

ГУАП

КАФЕДРА № 23

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

профессор, д-р техн. наук,
профессор

должность, уч. степень, звание

5

 29.03.2021

подпись, дата

А.Н. Якимов

инициалы, фамилия

ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ПРАКТИЧЕСКОГО
ЗАДАНИЯ

ДИАГНОСТИРОВАНИЕ БРОНХИАЛЬНОЙ АСТМЫ

по курсу: ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ ЭС

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 2733



27.03.2021

подпись, дата

Ш.И. Мирзашарифов

инициалы, фамилия

Санкт-Петербург 2021

Цель работы

Изучить возможности использования технологии искусственного интеллекта (ИИ) в заданной предметной области.

1 Индивидуальное задание

С использованием технологии ИИ разработать систему поддержки принятия решений по диагностике бронхиальной астмы:

- а) построить продукционную модель представления знаний об объекте заданной предметной области;
- б) разработать алгоритм, реализующий прямую или обратную цепочку рассуждений продукционной модели;
- в) выполнить программную реализацию разработанного алгоритма в одном из пакетов прикладных программ.

2 Описание объекта моделирования

Болезни легких опасны и чреваты как серьезными нарушениями в состоянии человека, так и летальным исходом в запущенных состояниях. Различных заболеваний системы дыхания огромное количество, и все они имеют свои особенности. Врач и пациент должны быть очень внимательными, чтобы вовремя распознать ту или иную патологию. Почему врачи говорят, что внимательным должен быть и пациент? Потому что именно он должен распознать и верно описать врачу свои симптомы. Одного анализа жалоб недостаточно, ведь часто для разных заболеваний характерны одни и те же симптомы. Например, одышка может быть признаком как пневмонии, так и хронической обструктивной болезни легких (ХОБЛ), а также бронхиальной астмы.

Чтобы понимать, с чем приходится иметь дело, нужно разобрать характерные симптомы тех или иных патологий.

Бронхиальная астма - это хроническое воспалительное заболевание дыхательных путей, которое проявляется сужением просвета бронхов за счет трех компонентов: бронхоспазма (непроизвольного сокращения гладких мышц бронхов), отека слизистой оболочки бронхов и вязкого бронхиального секрета. Заболеванию характерны такие симптомы как: свистящее дыхание, дистанционные хрипы; пониженная температура; сочетание кашля и одышки с удлиненным выдохом; заложенность в груди; приступообразный кашель преимущественно ночью или под утро, заканчивающийся отделением прозрачной (стекловидной) вязкой, скудной мокроты.

Пневмония - это знак того, что в организме уже есть проблемы, связанные с патологией других органов. Чаще всего это болезни сердечно-сосудистой и дыхательной

систем, наличие онкологических заболеваний, хронической алкогольной интоксикации и др., т. е. пациент страдает коморбидной патологией. Болезнь имеет следующие симптомы: повышение температуры тела до высоких цифр — выше 38 °С (не обязательно); кашель кратковременно непродуктивный, сменяющийся продуктивным; гнойная или «ржавая» мокрота; одышка; боль в грудной клетке.

Бронхит — заболевание бронхов воспалительного характера. Симптомами болезни являются: сухой кашель (надсадный и приступообразный), сменяющийся влажным; мокрота зеленоватого и желтоватого цвета свидетельствует в пользу бактериальной инфекции, а белая или прозрачная слизь — об ее отсутствии; температура 38°С (отмечалась хотя бы один раз); боль за грудиной и между ребрами, усиливающаяся при кашле; появление одышки при наличии бронхообструкции.

ХОБЛ - это прогрессирующее заболевание, характеризующееся воспалительным компонентом, нарушением бронхиальной проходимости на уровне дистальных бронхов и структурными изменениями в легочной ткани и сосудах. К симптомам заболевания относятся: свистящее дыхание; сочетание кашля и одышки с удлиненным выдохом; выделение мокроты любого характера.

