

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ МОРДОВИЯ

**Государственное автономное профессиональное
образовательное учреждение Республики Мордовия
«Саранский автомеханический техникум»**

У Т В Е Р Ж Д А Ю

Заместитель директора по УМР
ГАПОУ РМ «Саранский
автомеханический техникум»
_____ Е.С. Синичкина

«31» августа 2022 г.

Методические указания
по выполнению практических работ
ПМ.05. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА
ИНФОРМАЦИОННЫХ СИСТЕМ
МДК. 05.03 ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

09.02.07 Информационные системы и программирование

Лабораторная работа №1 «Разработка тестового сценария проекта» Цель: получить навыки разработки тестовых сценариев.

Теоретические вопросы

- Оценка стоимости и причины ошибок в программном обеспечении.
- Виды и методы тестирования.
- Понятие теста.
- Требования к разработке тестовых сценариев.
- Правила разработки тестовых сценариев.

Задание № 1

Написать программу решения квадратного уравнения $ax^2 + bx + c = 0$.

Задание № 2

Найти минимальный набор тестов для программы нахождения вещественных корней квадратного уравнения $ax^2 + bx + c = 0$. Решение представлено в таблице.

Но-мер теста	a	b	c	Ожидаемый результат	Что проверяется
1	2	-5	2	$x_1=2, x_2=0,5$	Случай вещественных корней
2	3	2	5	Сообщение	Случай комплексных корней
3	3	-12	0	$x_1=4, x_2=0$	Нулевой корень
4	0	0	10	Сообщение	Неразрешимое уравнение
5	0	0	0	Сообщение	Неразрешимое уравнение
6	0	5	17	Сообщение	Неквадратное уравнение
7	9	0	0	$x_1=x_2=0$	Нулевые корни

Таким образом, для этой программы предлагается минимальный набор функциональных тестов, исходя из 7 классов выходных данных.

Заповеди по отладки программного средства, предложенные Г. Майерсом.

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам, нежелательно тестировать свою собственную программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Документируйте пропуск тестов через компьютер, детально изучайте результаты каждого теста, избегайте тестов, пропуск которых нельзя повторить. *Заповедь 5.* Каждый модуль подключайте к программе только один раз, никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

Задание № 3

Разработайте набор тестовых сценариев (как позитивных, так и негативных) для следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2 строки. На выходе приложение выдает число вхождений второй строки в первую. Например:

Строка 1	Строка 2	Вывод
абвгабвг	аб	2
стстсап	стс	2

Набор тестовых сценариев запишите в виде таблицы, приведенной выше.

Задание № 4

Оформить отчет.

Лабораторная работа №21 «Разработка тестовых пакетов» Цель: получить навыки разработки тестовых пакетов.

Теоретические вопросы

- Системные основы разработки требований к сложным комплексам программ.
- Формализация эталонов требований и характеристик комплекса программ.
- Формирование требований компонентов и модулей путем декомпозиции функций комплексов программ.
- Тестирование по принципу «белого ящика».

Задание № 1

В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пользуясь изложенным способом создать программу, которая: а) зашифрует введенный текст и сохранит его в файл; б) считает зашифрованный текст из файла и расшифрует данный текст.

Задание № 2

Спроектировать тесты по принципу «белого ящика» для программы, разработанной в задании № 1. Выбрать несколько алгоритмов для тестирования и обозначить буквами или цифрами ветви этих алгоритмов. Выписать пути алгоритма, которые должны быть проверены тестами для выбранного метода тестирования. Записать тесты, которые позволят пройти по путям алгоритма. Протестировать разработанную вами программу. Результаты оформить в виде таблиц:

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
...

Задание № 3

Проверить все виды тестов и сделать выводы об их эффективности

Задание № 4

Оформить отчет.

Лабораторная работа №22 «Использование инструментария анализа качества» Цель: получить навыки использования инструментария анализа качества.

Теоретические вопросы

- Общие требования к качеству функционирования сложных программных комплексов.
- Требования к характеристикам качества сложных программных комплексов.
- Требования к эффективности использования ресурсов ЭВМ программным комплексом в реальном времени.
- Проверка корректности функциональных требований к сложным комплексам программ.

Задание № 1

Написать программу, генерирующую массив вещественных чисел в диапазоне от -10 до 10 и определяющую все минимальные положительные элементы.

Задание № 2.

Оценить эффективность разработанной программы:

	Исходная программа		Улучшенная программа	
	Недостатки	Количественная оценка	Улучшения	Количественная оценка
Время выполнения				
Оперативная память				
Внешняя память				

Задание № 3

Оценить качество разработанной программы

	Правильность	Универсальность	Проверяемость	Точность результатов
Недостатки				
Оценка				

Задание № 4

Оформить отчет.

