

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ПРИАМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ШОЛОМ-АЛЕЙХЕМА»

Р.И.Баженов, О.Ю.Пронина

КАК СТУДЕНТУ ЗАРЕГИСТРИРОВАТЬ СОБСТВЕННУЮ ПРОГРАММУ ДЛЯ ЭВМ В РОСПАТЕНТЕ

Учебно-методическое пособие

Биробиджан
2017

УДК 001.92:37
ББК 72.4 я7
К16

Рецензент

Штепа Ю.П., кандидат педагогических наук, доцент, доцент кафедры информационных систем, математики и методики обучения, Приамурский государственный университет им.Шолом-Алейхема.

Как студенту зарегистрировать собственную программу для ЭВМ в Роспатенте/ сост. Р.И.Баженов, О.Ю.Пронина. – Биробиджан: ИЦ ПГУ им. Шолом-Алейхема, 2017. – 65 с.

В учебно-методическом пособии рассматриваются технология регистрации разработанной компьютерной программы в Роспатенте.

Пособие адресовано студентам всех направлений, может использоваться при изучении курса «Основы научных исследований».

Пособие публикуется в рамках мероприятия «Эльбрус науки» Программы развития деятельности студенческих объединений «Поехали!» ПГУ им. Шолом-Алейхема, поддержанной Министерством образования и науки Российской Федерации в 2017 году.

УДК 001.92:37
ББК 72.4 я7
К16

© Баженов Р.И.,Пронина О.Ю. 2017
© ФГБОУ ВО «ПГУ им. Шолом-Алейхема», 2017

Оглавление

1. Как студенту зарегистрировать собственную программу для ЭВМ в Роспатенте.....	4
2 Примеры документов	6
2.1 Заявление (лицевая часть)	6
2.2 Заявление (оборотная часть).....	7
2.3 Заявление. Дополнение (лицевая часть)	9
2.4 Заявление. Дополнение (оборотная часть)	10
2.5 Реферат.....	12
2.6 Материалы аудиовизуальных отображений, порождаемых программой для ЭВМ	13
2.7 Исходный текст программы	18
2.8 Согласие автора на указание сведений об авторе, указанных в заявлении	53
2.9 Согласие на обработку персональных данных.....	55
2.10 Сопроводительное письмо	56
2.11 Справочная информация.....	57
Список литературы.....	64

1. Как студенту зарегистрировать собственную программу для ЭВМ в Роспатенте

Когда студент совместно с преподавателем занимается научными исследованиями, то часто создаются различные компьютерные программы. Потом возникает вопрос: «Как защитить свою интеллектуальную собственность?».

Для решения этого вопроса можно поступить следующим образом – зарегистрировать разработанную программу в Роспатенте.

Авторское право на программу уже зафиксировано при ее создании, поэтому регистрация в Роспатенте позволяет ее продавать и показать юридически, что программа принадлежит авторам.

Руководство вузов положительно относиться к регистрации программ для ЭВМ, поэтому необходимо вместе с руководителем создать необходимые документы и передать их в соответствующий отдел вуза.

Требуется подготовить:

1. Заявление;
2. Дополнение к заявлению;
3. Распечатка исходного текста программы;
4. Реферат;
5. Материалы аудиовизуальных отображений, порождаемых программой для ЭВМ;
6. Согласие на обработку персональных данных;

7. Согласие авторов на указание сведений об авторах, указанных в заявлении;

8. Платежный документ об уплате госпошлины.

После создания документов, необходимо их записать на диск и все вместе отдать в соответствующий отдел вуза (отдел патентования или управление научной деятельностью).

Авторы имеют опыт регистрации нескольких программ и предоставили примеры из своих разработок.

Прочую справочную информацию можно посмотреть на сайтах Роспатента и Федерального института промышленной собственности, указанных в списке литературы.

3. РЕГИСТРИРУЕМЫЙ ОБЪЕКТ	
<input checked="" type="checkbox"/> Не содержит персональные данные	
<input type="checkbox"/> Содержит персональные данные Регистрационный номер _____ в Реестре операторов, осуществляющих обработку персональных данных	
4. ГОД СОЗДАНИЯ РЕГИСТРИРУЕМОЙ ПРОГРАММЫ ДЛЯ ЭВМ ИЛИ БАЗЫ ДАННЫХ <u>2017</u>	
5. СТРАНА И ГОД ОБНАРОДОВАНИЯ (ПЕРВОГО ОПУБЛИКОВАНИЯ) РЕГИСТРИРУЕМОЙ ПРОГРАММЫ ДЛЯ ЭВМ ИЛИ БАЗЫ ДАННЫХ:	
Страна: _____	Нет Год: _____
5А. ГОД ОБНОВЛЕНИЯ РЕГИСТРИРУЕМОЙ БАЗЫ ДАННЫХ, государственная регистрация которой осуществляется в соответствии с пунктом 3 статьи 1334 Кодекса <u>Нет</u>	
6. Представленная на регистрацию база данных зарегистрирована в Реестре баз данных как:	
<input type="checkbox"/> База данных, государственная регистрация которой осуществляется в соответствии с пунктом 4 статьи 1259 Кодекса. Рег. № _____ от _____	
<input type="checkbox"/> База данных, государственная регистрация которой осуществляется в соответствии с пунктом 3 статьи 1334 Кодекса. Рег. № _____ от _____ <input checked="" type="checkbox"/> нет	

