибольшего числа, кратного десяти; удаление всех чисел, в которых разряд десятков и единиц совпадает, добавление после каждого числа, пропорционального последней цифре частного от деления числа на последнюю цифру.
- 50 Нахождение количества чисел, у которых последняя цифра – 5; удаление всех чисел, стоящих после числа, на единицу большего (в исходном списке), добавление после положительных чисел элемента списка, равного единице.
- 51 Нахождение произведения однозначных чисел списка; удаление чисел, меньших предыдущего (в исходном списке), добавление перед каждым элементом остатка от деления его на 3.
- 52 Нахождение суммы всех четных чисел списка; удаление чисел, в которых разряд сотен равен разряду единиц, вставление перед каждым элементом списка суммы его и первого элемента списка.
- 53 Удаление элементов равных номеру элемента в исходном списке, добавление перед каждым элементом числа его единиц, нахождение минимума из оставшихся элементов списка.
- 54 Удаление элементов, пропорциональных последней цифры, добавление перед каждым элементом его номера (каким бы он был после удаления элементов), нахождение максимума из оставшихся элементов списка.
- 55 Удаление элементов – однозначных чисел, добавление перед каждым элементом его последней цифры, нахождение суммы оставшихся элементов списка.
- 56 Нахождение суммы всех положительных элементов исходного списка, удаление всех элементов, меньших 0, добавление куба элемента перед каждым элементом списка, полученного после завершения предыдущих преобразований.
- 57 Нахождение произведения всех положительных элементов исходного списка, удаление всех элементов, больших 0, добавление на единицу большего элемента после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
- 58 Нахождение максимума всех отрицательных элементов исходного списка, удаление всех нечетных элементов, добавление числа, на единицу меньшего, чем элемент списка, перед каждым элементом списка.
- 59 Нахождение минимума всех положительных элементов исходного списка, удаление всех четных элементов, добавление перед каждым элементом исходного списка его номера (каким он был в исходном списке).
- 60 Нахождение максимума всех нечетных элементов исходного списка, удаление всех четных элементов и добавление перед каждым элементом списка, являющегося кубом целого числа, значения -1 .

6 Практическая работа №5. Компоновка программ, написанных на различных языках программирования (2 часа)

Теоретическое введение

При вызове подпрограмм используются различные соглашения о способе их вызова. Транслятор gcc по умолчанию использует следующее соглашение для функций, имеющих три аргумента целого типа (ABI SysV):

- при использовании 32-битного режима для передачи параметров используется стек: в него записываются аргументы в обратном порядке (на каждый аргумент по 4 байта), после чего осуществляется вызов функции (при котором в стек заносится адрес возврата, занимающий также 4 байта). Таким образом первый аргумент расположен по адресу $[4+ESP]$, второй – $[8+ESP]$ и так далее. Возвращаемое значение передается через регистр EAX.
- при использовании 64-битного режима для передачи параметров используются регистры процессора: rdi, rsi, rdx; результат возвращается через регистр rax.

Для простоты в ваших программах рекомендуется сохранять значения всех регистров, кроме eax (rax).

Для трансляции объектного файла в исходном коде FASM записывается строка: FORMAT ELF (FORMAT ELF64) для 32-битного и 64-битного режима соответственно. Функция задается меткой, при этом необходимо написать строчку

```
public <название функции>
```

где название функции указывается без угловых скобок. В программе на языке Си указывается объявление вашей функции, описанной на языке Ассемблера.

Линковку следует осуществлять с помощью команды

```
gcc <список объектных файлов> -o <исполняемый файл>
```

Задание

1. Цель работы: практическое закрепление знаний о редакторах связей и программах автоматизации сборки при использовании различных языков программирования
2. Порядок выполнения:
 - (a) напишите программу, осуществляющую ввод трех чисел и вызов подпрограммы, описанной в другом объектном файле;
 - (b) напишите исходный файл на языке ассемблера, предназначенный для вычисления функции согласно вашему варианту;
 - (c) напишите Makefile для компиляции ваших исходных файлов
 - (d) проведите компиляцию исходных файлов, попутно осуществляя тестирование и отладку.