Лабораторная работа №23 «Анализ и обеспечение обработки исключительных ситуаций»

Цели: получение навыков анализа и обеспечения обработки исключительных ситуаций.

Теоретические вопросы

- Исключения в C++.
- Установленные исключения.
- Спецификация исключения.

Задание № 1

Написать программу, в которой обрабатываются следующие исключительные ситуации: "отрицательное значение возраста" и "год рождения больше текущего":

```
#include <iostream>
#include <ctime>
using namespace std;

int AgeCalc(int year)
{
    if (year <= 0)
    {
        throw "ERROR: negative value of the birth year!!!";
    }

    struct tm *CDate;
    time_t tt = time(NULL);
    CDate = localtime(&tt);

    if (year > (1900 + CDate->tm_year) )
    {
        throw "ERROR: The birth year value is greater than current year value!!!";
    }

    return 1900 + CDate->tm_year - year;
}

int main()
{
    int BYear = 1980;
    int PAge = 0;

    try{
        PAge = AgeCalc(BYear);
    }
    catch (const char * s)
    {
        cout << s << endl<<endl;
    }
    catch(...)
    {
        cout << "Unknown exception" << endl<<endl;
    }

    cout << "For birth year " << BYear << " the age is " << PAge << endl;

    return 0;
}
```

Задание № 2

Составить программу циклического вычисления значений функций, определенных из таблицы вариантов заданий. Значения R должны вводиться с клавиатуры. R1 и R2 – вещественные, R3 – комплексное. Предусмотреть

Вывод подсказок в виде (например):

Funkciya $\sin(x)$

Q –Viyhod iz
programmyi Vvedite
chislo ili Q:

Для вычисления значений функции написать функцию, вычисляющую требуемые по заданию значения. При разработке функции разрешается использовать функции модуля `math.h`.

Предусмотреть анализ всей введенной информации на ошибки, обработку ошибок реализовать с использованием обработчиков `try...` в зависимости от варианта задания. Предусмотреть вывод имени функции, в которой произошла ошибка. Вывод на экран и чтение с клавиатуры организовать при помощи стандартных потоков ввода/вывода/ошибки.

Вывести исходные данные и результат в виде (например):

`Sin(R) = rez;`

Где `rez` –результаты вычисления (вещественный).

Вариант	Функция	Обработчики
1	<code>Sin(R1)*(pi)/R2-R3</code>	Потеря разряда Деление на 0
2	<code>Sin(R2)/pi*R1+R3</code>	Потеря разряда Переполнение
3	<code>Tan(R1)/R3+Cmod(R3)</code>	Потеря разряда Прерывание
4	<code>Arctan(R1)*R2+ R3</code>	Потеря разряда Переполнение
5	<code>Ln(R1-R2)*R2-R3</code>	Обл.опр.арг. Исчезновение порядка

Задание № 3

Реализуйте класс «очередь» из строк. Реализуйте методы для вставки в очередь удаления.

Породите и обработайте ошибки динамического выделения памяти, переполнения очереди.

Задание № 4

Оформите отчет.

Лабораторная работа №24 «Функциональное тестирование» Цель: получение навыков проведения функционального тестирования.

Теоретические вопросы

- Особенности функционального тестирования программного обеспечения (тестирования «черного ящика»).
- Ошибки, выявляемые при функциональном тестировании.
- Задачи, решаемые при функциональном тестировании.

Задание № 1

Опишите методы формирования тестовых наборов при использовании стратегии "черного ящика":

Эквивалентное разбиение	
Анализ граничных значений	
Анализ причинно-следственных связей	
Предположение об ошибке	

Задание № 2

Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной из осей координат.

В основе программы лежит решение системы линейных уравнений $Ax + By = C$ и $Dx + Ey = F$.

1. Используя метод эквивалентных разбиений, получаем для всех коэффициентов один правильный класс эквивалентности (коэффициент – вещественное число) и один неправильный (коэффициент – не вещественное число). Откуда можно предложить 7 тестов:
 - 1) все коэффициенты – вещественные числа;
 - 2)– 7) поочередно каждый из коэффициентов – не вещественное число.
2. По методу граничных условий можно считать, что для исходных данных граничные условия отсутствуют (коэффициенты – "любые" вещественные числа); для результатов – получаем, что возможны варианты: единственное решение, прямые сливаются (множество решений), прямые параллельны (отсутствие решений). Следовательно, можно предложить тесты, с результатами внутри области и с результатами на границе.
3. По методу анализа причинно-следственных связей определяем множество условий. а) для определения типа прямой; б) для определения точки пересечения.