2.2 Заявление (оборотная часть)

7. АВТОРЫ Всего авторов: <u>3</u> <input type="checkbox"/> авторы отказались быть упомянутыми в качестве таковых	
7А. СВЕДЕНИЯ ОБ АВТОРЕ:	
Фамилия имя отчество: Седова Нелли Алексеевна	
Дата рождения: число: 27 месяц: 11 год: 1900 Гражданство: Россия	
Автор согласен с обработкой указанных персональных данных в объеме действий, предусмотренных предоставляемой государственной услугой, и в течение срока действия исключительного права на регистрируемый объект	
Место жительства, включая указание страны: 690001, Российская Федерация, Приморский край, г. Владивосток, ул.Статса, д.111, кв. 33 (RU)	
Краткое описание творческого вклада автора при создании регистрируемой программы для ЭВМ или базы данных: Написание программного комплекса, отладка программного комплекса	
При публикации сведений о государственной регистрации программы для ЭВМ или базы данных автор просит: (отметить [X])	
<input checked="" type="checkbox"/> упоминать его под своим именем <input type="checkbox"/> не упоминать его (анонимно)	
<input type="checkbox"/> упоминать его под псевдонимом: _____	
8. СВЕДЕНИЯ О ПЛАТЕЛЬЩИКЕ (указываются полное имя физического лица или наименование юридического лица) ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема»	
<input type="checkbox"/> Для физического лица Серия и номер документа, удостоверяющего личность:	<input checked="" type="checkbox"/> для юридического лица ИНН: 7901009072 КПП: 790101001 КИО:
ИНН:	СНИЛС:

9. СВЕДЕНИЯ О СОДЕРЖАЩИХСЯ В ЗАЯВКЕ ДОКУМЕНТАХ (отметить [X])

<input checked="" type="checkbox"/> идентифицирующие программу для ЭВМ материалы в форме распечатки исходного текста	на 19 л. в 1 экз.
<input type="checkbox"/> идентифицирующие программу для ЭВМ материалы в иной форме	на л. в 1 экз.
<input checked="" type="checkbox"/> материалы аудиовизуальных отображений, порождаемых программой для ЭВМ	на 7 л. в 1 экз.
<input type="checkbox"/> материалы, идентифицирующие базу данных	на л. в 1 экз.
<input type="checkbox"/> документы, подтверждающие существенные затраты на создание базы данных	на л. в 1 экз.
<input checked="" type="checkbox"/> реферат	на 1 л. в 2 экз.
<input type="checkbox"/> иные материалы:	на л. в 1 экз.
<input checked="" type="checkbox"/> дополнение к заявлению	на 1 л. в 1 экз.
<input type="checkbox"/> доверенность(и)	на л.
<input checked="" type="checkbox"/> документ(ы) об уплате государственной пошлины (представляется по инициативе заявителя)	на 1 л. в 1 экз.
<input type="checkbox"/> документ, подтверждающий наличие оснований для освобождения от уплаты государственной пошлины либо для уменьшения ее размера, либо для отсрочки ее уплаты	на л. в 1 экз.
<input type="checkbox"/> ходатайство(а)	на л. в 1 экз.
<input checked="" type="checkbox"/> иные документы (указать наименование документа) Приложения №3 и № 4	на 6 л. в 1 экз.

10. КОНТАКТНЫЕ РЕКВИЗИТЫ ДЛЯ ПРЕДСТАВЛЕНИЯ ТРЕТЬИМ ЛИЦАМ (тел., адрес электронной почты и др.):**rectorat@pgusa.ru**

11. Заявителю известно, что в соответствии с подпунктом 4 пункта 1 статьи 6 Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных» Федеральная служба по интеллектуальной собственности осуществляет обработку персональных данных субъектов персональных данных, указанных в заявлении, в целях и объеме, необходимых для предоставления государственной услуги. Заявитель настоящим подтверждает, что имеет согласия субъектов персональных данных, указанных в заявлении (за исключением представителя), на обработку их персональных данных, приведенных в настоящем заявлении, в Федеральной службе по интеллектуальной собственности в связи с предоставлением государственной услуги. Согласия оформлены в соответствии со статьей 9 Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных».

12. ПОДПИСЬ(И) ЗАЯВИТЕЛЯ(ЕЙ) (ПРАВООБЛАДАТЕЛЯ(ЕЙ) ИЛИ ЕГО (ИХ) ПРЕДСТАВИТЕЛЯ(ЕЙ))**Ректор ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема»****Н.Г. Баженова**М.П.**«14» сентября 2017 г.**

(от имени юридического лица заявление подписывается руководителем организации или иным лицом, уполномоченным на это в установленном законодательством Российской Федерации порядке, с указанием его должности и скрепляется печатью юридического лица при наличии печати)

Подпись любого лица должна быть расшифрована с указанием фамилии и инициалов и указана дата подписания заявления.