3 Продукционная модель объекта моделирования

Продукционная модель построена в виде дерева решений (оргафа), который представлен на рисунке 1. Диаграмма состоит из кружков и прямоугольников, которые называются вершинами. Каждой вершине присваивается номер. На вершины можно ссылаться по этим номерам. Линии, соединяющие вершины, называются дугами или ветвями. Кружки, содержащие вопросы, называются вершинами решений. Прямоугольники содержат цели диаграммы и означают логические выводы. Линии показывают направление диаграммы. Многие вершины имеют сразу по несколько ветвей, связывающих их с другими вершинами. Выбор выходящей из вершины ветви определяется проверкой условия, содержащегося в вершине.

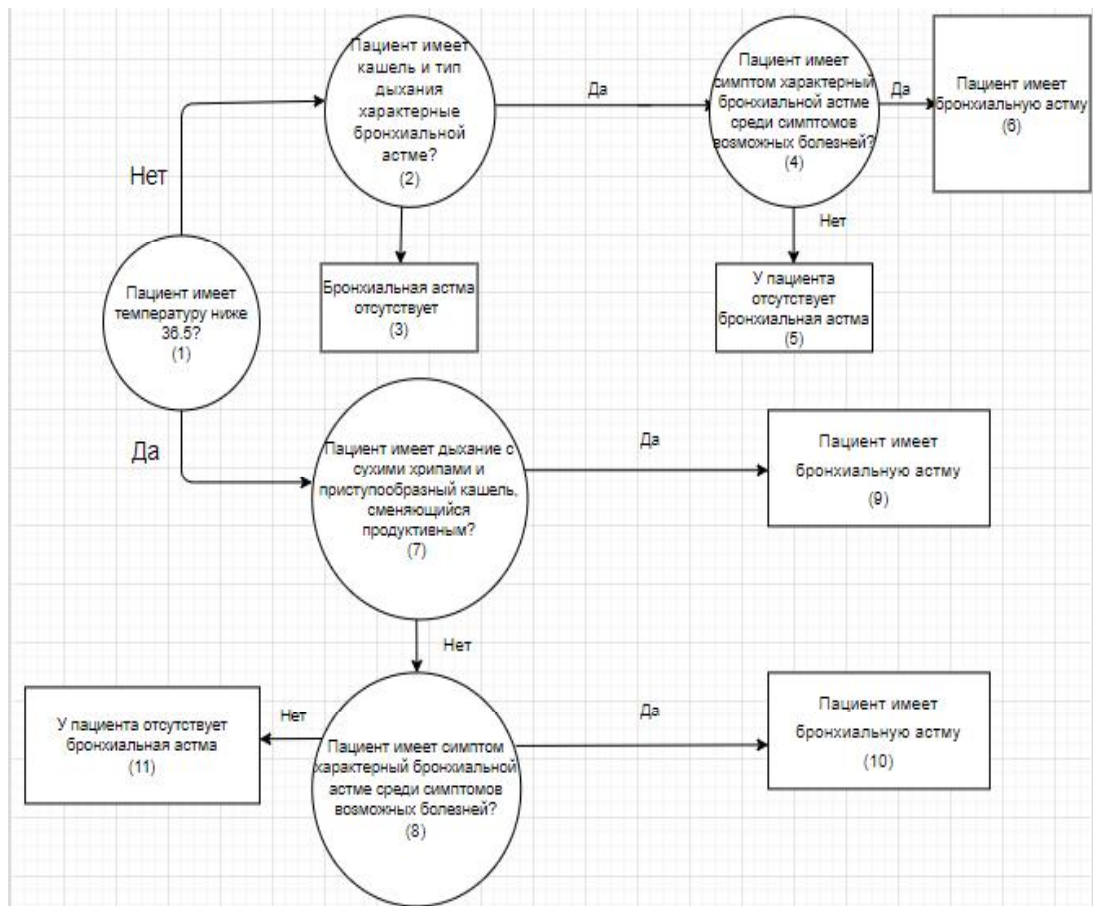


Рисунок 1 – Дерево решений (орграф) для определения болезни

4 Алгоритм, реализующий цепочку рассуждений продукционной модели

Упрощенный алгоритм реализующий прямую цепочку рассуждений представлен на рисунке 2.

Здесь:

A – есть жалобы на здоровье;

S – есть симптомы легочной болезни;

C – температура за пределами $0,5^{\circ}\text{C}$ от нормы;

H – отсутствует ХОБЛ;

B – отсутствует бронхит;

P – отсутствует пневмония;

R – имеется бронхиальная астма.