Индивидуальные варианты:

- 1 $(x+2*y+3*z)/4$
- 2 $(3*x+z*y)\%4$
- 3 $(x-y*z+3)/5$
- 4 $x/5+y*2-3*z$
- 5 $(x-2*y)/z$
- 6 $(2*x+z)\%y$
- 7 $x\%2-y/z$
- 8 $x*x+z\%y$
- 9 $(x/y+z)\%x$
- 10 $((10-x)*(y+2)\%z$
- 11 $(x*y-z\%5)/2$
- 12 $(x\%5+y|z)*3$
- 13 $(x-y\&(z+15))\%10$
- 14 $(3-x\%5)/(y+2*z)$
- 15 $(x+(y<<2)\%5+z)$
- 16 $(x/3+(y>>3)*z)$
- 17 $(x|3+y\&(\sim 3))\%125$
- 18 $((x*y*y-z)\%65)$
- 19 $(x/4+y\%z)<<2$
- 20 $(x\%5+z/y)>>2$
- 21 $(x|(y\&z))\%125$
- 22 $(x<<2+y>>2+z\%3)$
- 23 $(x/(y/z)+y*3)$
- 24 $(x|(\sim y))/z-23$
- 25 $(y\&(x|5)-z\%256)$

3. Содержание отчета:

- (a) цель работы;
- (b) Исходный текст программы
- (c) Разработанный Makefile
- (d) Протокол git
- (e) Протокол системы отслеживания ошибок
- (f) Ответ на контрольные вопросы

4. Контрольные вопросы:

- (a) Как осуществляется компиляция и сборка программ, разбитых на несколько исходных файлов с использованием gcc и fasm?
- (b) Как передаются аргументы в функцию, реализованную на языке Си?
- (c) Опишите использованные вами команды языка Ассемблера.

7 Практическая работа №6. Автоматизация тестирования GUI (2 часа)

1. Цель работы: практическое овладение инструментом автоматизации тестирования AutoIT.
2. Порядок выполнения:
 - (a) С помощью средства AutoIt проведите тестирование любой формы в вашем приложении (из курсовой работы или дипломного проекта). Форма должна содержать хотя бы три активных элемента управления.
3. Содержание отчета:
 - (a) цель работы;
 - (b) исходный текст скрипта AutoIt;
 - (c) изображение экрана с тестируемой программой;
 - (d) ответ на контрольный вопрос.
4. Контрольные вопросы:
 - (a) опишите методику работы с программой AutoIT.

8 Практическая работа №7. Средства управления требованиями (2 часа)

1. Цель работы: практическое овладение инструментом управления требованиями
2. Порядок выполнения:
 - (a) осуществите внесение списка требований к вашему курсовому или дипломному проекту с использованием специализированного инструментального средства (например, [ProR]<http://www.eclipse.org/rmf/pror/>).
 - (b) разработайте дополнительный список требований на основе требований, разработанных в предыдущем пункте, описывающий способ их реализации с точки зрения проектировщика/программиста (эту часть работу достаточно осуществить в рамках отведенного времени на работу). Свяжите исходные требования с требованиями из дополнительного списка.
3. Содержание отчета:
 - (a) цель работы;
 - (b) экранные копии списка требований.
 - (c) ответы на контрольные вопросы.
4. Контрольные вопросы:
 - (a) опишите методику работы с выбранным вами средством управления требованиями;
 - (b) ответы на контрольные вопросы.

9 Практическая работа №8. Средство создания документации (2 часа)

1. Цель работы: практическое овладение средством создания документации (технология DOCBOOK).
2. Порядок выполнения:
 - (a) с использованием технологии DOCBOOK создайте пользовательскую документации для курсового проекта или для дипломного проекта (в процессе работы исходный текст необходимо создавать «руками»). Размер документации должен соответствовать тому, что на работу отводится одна пара. Для выполнения работы достаточно ознакомиться со статьей <http://www.osp.ru/os/2005/03/185403/>
3. Содержание отчета:
 - (a) цель работы;
 - (b) исходный текст пользовательской документации на языке DOCBOOK;
 - (c) экранная копия результата его преобразования в формат HTML, WORD или PDF;
 - (d) ответы на контрольные вопросы.
4. Контрольные вопросы:
 - (a) опишите методику работы с DOCBOOK;
 - (b) перечислите и опишите использованные элементы и атрибуты DOCBOOK.