Правообладатели – физические лица подтверждают согласие с обработкой указанных в данном заявлении персональных данных в целях и объеме, необходимых для предоставления государственной услуги, и в течение срока действия исключительного права на регистрируемый объект)

2.3 Заявление. Дополнение (лицевая часть)

ВХОДЯЩИЙ НОМЕР ВК _____ Дата поступления: ____ . ____ . 20 ____	РЕГИСТРАЦИОННЫЙ НОМЕР ПРОГРАММЫ ДЛЯ ЭВМ или БАЗЫ ДАННЫХ в РЕЕСТРЕ _____ Дата регистрации: ____ . ____ . 20 ____	РЕГИСТРАЦИОННЫЙ номер заявки _____ Дата поступления: ____ . ____ . 20 ____
Дополнение к 3 А Я В Л Е Н И Ю о государственной регистрации (отметить [X]): <input checked="" type="checkbox"/> Программы для ЭВМ <input type="checkbox"/> Базы данных, государственная регистрация которой осуществляется в соответствии с пунктом 4 статьи 1259 Кодекса <input type="checkbox"/> Базы данных, государственная регистрация которой осуществляется в соответствии с пунктом 3 статьи 1334 Кодекса		В Федеральную службу по интеллектуальной собственности Бережковская наб., д. 30, корп.1, г. Москва, Г-59, ГСП-3, 125993
Представляя указанные ниже документы, заявитель подтверждает отсутствие в регистрируемом объекте сведений, составляющих государственную тайну, и предоставляет Роспатенту право на воспроизведение материалов, идентифицирующих регистрируемый объект, на бумажном носителе и путем записи на электронном носителе, в том числе записи в память ЭВМ, в целях, необходимых для предоставления государственной услуги		
1. НАЗВАНИЕ ПРЕДСТАВЛЕННОЙ НА РЕГИСТРАЦИЮ ПРОГРАММЫ ДЛЯ ЭВМ ИЛИ БАЗЫ ДАННЫХ «Программный комплекс по классической теории множеств»		
2. ПРАВООБЛАДАТЕЛЬ (ЗАЯВИТЕЛЬ(И)) ОГРН: <u>1027900508195</u> ИНН: <u>7901009072</u> Серия и номер документа, удостоверяющего личность _____ СНИЛС: _____ <p style="text-align: center;">Федеральное государственное бюджетное образовательное учреждение высшего образования «Приамурский государственный университет имени Шолом-Алейхема» 679015, Российская Федерация, Еврейская автономная область, г. Биробиджан, ул. Широкая, д. 70а (RU)</p> <i>(указываются фамилия, имя, отчество (последнее – при наличии) и место жительства (для физического лица) или наименование юридического лица (согласно учредительным документам) и место нахождения (для юридического лица), включая название страны. Данные о месте жительства автора (ов)-заявителя(ей) приводятся в графе 7А)</i> <input type="checkbox"/> регистрируемый объект создан за счет средств федерального бюджета Указанное лицо является: <input type="checkbox"/> государственным заказчиком <input type="checkbox"/> муниципальным заказчиком исполнитель работ: <input type="checkbox"/> исполнителем работ по <input type="checkbox"/> государственному контракту <input type="checkbox"/> муниципальному контракту заказчик работ: Контракт от: _____ 20 ____ № _____ <i>(заполняется только в случае создания программы для ЭВМ или базы данных по государственному или муниципальному контракту)</i>		
7А. СВЕДЕНИЯ ОБ АВТОРЕ (Дополнение к графе 7А Заявления) Фамилия имя отчество: Седов Виктор Александрович Дата рождения: число: 04 месяц: 12 год: 1900 Гражданство: Россия Автор согласен с обработкой указанных персональных данных в объеме действий, предусмотренных предоставляемой государственной услугой, и в течение срока действия исключительного права на регистрируемый объект		

<p>Место жительства, включая указание страны: 690001, Российская Федерация, Приморский край, г. Владивосток, ул. Бороды, д. 233, кв. 11 (RU)</p>
<p>Краткое описание творческого вклада автора в создание регистрируемой программы для ЭВМ или базы данных: Разработка интерфейса программного комплекса, тестирование программного комплекса</p>
<p>При публикации сведений о государственной регистрации программы для ЭВМ или базы данных автор согласен: (отметить [X])</p> <p><input checked="" type="checkbox"/> упоминать его под своим именем <input type="checkbox"/> не упоминать его (анонимно)</p> <p><input type="checkbox"/> упоминать его под псевдонимом _____</p>

2.4 Заявление. Дополнение (оборотная часть)