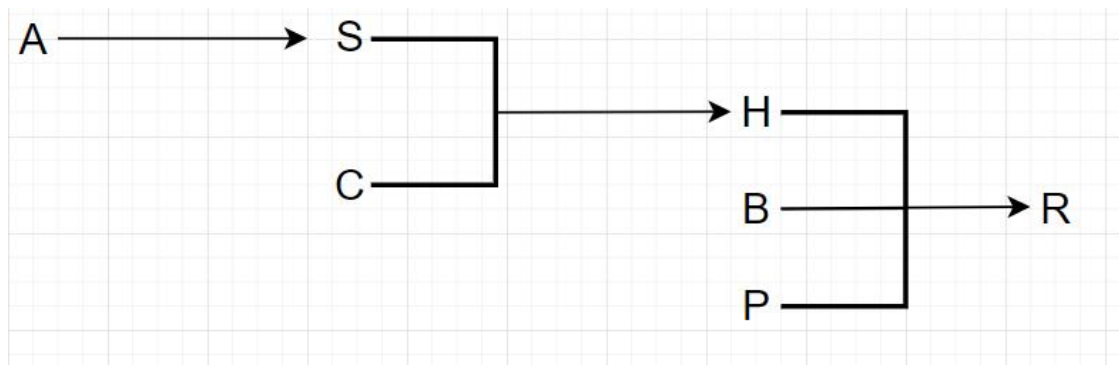


Рисунок 2 – Использование правил продукции

1. Если есть жалобы на здоровье (A) то есть симптомы легочной болезни (S).
2. Если есть симптомы легочной болезни и температура за пределами $0,5^{\circ}\text{C}$ от нормы (C) то отсутствует ХОБЛ.
3. Если отсутствует ХОБЛ (H), отсутствует бронхит (B) и отсутствует пневмония (P), то у пациента бронхиальная астма (R).

Прямой способ рассуждений представлен на рисунке 3. В нем по известным фактам отыскивается заключение, которое следует из этих фактов и накапливается рабочая память.

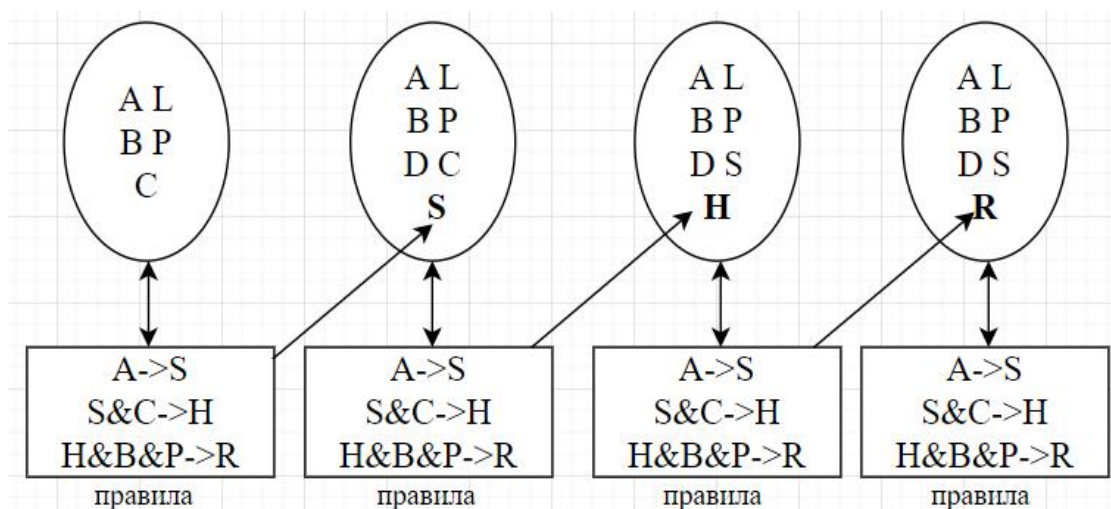


Рисунок 3 – Прямая цепочка рассуждений

На шаге 1 по факту A отыскивается S, выполняется первое правило $A \rightarrow S$, S помещается в рабочую память.