10 Практическая работа №9 Инструментальные средства проектирования (4 часа)

1. Цель работы: практическое овладение инструментальным средством проектирования

2. Порядок выполнения:

- (a) выберите инструментальное средство проектирования, использующее нотацию UML, позволяющее осуществить автоматическую генерацию исходного кода на выбранном вами языке программирования (рекомендуется Umbrello);
- (b) осуществите проектирование путем разработки диаграммы классов для индивидуального задания с использованием всех принятых в сообществе программистов соглашений (приватность свойств, использование инкапсуляции, наследования, полиморфизма);
- (c) осуществите автоматическую генерацию исходного кода на выбранном вами языке программирования;
- (d) допишите программу в соответствии с вашим заданием (проверять корректность исходных данных не требуется).

Варианты задания:

- 1 Товары Интернет-магазина – книги и диски. Все товары определяются ценой, книги имеют название, автора, количество страниц; диски – название, количество треков. Выведите на экран все товары со стоимостью меньше 100 рублей.
- 2 Преподаватели определяются ФИО. Для тех, кто имеют диссертацию дополнительно вводится ее название; для остальных – стаж работы. Вывести всех преподавателей, у которых ФИО начинается на букву «А».
- 3 Телефоны определяются названием модели. Проводные телефоны дополнительно определяются типом номеронабирателя (диск или кнопки); а беспроводные – дальностью действия радиосигнала. Вывести все телефоны, название которых начинается на «А».
- 4 Покатушки определяются названием и расстоянием. Однодневные катушки дополнительно определяются плановым временем поездки (в часах). Многодневные катушки определяются количеством дней и категорией сложности похода (от 1 до 6). Вывести все покатушки длиной более 100 км.
- 5 Музыкальная композиция определяется названием и композитором. Дополнительно для песни указывается автор стихов. Выведите информацию о всех композициях, у которых композитор начинается на букву «А».
- 6 Олимпиада определяется названием. Если олимпиада участвует в программе приема в ВУЗы дополнительно указывается уровень олимпиады (1–3), если олимпиада – этап всероссийской, то указывается название этапа (школьная, окружная, региональная, всероссийская), в остальных случаях – размер призового фонда. Выведите все олимпиады, название которых начинается на букву «А».
- 7 Проездной билет определяется стоимостью. Билет на количество поездок определяется количеством поездок. Билет на неограниченное количество поездок определяется сроком действия (1 день, 5 дней, 10 дней, 15 дней, месяц, три месяца, 6 месяцев, год). Выведите информацию о билетах, стоимостью меньше 300 рублей.
- 8 Информация о студенте определяется ФИО. Для студентов, не имеющих автомата, указывается балл, полученный на экзамене (2–5); для студентов, имеющих автомат указывается основание (олимпиада или контрольные работы). В случае, если контрольная работа – то также указывается средний балл за к/р. Выведите всю информацию о студентах с фамилией, начинающейся на буквы от А до К.
- 9 Сотовый телефон определяется названием. Для смартфонов указывается операционная система. А для других телефонов – наличие браузера. Выведите информацию о телефонах, название которых содержит слово «Nokia».
- 10 Куртка определяется названием модели, наличием капюшона. Для мембранных курток указывается степень водонепроницаемости (число в мм рт. ст.), для остальных – наличием пропитки. Выведите информацию обо всех куртках, имеющих капюшон.
- 11 Жесткий диск определяется названием и емкостью. Внешние жесткие диски определяются дополнительно наличием системы, смягчающей последствия падения. Внутренние жесткие диски – размером (2.5/3.5 дюйма). Выведите информацию о дисках, емкостью больше 200 Гб.
- 12 Велосипед определяется названием модели. Горному велосипеду соответствует количество скоростей, BMX – тип конструкции (фривил, кассетная, фрикостер). Выведите информацию обо всех велосипедах, содержащих в названии «Nogso».
- 13 Электронная книга определяется названием и размером экрана. Для EInk-дисплея указывается поколение (pearl, vizplex); для LCD – количество поддерживаемых цветов. Выведите информацию о всех книгах с размером экрана не менее 7 дюймов.
- 14 GPS определяется названием, диагональю экрана. Для переносных GPS указывается наличие велосипедного крепления; для автомобильных – поддержка отображения пробок и наличие радар-детектора. Выведите информацию обо всех GPS с размером экрана менее 7 дюймов.
- 15 Пылесос определяется названием модели. Для обычного пылесоса указывается мощность, для пылесоса-робота – размер убираемого помещения и количество виртуальных стен. Выведите информацию обо всех пылесосах, содержащих в названии слово Indesit.