<p>7А. СВЕДЕНИЯ ОБ АВТОРЕ (Дополнение к графе 7А Заявления)</p> <p>Фамилия имя отчество: Баженов Руслан Иванович</p> <p>Дата рождения: число: 07 месяц: 09 год: 1970 Гражданство: Российская Федерация</p> <p>Автор согласен с обработкой указанных персональных данных в объеме действий, предусмотренных предоставляемой государственной услугой, и в течение срока действия исключительного права на регистрируемый объект</p>
<p>Место жительства, включая указание страны: 679015, Российская Федерация, Еврейская автономная область, г.Биробиджан, ул. Широкая, д. 70а, кв. 305 (RU)</p>
<p>Краткое описание творческого вклада автора в создание регистрируемой программы для ЭВМ или базы данных: Постановка задачи, разработка интерфейса программного комплекса</p>
<p>При публикации сведений о государственной регистрации программы для ЭВМ или базы данных автор согласен: (отметить [X])</p> <p><input checked="" type="checkbox"/> упоминать его под своим именем <input type="checkbox"/> не упоминать его (анонимно)</p> <p><input type="checkbox"/> упоминать его под псевдонимом _____</p>
<p>7А. СВЕДЕНИЯ ОБ АВТОРЕ (Дополнение к графе 7А Заявления)</p> <p>Фамилия имя отчество:</p> <p>Дата рождения: число: месяц: год: Гражданство:</p> <p>Автор согласен с обработкой указанных персональных данных в объеме действий, предусмотренных предоставляемой государственной услугой, и в течение срока действия исключительного права на регистрируемый объект</p>
<p>Место жительства, включая указание страны:</p>
<p>Краткое описание творческого вклада автора в создание регистрируемой программы для ЭВМ или базы данных:</p>
<p>При публикации сведений о государственной регистрации программы для ЭВМ или базы данных автор согласен: (отметить [X])</p> <p><input type="checkbox"/> упоминать его под своим именем <input type="checkbox"/> не упоминать его (анонимно)</p> <p><input type="checkbox"/> упоминать его под псевдонимом _____</p>

Дополнение к графе № _____ Заявления

Дополнение к графе № _____ Заявления

12. ПОДПИСЬ(И) ЗАЯВИТЕЛЯ (ЕЙ) (ПРАВООБЛАДАТЕЛЯ(ЕЙ) ИЛИ ЕГО (ИХ) ПРЕДСТАВИТЕЛЯ(ЕЙ))

Ректор ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема»

Н.Г. Баженова

М.п.

«14» сентября 2017 г.

(от имени юридического лица заявление подписывается руководителем организации или иным лицом, уполномоченным на это в установленном законодательством Российской Федерации порядке, с указанием его должности и скрепляется печатью юридического лица при наличии печати.

Подпись любого лица должна быть расшифрована с указанием фамилии и инициалов с указанием даты подписания заявления.

Правообладатели – физические лица подтверждают согласие с обработкой указанных в данном заявлении персональных данных в целях и объеме, необходимых для предоставления государственной услуги, и в течение срока действия исключительного права на регистрируемый объект)

2.5 Реферат

РЕФЕРАТ

Авторы: Седова Нелли Алексеевна, Седов Виктор Александрович,
Баженов Руслан Иванович

Правообладатель: Федеральное государственное бюджетное образовательное учреждение высшего образования "Приамурский государственный университет имени Шолом-Алейхема"

Программа: Программный комплекс по классической теории множеств

Аннотация: Программный комплекс предназначен для составления практических, индивидуальных и контрольных заданий по классической теории множеств, являющейся одним из подразделов дисциплины «Дискретная математика». Программный комплекс функционирует в двух режимах: в первом режиме возможен ввод формулы в виде обратной польской записи пользователем, во втором – автоматическая генерация формулы. Функционал программного комплекса включает возможность усложнения или упрощения вводимых пользователем или сгенерированных автоматически формул, а также визуализация формулы в виде неориентированного дерева с учётом приоритета операций над множествами.

Тип ЭВМ: ПК с процессором Pentium 333 и выше.

Язык: Delphi

ОС: Windows 98 / 2000 / XP/7 и выше.

Объем программы: 601 Кб

2.6 Материалы аудиовизуальных отображений, порождаемых программой для ЭВМ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

"Приамурский государственный университет имени Шолом-Алейхема"