На шаге 2 выполняется второе правило $S \& C \rightarrow H$, факт H помещается в рабочую память.

На шаге 3 выполняется третье правило $H \& B \& P \rightarrow R$, вследствие этого R заносится в рабочую память. Так как R является целью, то поиск заканчивается.

5 Программная реализация разработанного алгоритма

Программа реализована в объектно-ориентированном стиле программирования, в основном на языке PHP. В качестве соединения с базой данных используется MySQL.

Файловая система включает следующие папки с содержимыми.

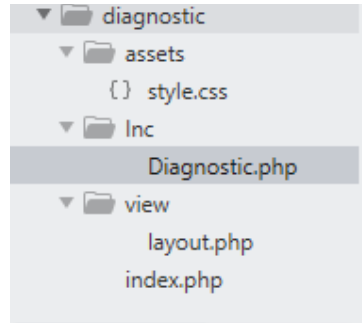


Рисунок 4 – Файловая система программы

В папке `assets` расположен файл `style.css`, в нем хранятся стили – цвет и размеры шрифта, фон, отступы, размеры блоков и тд.

В папке `Inc` расположен файл `Diagnostic.php` с соответствующим содержимым классом.

Свойства и методы класса приведены на рисунках 5-7.

Метод `__construct` – устанавливает связь с базой данных.

Метод `formTest` – формирует тест, для выбора температуры, типа дыхания кашля, параметром принимает массив, в котором ключами являются названия симптомов, значениями – столбцы в таблице базы данных – рисунок 9.

Метод `getIllness` – определяет названия болезней, на основании, выбранных симптомов.

```

<?php
namespace Inc;
class Diagnostic{
    public $link;
    protected $illness = [];

    public function __construct(){
        session_start();
        $this->link = mysqli_connect('diagnostic', 'root', '', 'diagnostic');
    }

    public function formTest($columns){
        $part = "<form method=\"POST\">";
        foreach($columns as $title=>$column){
            $data = $this->getData('illness', $column);
            $part .= "<h3>$title</h3>";
            $part .= "<select name=\"$column\">";
            foreach($data as $elem){
                $part .= "<option>$elem[$column]</option>";
            }
            $part .= "</select>";
        }
        $part .= "<input type=\"submit\" value=\"Диагноз\" name=\"send\">";
        $part .= "</form>";
        return $part;
    }

    public function getIllness($columns){
        $part = [];
        foreach($columns as $title=>$column){
            $illness = $this->getData('illness', 'name', "WHERE $column='$_POST[$column]'");
            $part[] = $illness['name'];
        }
        $this->illness = array_unique($part);
        $_SESSION['illness'] = $this->illness;
        return $this->result($this->illness);
    }
}

```

Программная
реализация!

Рисунок 5 – Содержание методов formTest и getIllness, класса Diagnostic

Метод decide – запускается первым после метода-конструктора и выбирает метод, который необходимо запустить на основе входных данных.

Метод result – запускается в метода getIllness, параметром принимает массив возможных болезней, если болезнь всего одна, то выдает заключение, если больше одной, то вызывает метода formResultTest – рисунок 7.

```

41 public function decide($columns){
42     if(isset($_POST['send'])){
43         $test = $this->getIllness($columns);
44     }elseif(!isset($_POST['send']) AND !isset($_POST['final']) and !isset($_POST['again'])){
45         $test = $this->formTest($columns);
46     }elseif(isset($_POST['final'])){
47         $test = "Имеющаяся болезнь:". "<br><p class='diag'>$_POST[sick]</p><br>
48             <a href='\"/\"'>Пройти тест заново</a>";
49         $_SESSION['illness'] = null;
50     }elseif(isset($_POST['again'])){
51         $test = $this->formResultTest($_SESSION['illness']);
52     }
53 }
54 return $test;
55 }
56
57 private function result($illness){
58     if(count($illness) == 1){
59         $part = "Имеющаяся болезнь:". "<br><p class='diag'>$_illness[0]</p><br>
60             <a href='\"/\"'>Пройти тест заново</a>";
61     }elseif(count($illness) > 1){
62         $part = $this->formResultTest($illness);
63     }
64 }
65 return $part;
66 }
67 }

```

Рисунок 6 – Содержание метода decide и result, класса Diagnostic

Метод formResultTest – вспомогательный, запускается, если количество возможных болезней оказалось больше одной, метод выбирает по одному случайному симптому к каждой возможной болезни и составляет финальный тест, после которого выдается заключение.