Выделяем три группы причинно-следственных связей (определение типа и существования первой линии, определение типа и существования второй линии, определение точки пересечения) и строим таблицы истинности.

К уже имеющимся тестам добавляются:

- а) проверки всех случаев расположения обеих прямых – 6 тестов по первой прямой вкладываются в 6 тестов по второй прямой так, чтобы

варианты не совпадали, – 6 тестов;

б) выполняется отдельная проверка несовпадения условия $x = 0$ или $y = 0$ (в зависимости

от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

4. По методу предположения об ошибке добавим тест, при котором все коэффициенты

– нули. Всего получили 20 тестов по всем четырем методикам. Если еще попробовать вложить независимые проверки, то возможно число тестов можно еще сократить.

Задание № 3

Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

Предлагаемые тесты свести в таблицу.

Номер теста	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод

Задание № 4

Разработать программу решения уравнения $ax^2 + bx + c = 0$, где a, b, c – любые вещественные числа.

Предлагаемые тесты свести в таблицу.

Номер теста	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод

Задание № 5

Оформить отчет.

Лабораторная работа №25 «Тестирование безопасности» Цель: получение навыков тестирования безопасности информационной системы. **Теоретические вопросы**

- Тестирование восстановления.
- Тестирование безопасности.
- Технологии тестирования безопасности.
- Тестирование безопасности – оценка уязвимости программного обеспечения к различным атакам.

Компьютерные системы очень часто являются мишенью незаконного проникновения. Под проникновением понимается широкий диапазон действий: попытки хакеров проникнуть в систему из спортивного интереса, месть рассерженных служащих, взлом мошенниками для незаконной наживы. Тестирование безопасности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение. В ходе тестирования безопасности испытатель играет роль взломщика. Ему разрешено все:

- попытки узнать пароль с помощью внешних средств;
- атака системы с помощью специальных утилит, анализирующих защиты;
- подавление, ошеломление системы (в надежде, что она откажется обслуживать других клиентов);
- целенаправленное введение ошибок в надежде проникнуть в систему в ходе восстановления;
- просмотр несекретных данных в надежде найти ключ для входа в систему.

При неограниченном времени и ресурсах хорошее тестирование безопасности взламывает любую систему. Задача проектировщика системы – сделать цену проникновения более высокой, чем цена получаемой в результате информации.

Задание № 1

Изучите и опишите одно из средств выявления уязвимостей:

Таблица 1. Обзор средств выявления уязвимостей, работающих на уровне кода

Наименование средства	Назначение	Поддерживаемый язык и программирования	Примечание
	Иностраные средства выявления уязвимостей		

Its4	Статически просматривает исходный код для обнаружения потенциальных уязвимостей защиты	C/c++	Отмечает вызовы потенциально опасных функций, таких, как strcpy/memcpy, и выполняет поверхностный семантический анализ, пытаюсь оценить, насколько опасен такой код, а также дает советы по его улучшению
Rats(rough auditing tool for security)	Просматривает исходный текст, находя потенциально опасные обращения к функциям	C/c++, ph p, perl, python	глубокого семантического анализа в поисках дефектов, способных привести к переполнению буфера, полученных из tops
Flawfinder	Просматривает исходный текст, находя потенциально опасные обращения к функциям	C/c++	Выполняет поиск функций, которые чаще всего используются некорректно, присваивает им коэффициенты риска (опираясь на такую информацию, как передаваемые параметры) и составляет список потенциально уязвимых мест, упорядочивая их по степени риска

Flexelint (pc - lint)	Производит семантический анализ исходного кода, анализ потоков данных и управления	C/c++	В конце работы выдаются сообщения нескольких основных типов: – возможен нулевой указатель – проблемы с выделением памяти (например, нет free() после malloc()) – проблемный поток управления (например, недостижимый код); – возможно переполнение буфера, арифметическое переполнение; – предупреждения о плохом и потенциально опасном стиле кода
Parasoft c+ + test Coverity	Формирование тестов анализа уязвимостей на уровне метода, класса, файла и проекта Используется для выявления и исправления дефектов безопасности и качества в приложениях критического назначения	C++ C/c++, java	Генерирует тестовый код, вызывая для его подготовки компилятор visual c++ Способен с минимальной погрешностью обрабатывать десятки миллионов строк кода, обеспечивая 100-процентное покрытие трассы