Материалы аудиовизуальных отображений, порождаемых
программой для ЭВМ

Программный комплекс по классической теории множеств

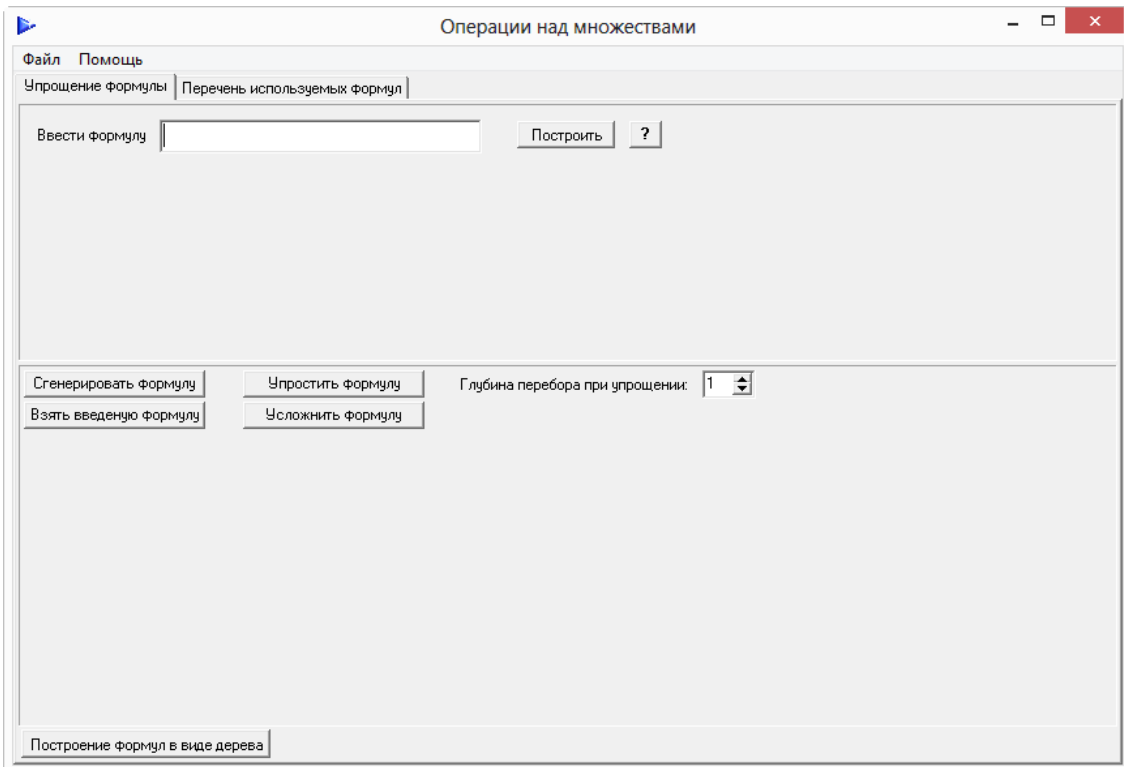
Листов 05

Авторы: Н.А.Седова,
В.А. Седов,
Р.И. Баженов

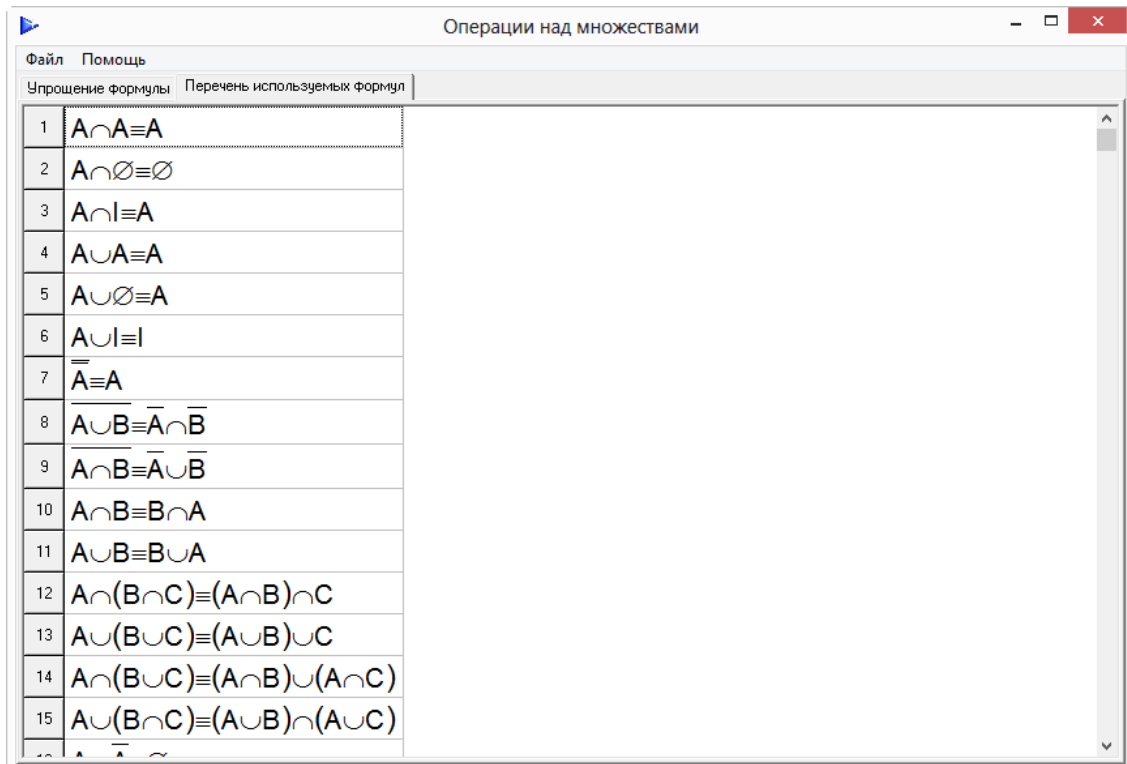
© ПГУ им. Шолом-Алейхема, 2017

Биробиджан 2017 г.