- 16 Туры определяются названием. Для пляжного тура указывается тип пляжа (галечный, песок); для экскурсионного – количеством экскурсий. Выведите информацию обо всех турах, содержащих слово Египет.
- 17 Язык программирования определяется названием. Алгоритмические языки определяются поддержкой ООП (отсутствует, на классах, прототипная), остальные языки – типом (функциональный, логический, стиль). Выведите информацию обо всех языках, название которых начинается с буквы «А».
- 18 Контрагенты определяются названием. Индивидуальные предприниматели дополнительно определяются наличием счета в банка, а юридические лица – формой организации (ООО, ОАО, ЗАО). Выведите информацию обо всех контрагентах, название которых начинается с буквы «А».
- 19 Счет в банке определяется номером. Для текущего счета указывается плата за обслуживание, для сберегательного счета – проценты годовых и наличие капитализации. Выведите информацию обо всех счетах, номер которого начинается с 408178...
- 20 Автомобильная дорога определяется названием и километражем. Бесплатная дорога определяется статусом автомагистрали (автомагистраль или нет), а платная – стоимостью за километр для обычных пользователей. Выведите информацию о дорогах, длина которых менее 100 км.
- 21 Офисное здание определяется адресом. В случае наличия стоянки указывается количество машиномест и стоимость аренды за месяц. Выведите информацию о зданиях, в адресе которых присутствует слово Тверская.
- 22 Товары Интернет-магазина – GPS-навигаторы и карты. Все товары определяются ценой и названием, GPS-навигаторы имеют назначение (ручной, автомобильный) и признак возможности загрузки карт; карты – размером (в Мб). Выведите информацию о всех товарах со стоимостью менее 4000 рублей.
- 23 Товары Интернет-магазина – чай и кофе. Все товары определяются ценой, названием и весом, кофе – типом (растворимый, молотый, в зернах), чай – типом (черный, зеленый). Выведите информацию о всех товарах с весом менее 150 г.
- 24 Объекты продаваемые в коттеджном поселке: участки (определяются площадью, стоимостью, наличием подряда), дома (определяются этажностью, площадью и стоимостью). Выведите все объекты со стоимостью меньше 1000000 рублей.
- 25 Вопросам теста соответствует формулировка и количество баллов за правильный ответ. Вопросам с вариантами правильных ответов соответствует 4 варианта ответа и номер правильного ответа; остальным вопросам – формулировка правильного ответа. Выведите все вопросы, оцениваемые в 10 баллов и выше.
- 26 Слова определяются собственно словом. Для существительных указывается род, для глаголов – спряжение. Выведите информацию обо всех словах, начинающихся на букву «А».
- 27 Операционная система определяется названием. Для операционной системы на базе Linux указывается название менеджера пакетов; для остальных – стоимость лицензии. Вывести все операционные системы, у которых название начинается на букву «А».
- 28 Рюкзаки определяются названием модели и емкостью. Для городских рюкзаков указывается наличие «вентилируемой спины» для походных – количество отделений и наличие крепления для трекинговых палок. Вывести информацию обо всех рюкзаках, в названии которых присутствует слово «Trek».
- 29 Автостоянка определяется названием, количеством машиномест. Для крытой автостоянки указывается количество этажей. Для открытой стоянки – наличие охраны. Вывести информацию обо всех автостоянках с количеством мест больше 20.
- 30 Партия определяется названием. Для тех партий, что финансируются из бюджета указывается размер ассигнований, а для остальных – количество депутатов в каких-либо представительных органах власти. Вывести информацию обо всех партиях с названиями, начинающимися на буквы от «А» до «К».