Klocwork k7	Предназначен для автоматизированного статического анализа кода, выявления и предотвращения дефектов программного обеспечения и проблем безопасности	C/c++, java	Выявляет коренные причины недостатков качества и безопасности программного обеспечения
Codesurfer	Может применяться для поиска ошибок в исходном коде, для улучшения понимания исходного кода	C/c++	Позволяет проводить анализ указателей, использовать и определять переменные, зависимости данных, строить графы вызовов
Fxcop	Способен обнаружить более 200 недочетов (или ошибок) в следующих областях: – архитектура библиотеки; – правила именования; – производительность; – безопасность	C/c++	Откомпилированный код проверяется с помощью механизмов рефлексии, парсинга msil и анализа графа вызовов

Qaudit	Быстрый анализ исходных файлов на наличие переполнения буфера, ошибок форматной строки, запросов исполняемых вызовов, переменных среды, и функций, имеющих проблемы защиты	C/c++	Написать на интерпретируемом языке perl, прост в использовании
Российские средства выявления уязвимостей			
Ак-вс	Автоматизированный анализ исходных текстов, с целью выявления потенциально опасных сигнатур	C/c++, java, pascal, c#, php, assembler	Позволяет проводить статический анализ исходных текстов, динамический анализ, имеет базы сигнатур для каждого из поддерживаемых языков программирования
Аист-с	Автоматизированный анализ исходных текстов	C/c++	Позволяет проводить статический анализ исходных текстов
Ксайт	Автоматизированный анализ исходных текстов	C/c++	Позволяет проводить статический анализ исходных текстов
Уса	Предназначено для выявления потенциально опасных сигнатур	C/c++, pascal, perl, plm	Имеет базы сигнатур для каждого из поддерживаемых языков программирования

Viva64	Помогает отслеживать в исходном коде потенциально опасные фрагменты, связанные с переходом от 32- битных систем к 64- битным	C/c++	Помогает писать корректный и оптимизированный код для 64- битных систем
--------	--	-------	---

Задание № 2

Разработать приложение, интерфейс которого представлен на рисунке.

Коррекция

Имя:

Телефон:

Добавить

Удалить

Найти

Список:

- Валентина:333-33-33
- Василий:222-22-22
- Ирина:555-55-55
- Максим:111-11-11

Всего: 4

Задание № 3

Добавить в программу форму авторизации по имени и паролю.

Лабораторная работа №26 «Нагрузочное тестирование, стрессовое тестирование»

Цель: получение навыков проведения нагрузочного и стрессового тестирования.

Теоретические вопросы

- Особенности нагрузочного тестирования.
- Особенности стрессового тестирования.

Задание № 1

Разработать Компилятор простых арифметических выражений, например $2 + (-5) * (7 - 8)$.

Вход и выход осуществляются в виде строк.

Задание № 2

Разработать тестовый сценарий нагрузочного тестирования. Ответить на вопрос – сколько запросов в секунду может обработать приложение при условии, что они идут последовательно. Построить график зависимости времени ответа от количества параллельных запросов (рассматривать логарифмическую шкалу по основанию два, т.е. 1, 2, 4, 8, 16, 32 и т.д. запроса) Ответить на вопрос – какое максимальное количество параллельных запросов может обработать приложение без сбоев.

Задание № 3

Оформить отчет.

Лабораторная работа №27 «Тестирование интеграции» Цель: получение навыков тестирования интеграции.

Теоретические вопросы

- Особенности тестирования интеграции.
- Методы интеграционного тестирования.
- Нисходящее тестирование интеграции.
- Восходящее тестирование интеграции.
- Сравнение нисходящего и восходящего тестирования интеграции

Задание № 1

Разработать приложение, состоящее из трех модулей:

- 1) главный модуль, считывающий из текстового файла координаты точек на плоскости;
- 2) модуль, содержащий функции расчета расстояния между двумя точками;
- 3) модуль, содержащий функцию, определяющую треугольник с максимальной площадью.

Задание № 2

Описать этапы нисходящего проектирования разработанного приложения.

Задание № 3

Описать этапы восходящего проектирования разработанного приложений.

Задание № 4

Оформить отчет.

Лабораторная работа №28 «Конфигурационное тестирование» Цели: получение навыков проведения конфигурационного тестирования.

Теоретические вопросы

– Особенности конфигурационного тестирования.

Конфигурационное тестирование (Configuration testing). Проверяется работоспособность при различных конфигурациях, предполагает тестирование работы системы на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

Задание № 1

Дана структура с именем ZNAK, состоящая из полей:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, которая выполняет следующие действия:

- ввод с клавиатуры данных в массив, состоящий из 8 элементов типа ZNAK, и занесение их в файл данных;
- чтение данных из файла и вывод их на экран;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет – вывести об этом сообщение);
- список должен быть упорядочен по знакам Зодиака.