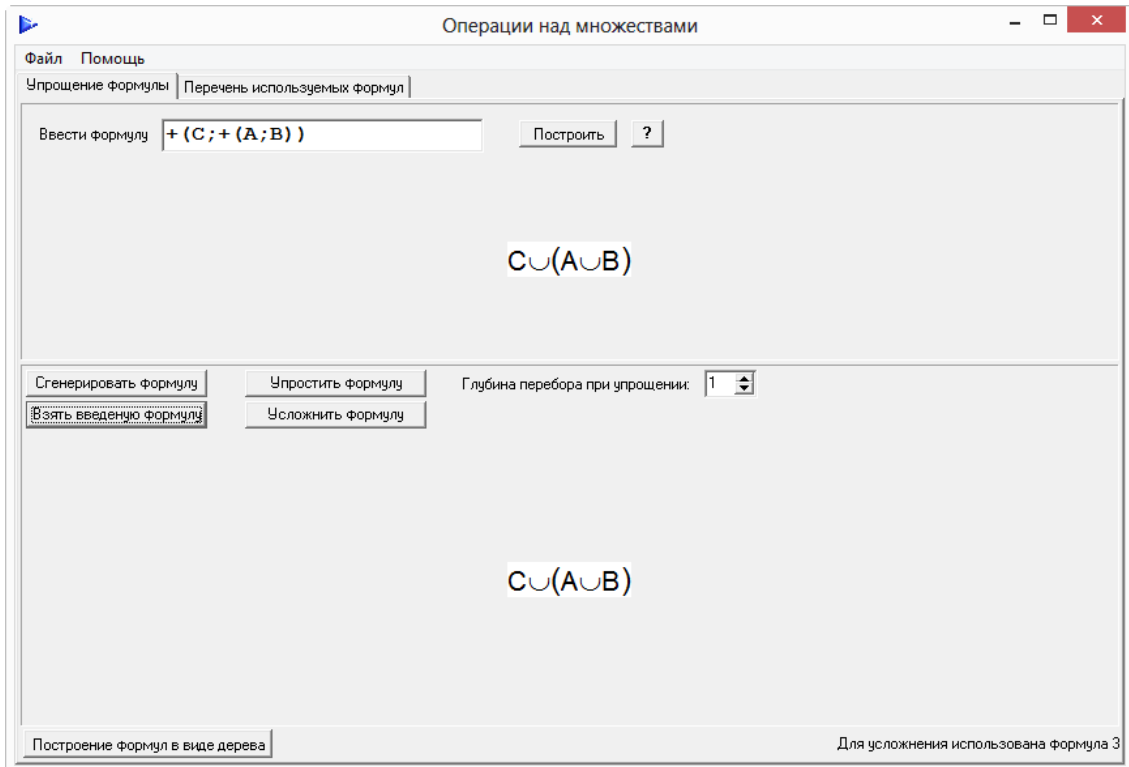
Окно «Главное окно»



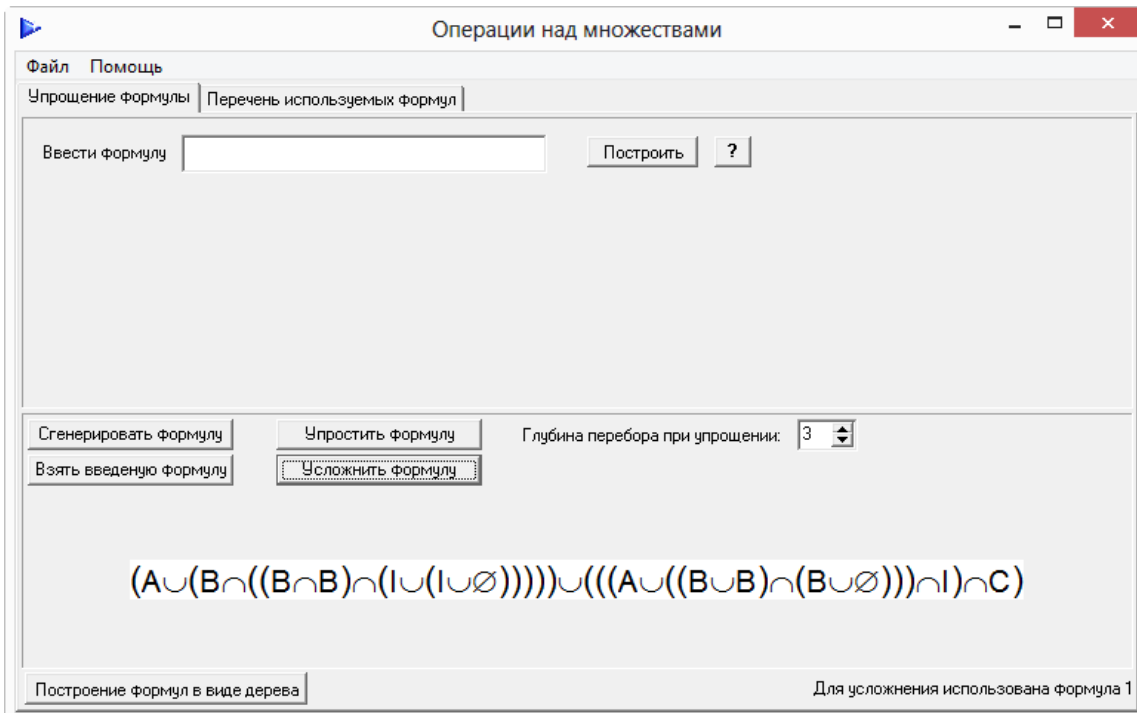
Окно «Перечень используемых формул»



Окно «Введённая пользователем вручную формула»



Окно «Результат генерации и усложнения формулы»



Окно «Дерево для формулы»

Операции над множествами

Файл | Помощь

Упрощение формулы | Перечень используемых формул

Ввести формулу

Глубина перебора при упрощении: 3

Построение формул в виде дерева Для усложнения использована формула 1

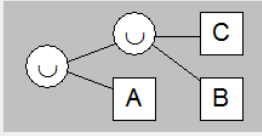
Окно «Отображение введенной и усложненной формулы в виде дерева»