3. Содержание отчета:

- (a) цель работы;
- (b) печатная копия диаграммы классов;
- (c) исходный код разработанного программного изделия;
- (d) ответ на контрольный вопрос.

4. Контрольные вопросы:

- (a) опишите элементы UML, используемые в диаграммах классов.

11 Практическая работа №10. Средства автоматизированного программирования (4 часа)

1. Цель работы: практическое ознакомление со средствами автоматизированного программирования
2. Порядок выполнения:

- (a) создайте исходный файл для программы flex в соответствии с вашим заданием (простейшей правильное решение занимает не более 10 строк);
- (b) *создайте исходный файл для программы yacc в дополнение к исходному файлу, созданному в первой части задания (по желанию);*
- (c) осуществите трансляцию исходных файлов с помощью flex, yacc, транслятора языка C;
- (d) осуществите тестирование и отладку программного изделия.

Варианты задания:

- 1 Программа проверки правильности оператора case с переменной-селектором типа char (без else), в который вложены операторы присваивания переменной строковой константы.
- 2 Программа проверки правильности оператора присваивания, в правой части которого допустимы операции сложения, вычитания, умножения, деления, переменные, целые и вещественные числа (включая показательную форму).
- 3 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для integer), а тип-элемента – название типа (идентификатор).
- 4 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), перечислимый тип, а тип-элемента – ограниченный тип для integer.
- 5 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для char), а тип-элемента – перечислимый тип.
- 6 Программа проверки описания перечислимого и ограниченного типа (для integer, char). Строка должна начинаться со слова type.
- 7 Программа проверки правильности оператора write(ln), у которого в качестве аргументов могут участвовать строковый литералы, целые и вещественные числа, выражения над числами, соединенные операциями сложения, вычитания, умножения и деления. Может также быть указан формат вывода.
- 8 Проверка правильности последовательности операторов read(ln), присваивания и write(ln). У writeln возможные аргументы – строковые литералы; у read(ln) – названия переменных, элементы массивов. У оператора присваивания слева название переменной или элемент массива, справа число (целое или вещественное).
- 9 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть как целые числа, так и символы, а тело цикла – оператор write(ln), у которого аргументы – целые и вещественные числа и переменные.
- 10 Проверка правильности оператора if, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Как в части then, так и в else (может быть опущен) указывается оператор write(ln) с аргументами – переменными и строками.
- 11 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа integer, переменные и операции +, -, *.
- 12 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа char, переменные и операции +, -, *.
- 13 Проверка правильности оператора while, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор while, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.
- 14 Проверка правильности оператора if, у которого условие имеет вид: <переменная><знак><строка>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно. Как в части then, так и в else (может быть опущен) указывается оператор присваивания, правая часть которого – выражение, содержащее переменные, целые и вещественные числа и знаки операций +, -, *, /.
- 15 Проверка правильности оператора repeat..until, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор repeat..until, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.
- 16 Программа проверки правильности оператора case с переменной-селектором перечислимого типа (без else), в который вложены операторы присваивания переменной константы типа integer.
- 17 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями перечислимого типа, переменные и операции +, -, *.
- 18 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть значения перечислимого типа, а тело цикла – оператор присваивания, в правой части которого записана числовая константа.