Задание № 2

Описать и обосновать итоги тестирования работы разработанного приложения на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

Лабораторная работа №29 «Тестирование установки» Цель: получение навыков тестирования установки.

Теоретические вопросы

– Комплексное тестирование приложения.

Задание №1

Разработать приложение, интерфейс которого представлен на рисунке 1.

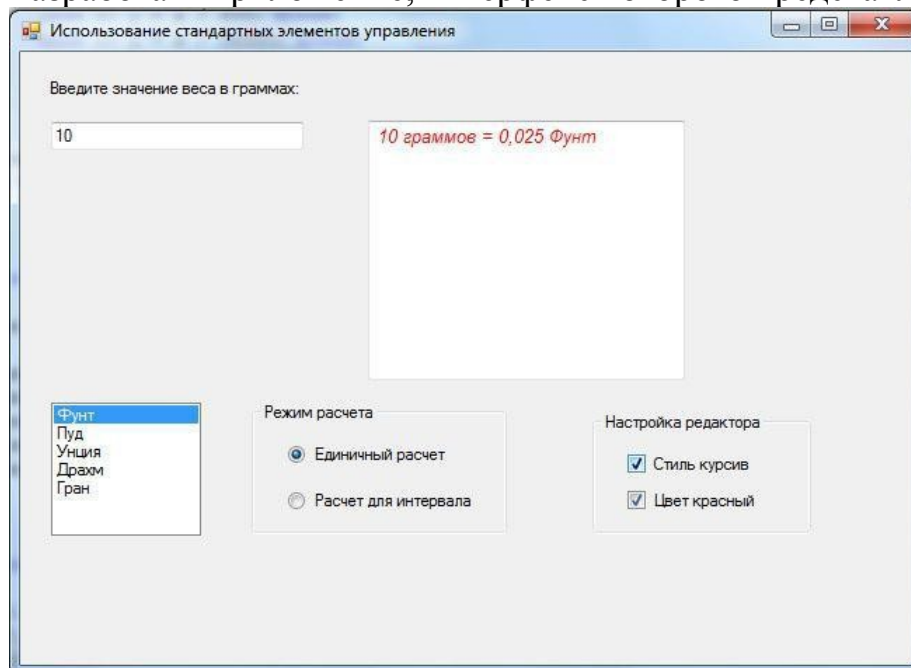


Рисунок 1. Интерфейс приложения

Задание №2

Провести комплексное тестирование разработанного приложения.

Задание №3

Оформить отчет.

Список литературы

Электронные издания (электронные ресурсы)

1. Гвоздева, В.А. Основы построения автоматизированных информационных систем [Электронный ресурс]: учебник / В.А. Гвоздева, И.Ю. Лаврентьева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 318 с. - Режим доступа: <http://znanium.com/catalog/product/989678> (ЭБСZnanium)
2. Гвоздева, В.А. Информатика, автоматизированные информационные технологии и системы [Электронный ресурс]: учебник / В.А. Гвоздева. — М.: ИД «ФОРУМ»: ИНФРА-М, 2019. — 542 с. - Режим доступа: <http://znanium.com/catalog/product/999615> (ЭБСZnanium)
3. Затонский, А.В. Информационные технологии: разработка информационных моделей и систем: учебное пособие / А.В. Затонский. — Москва: РИОР: ИНФРА-М, 2020. — 344 с. + Доп. материалы [Электронный ресурс]- Режим доступа: <http://znanium.com/catalog/product/1043097>
4. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учеб. пособие / Г.Н. Федорова. — М. :КУРС : ИНФРА-М, 2019. — 336 с.- Текст : электронный. -Режим доступа: <http://znanium.com/catalog/product/989682>
5. Языки программирования [Электронный ресурс]: учеб. пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2018. — 399 с. - Режим доступа: <http://znanium.com/bookread2.php?book=973007> (ЭБСZnanium)

Методические издания

1. Игнатенко, Е.С. Методические указания по выполнению практических работ по МДК

05.03 Тестирование информационных систем .– Нефтеюганск: НИК(филиал) ФГБОУ ВО «ЮГУ», 2019 [Электронный ресурс]
Режим доступа: локальная сеть филиала

Периодические издания

1. Программные продукты и системы [Электронный ресурс]: журнал.- Тверь: — Научно-исследовательский институт «Центрпрограммсистем» Электронно-библиотечная система «Лань» : [сайт]. — Режим доступа: <https://e.lanbook.com/journal/2276>