Метод getData – вспомогательный, формирует запросы к базе данных, параметрами принимает имя таблицы, имя столбца и условие, возвращает массив с данными.

```

68 private function formResultTest($illness){
69     $part = "Возможные болезни:". implode(', ', $illness). "<br>
70     <form method='POST'>";
71     foreach($illness as $sick){
72         $query = "SELECT other_symptoms.test as 'test'
73             FROM illness
74             LEFT JOIN rel ON rel.sick_id=illness.id
75             LEFT JOIN other_symptoms ON rel.symptom_id=other_symptoms.id WHERE illness.name='$_sick' ORDER BY RAND() LIMIT 1;";
76         $result = mysql_query($this->link, $query) or die(mysql_error($this->link));
77         $data = mysql_fetch_assoc($result);
78         $part .= "<label for='$_sick'><input type='radio' value='$_sick' name='$_sick' id='$_sick'> $data[test]
79             </label><br>";
80     }
81     $part .= "<input type='submit' name='final' value='Диагноз'>
82     </form>";
83     $part .= "<form method='POST'>
84     <input type='submit' name='again' value='Ничего из симптомов не подходит'>
85     </form>";
86     return $part;
87 }
88
89 private function getData($table, $column = '*', $where = ''){
90     $query = "SELECT $column FROM $table $where";
91     $result = mysql_query($this->link, $query) or die(mysql_error($this->link));
92     if($where == null){
93         for($data = []; $row = mysql_fetch_assoc($result); $data[] = $row);
94     }else{
95         $data = mysql_fetch_assoc($result);
96     }
97     return $data;
98 }
99 }

```

Рисунок 7 – Содержание методов formResultTest и getData, класса Diagnostic

Файл index.php является точкой входа, именно в нем создается экземпляр класса Diagnostic и вызывает его метод decide, содержание которого представлено на рисунке 6.

```

1  <?php
2  require_once 'Inc/Diagnostic.php';
3  use Inc\Diagnostic;
4
5  $array = [
6      'Температура'=>'temperature',
7      'Тип дыхания'=>'breathing',
8      'Кашель'=>'cough'
9  ];
10
11  $test = (new Diagnostic())->decide($array);
12
13  include 'view/layout.php';
14

```

Рисунок 8 – Создание экземпляра класса Diagnostic и вызов метода decide
Таблицы с симптомами и названиями болезней приведены на рисунках 9-11.

id	name	temperature	breathing	cough
1	Бронхиальная астма	Пониженная	С свистящим хрипом	Приступообразный кашель преимущественно ночью или
2	Пневмония	От 37°C и выше, держится более 5 суток или выше	Слабыми хрипами	Кашель, кратковременный непродуктивный, слизистый...
3	Бронхит	Хотя бы раз отмечалось около 38°C	Жесткий	Кашель сухой (надсадный и приступообразный), длительный
4	ХОБЛ	В норме	Свистящий	Кашель смешанный, чаще в первой половине дня, ред...

Рисунок 9 – Таблица болезней в базе данных с базовыми симптомами

id	text
1	Приступ удушья, который провоцируется воздействием...
2	Сочетание кашля и одышки с удлиненным выдохом
3	Заложенность в груди
4	Гнойная или «ржавая» мокрота
5	Одышка
6	Боль в грудной клетке
7	Слабость, недомогание, головная боль
8	Появление одышки при наличии бронхообструкции
9	Выделение мокроты любого характера
10	Одышка постоянная, нарастающая при физической нагр...
11	Боль за грудиной и между ребрами, усиливающаяся пр...

Рисунок 10 – Таблица other_symptoms в базе данных для точного выявления болезни

id	sick_id	symptom_id
1	1	1
2	1	2
3	1	3
4	2	4
5	2	5
6	2	6
7	3	7
8	3	8
9	4	9
10	4	10
11	3	11

Рисунок 11 – Таблица rel связывающая таблицу Illness и other_symptoms
Заставка ИИ для диагностирования бронхиальной астмы представлена на рисунке

12.