Операции над множествами

Файл | Помощь

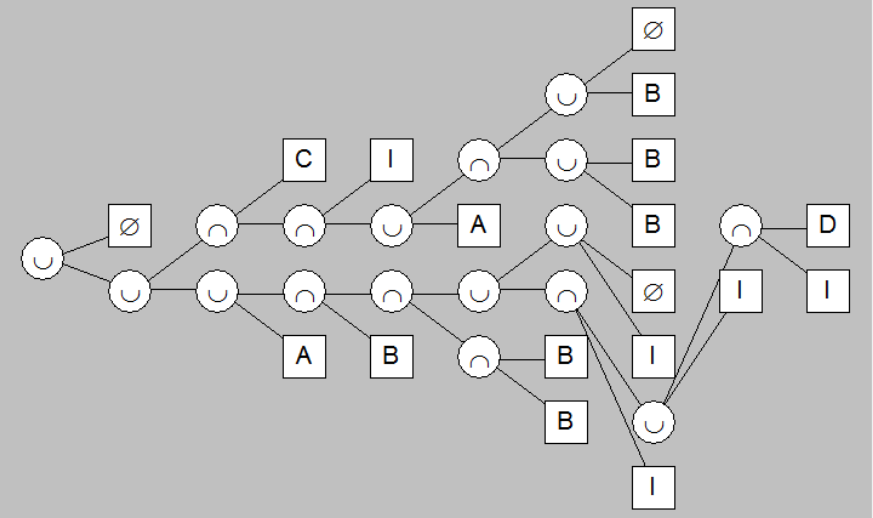
Упрощение формулы | Перечень используемых формул

Ввести формулу: Построить ?



Сгенерировать формулу | Упростить формулу | Глубина перебора при упрощении: 3

Взять введенную формулу | Усложнить формулу | Свернуть | Развернуть



Построение формул в виде дерева Для усложнения использована формула 18

2.7 Исходный текст программы

Федеральное государственное бюджетное образовательное учреждение высшего образования

"Приамурский государственный университет имени Шолом-Алейхема"

ПРОГРАММА ДЛЯ ЭВМ

Программный комплекс по классической теории множеств

Исходный текст программы

Листов 19

Авторы: Н.А.Седова,
В.А.Седов,
Р.И.Баженов

© ПГУ им. Шолом-Алейхема, 2017

Биробиджан 2017 г.

Листинг программы

Листинг программы

```
unit WinSrc;

interface

uses

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Formulas, Grids, ComCtrls, StdCtrls, ExtCtrls, Buttons, Spin;

type

  TMainForm = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    DrawGridLaws: TDrawGrid;
    Label1: TLabel;
    Panel1: TPanel;
    EditOPZ: TEdit;
    BitBtnBuild: TBitBtn;
    PaintBoxUser: TPaintBox;
    Label2: TLabel;
    Panel2: TPanel;
    PaintBoxGen: TPaintBox;
    Button1: TButton;
    Button2: TButton;
```

```
Button3: TButton;

Splitter1: TSplitter;

Bevel1: TBevel;

Bevel2: TBevel;

Button4: TButton;

Button5: TButton;

Panel3: TPanel;

BtnDrawAsTree: TSpeedButton;

Button6: TButton;

Label4: TLabel;

SpinEdit: TSpinEdit;

Label3: TLabel;

Button7: TButton;

procedure DrawGridLawsDrawCell(Sender: TObject; ACol, ARow: Integer;
    Rect: TRect; State: TGridDrawState);

procedure FormCreate(Sender: TObject);

procedure FormDestroy(Sender: TObject);

procedure BitBtnBuildClick(Sender: TObject);

procedure PaintBoxUserPaint(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure PaintBoxGenPaint(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure BtnDrawAsTreeClick(Sender: TObject);

procedure Button6Click(Sender: TObject);
```

```

private
  { Private declarations }

  Law : ALaw;

  UserFormula, GenFormula : PFormula;

public
  { Public declarations }

end;

```

```

var
  MainForm: TMainForm;

```

```

implementation

```

```

{$R *.DFM}

```

```

procedure TMainForm.DrawGridLawsDrawCell(Sender: TObject; ACol, ARow: Integer;

```

```

  Rect: TRect; State: TGridDrawState);

```

```

var

```

```

  b : TBitmap;

```

```

  s : string;

```

```

begin

```

```

  with Sender as TDrawGrid, Canvas do

```

```

    if ACol = 0 then

```

```

      begin

```

```

        s := IntToStr ( ARow+1 );

```

```

        TextOut ( (Rect.Right + Rect.Left - TextWidth(s)) div 2,

```

```

          (Rect.Bottom + Rect.Top - TextHeight(s)) div 2, s )

```

```

end
else if Length(Law) > ARow then
begin
Brush.Color := clWhite;
FillRect ( Rect );
b := DrawLaw ( Law[ARow] );
Draw ( Rect.Left+5, Rect.Top+2, b );
b.Free;
end;
end;

```

```

procedure TMainForm.FormCreate(Sender: TObject);
var
s : TStringList;
i : integer;
b : TBitmap;
begin
s := TStringList.Create; // читаем используемые в преобразованиях формулы
S.LoadFromFile ( 'Laws.txt' );
SetLength ( Law, S.Count );
DrawGridLaws.RowCount := S.Count;
DrawGridLaws.ColWidths[0] := 30;

for i := 1 to S.Count do
begin
Law[i-1] := BuildLaw ( s[i-1] );
b := DrawLaw ( Law[i-1] );