- 19 Проверка правильности оператора `if`, у которого условие – это логическое выражение над переменными логического типа. Как в части `then`, так и в `else` (может быть опущен) указывается оператор `write(ln)` с аргументами – переменными и числами.
- 20 Проверка правильности оператора `while`, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, умножения и деления.
- 21 Проверка правильности оператора `repeat..until`, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, `mod` и `div`.
- 22 Проверка правильности оператора `repeat..until`, у которого условие имеет вид: `<переменная><знак><строка>`, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа `string`. Телом цикла является либо снова оператор `repeat..until`, либо оператор `writeln`, аргументами которого являются целые и вещественные числа, возможно, с указанием формата.
- 23 Проверка правильности оператора `while`, у которого условие имеет вид: `<переменная><знак><строка>`, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа `string`. Телом цикла является либо снова оператор `while`, либо оператор `writeln`, аргументами которого являются выражения, содержащие целые числа, имена переменных и операции `+`, `-`, `*`, `/`.
- 24 Программа проверки правильности оператора `write(ln)`, у которого в качестве аргументов могут участвовать строковый литералы и переменные, соединенные операцией сложения; целые числа и переменные, соединенные операциями сложения, вычитания, `mod` и `div`. Может также быть указан формат вывода.
- 25 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются строковые литералы и переменные, соединенные знаком `+`; а также выражения, в которых используются литералы типа `integer`, переменные и операции сложения, вычитания, `div` и `mod`.

3. Содержание отчета:

- (a) цель работы;
- (b) исходные тексты для `flex` и, возможно, `yacc`;
- (c) ответ на контрольный вопрос.

4. Контрольные вопросы:

- (a) опишите синтаксис и семантику исходного текста для программы `flex` (и для программы `yacc` при ее использовании).
- (b) в чем смысл использования `flex`?

12 Практическая работа №11. Средства обратной разработки (2 часа)

1. Цель работы: практическое ознакомление со средствами обратной разработки

2. Порядок выполнения:

- (a) Путем декомпиляции программы, выданной преподавателем, определите по какой формуле осуществляется расчет программой (вычисление осуществляет функция `f`, ее аргументы называются `x`, `y` и `z`). **Указание:** декомпиляцию можно осуществить с помощью утилиты `objdump` (опция `-d`).

3. Содержание отчета:

- (a) цель работы;
- (b) экранная копия содержимого функции `f` в средстве обратной разработки;
- (c) формула, вычисляемая в программе (с использованием синтаксиса языка C);
- (d) ответы на контрольные вопросы.

4. Контрольные вопросы:

- (a) опишите команды ассемблера, встретившиеся в вашей работе и их роль в результирующей формуле

А Список используемого программного обеспечения

1. Современная ОС на базе GNU Linux
2. gcc (Linux)
3. bitbucket.org (облачный сервис)
4. <https://libcheck.github.io/check/> (Linux)
5. cppcheck (Linux)
6. automake (Linux)
7. autoconf (Linux)
8. git (Linux)
9. ar (Linux)
10. fasm (Linux)
11. valgrind (Linux)
12. autoit (Windows)
13. xsltproc (Linux)
14. docbook-xsl (Linux)
15. objdump (Linux)
16. flex (Linux)
17. Umbrello

В Список источников

1. Руководство по созданию статических и динамических библиотек: <https://www.opennet.ru/docs/RUS/zlp/003.html>
2. Руководство по gcc <https://www.opennet.ru/docs/RUS/gcc/>
3. Введение в make www.opennet.ru/base/dev/mini_make.txt.html
4. Руководство cppcheck <http://cppcheck.sourceforge.net/manual.pdf> (англ.)
5. Руководство git <https://git-scm.com/book/ru/v1>
6. Пошаговая инструкция по использованию autotools <http://blablacode.ru/linux/500>
7. Руководство по libcheck https://libcheck.github.io/check/doc/check_html/check_3.html (англ.)
8. Введение в valgrind http://www.opennet.ru/base/dev/valgrind_memory.txt.html
9. Введение в gcov <http://www.linux-mag.com/id/1409/> (англ.)
10. Перевод документации autoit http://azjio.narod.ru/autoit3_docs
11. Обзор docbook <http://www.osp.ru/os/2005/03/185403/>
12. Лекция, описывающая flex http://csc.sibsutis.ru/sites/csc.sibsutis.ru/files/courses/spo/Lecture_8.pdf