Document

Небезопасно | diagnostic/ На экран

Diagnostic - ГУАП 2733

Температура
Пониженная

Тип дыхания
С трудом при вдохе

Кашель
Пароксизмальный кашель преимущественно ночью или под утро, связанный с выделением прозрачной (беловатой) слизи, сухой ночью

Диагноз

Рисунок 12 –Заставка ИИ для диагностирования бронхиальной астмы
Результат работы программы с заключением, представлен на рисунке 13.

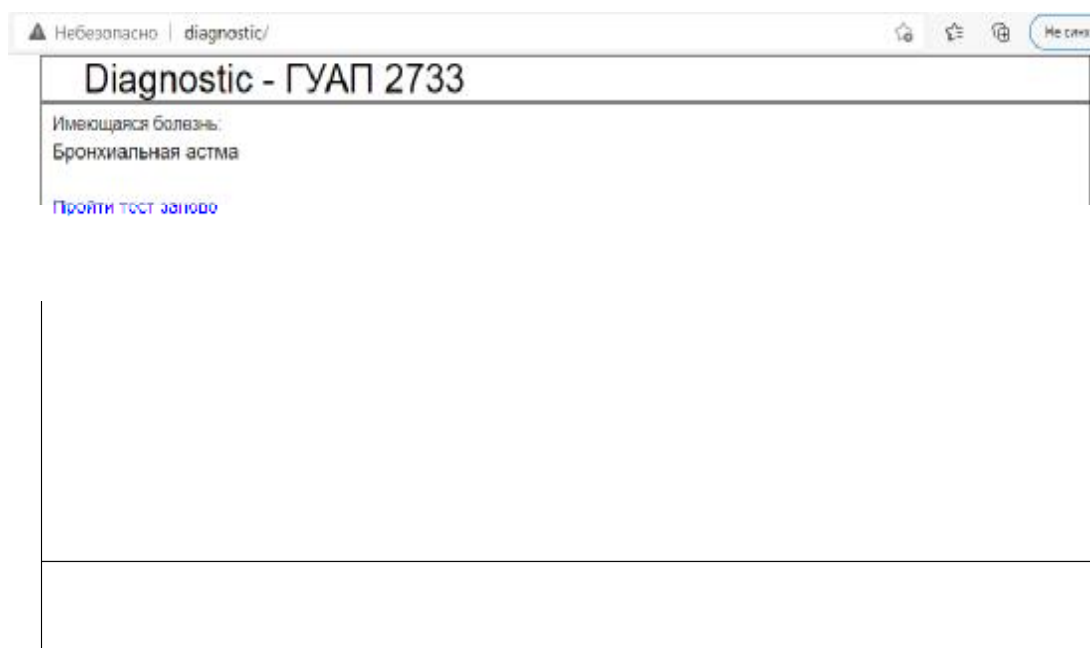


Рисунок 13 – Результат работы программы.

Выводы

Система с использованием продукционной модели позволяет наглядно представить логические выводы в предметной области, с помощью орграфа, с таким представлением реализация программы требует меньше времени. Орграф поддерживает вершины решений, которые с помощью логических высказываний ссылаются на логические выводы, что позволяет добавлять новые вершины решений и логические выводы или модифицировать имеющиеся, не затронув при этом остальные, проявляется такое преимущество продукционных моделей, как - модульность.

Литература

- 1.Интеллектуальные системы конструкторско-технологического проектирования: лабораторный практикум для студ. вузов / А.Н. Якимов А.Н., Бестугин А.Р., Киршина И.А. – СПб.: ГУАП, 2020. – 49 с.
- 2.Зандстра М. /РНР: объекты, шаблоны и методики программирования, 4-е изд. Пер. с англ. -М.: ООО "И.Д. Вильямс", 2015. - 576 с.
- 3.Базы данных : учеб. пособие для студ. учреждений сред, проф. образования / Фуфаев Э. В., Фуфаев Д. Э. — 7-е изд., стер. — М.: Издательский центр «Академия», 2012. — 320 с.
4. Моргунов, Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Моргунов Е. П.; под ред. Рогова Е. В., Лузанова П. В. — СПб.: БХВ-Петербург, 2018. — 336 с.