```

```

with DrawGridLaws do

begin

  if ColWidths[1] < b.Width then ColWidths[1] := b.Width;

  if RowHeights[i-1] < (b.Height+4) then RowHeights[i-1] := (b.Height+4);

end;

b.Free;

end;

DrawGridLaws.ColWidths[1] := DrawGridLaws.ColWidths[1] + 10;

s.Free;

UserFormula := nil;

EditOPZ.Hint := ' Формула вводится в функциональной записи.'+#13+#10+
  'Используются следующие обозначения:' + #13+#10+#13+#10+
  ' +(<операнд1>;<операнд2>) - объединение;' + #13 + #10+
  ' *(<операнд1>;<операнд2>) - пересечение;' + #13 + #10+
  ' -(<операнд1>;<операнд2>) - разность;' + #13 + #10+
  ' $(<операнд1>;<операнд2>) - симметрическая разность;' + #13 + #10+
  ' ~(<операнд1>) - отрицание.' + #13+#10+#13+#10+
  ' 0 (ноль) - пустое множество'+#13+#10+
  ' I - полное множество'+#13+#10+
  ' В качестве имени множества можно использовать'+#13+#10+
  'любой из оставшихся символов, имя множества состоит'+#13+#10+
  'из одного символа.';

end;

```

```
procedure TMainForm.FormDestroy(Sender: TObject);
```

```
var
```

```
  i : integer;
```

```
begin
```

```
  for i := High ( Law ) downto 0 do
```

```
    FreeLaw ( Law[i] );
```

```
  Law := nil;
```

```
  FreeFormula ( UserFormula );
```

```
end;
```

```
procedure TMainForm.BitBtnBuildClick(Sender: TObject);
```

```
begin
```

```
  if UserFormula <> nil then FreeFormula ( UserFormula );
```

```
  BuildFormula ( EditOPZ.Text, UserFormula );
```

```
  PaintBoxUser.Repaint;
```

```
end;
```

```
procedure TMainForm.PaintBoxUserPaint(Sender: TObject);
```

```
var
```

```
  b : TBitmap;
```

```
begin
```

```
  if UserFormula <> nil then
```

```
    begin
```

```
      if BtnDrawAsTree.Down then b := DrawFormulaTree ( UserFormula, false )
```

```
      else b := DrawFormula ( UserFormula, false );
```

```
      with PaintBoxUser, Canvas do
```

```
        Draw ( (ClientWidth - b.Width) div 2, (ClientHeight - b.Height) div 2, b );
```



```

    b.Free;

end;

end;

procedure TMainForm.Button1Click(Sender: TObject);

var

    A, B : PFormula;

begin

    if GenFormula <> nil then FreeFormula ( GenFormula );

    New ( A );

    fillchar ( A^, SizeOf(TFormula), 0 );

    A.Used := 1;

    A.Kind := tmSet;

    A.Data := ord ( 'A' );

    New ( B );

    fillchar ( B^, SizeOf(TFormula), 0 );

    B.Used := 1;

    B.Kind := tmSet;

    B.Data := ord ( 'B' );

    New ( GenFormula );

    GenFormula.Used := 1;

    GenFormula.Operand[0] := A;

    GenFormula.Operand[1] := B;

    GenFormula.Data := 0;

    GenFormula.Kind := TFormulaKind(ord(opIntersect) + Random(4));

```

```
PaintBoxGen.Repaint;
```

```
end;
```

```
procedure TMainForm.PaintBoxGenPaint(Sender: TObject);
```

```
var
```

```
  b : TBitmap;
```

```
begin
```

```
  if GenFormula <> nil then
```

```
    begin
```

```
      if BtnDrawAsTree.Down then b := DrawFormulaTree ( GenFormula, false )
```

```
      else b := DrawFormula ( GenFormula, false );
```

```
      with PaintBoxGen, Canvas do
```

```
        Draw ( (ClientWidth - b.Width) div 2, (ClientHeight - b.Height) div 2, b );
```

```
        b.Free;
```

```
      end;
```

```
end;
```

```
procedure TMainForm.Button2Click(Sender: TObject);
```

```
begin
```

```
  if GenFormula <> nil then
```

```
    begin
```

```
      Mutate ( GenFormula, Law );
```

```
      Label3.Caption := 'Для усложнения использована формула ' + IntToStr ( LawId+1 );// + ' ; часть ' +  
IntToStr ( PartId );
```

```
      PaintBoxGen.Repaint;
```

```
    end;
```

```
end;
```

```
procedure TMainForm.Button3Click(Sender: TObject);  
begin  
  if GenFormula <> nil then FreeFormula ( GenFormula );  
  if UserFormula <> nil then GenFormula := Duplicate ( UserFormula )  
  else GenFormula := nil;  
  PaintBoxGen.Repaint;  
end;
```

```
procedure TMainForm.Button4Click(Sender: TObject);  
begin  
  CompressFormula ( GenFormula );  
  PaintBoxGen.Repaint;  
end;
```

```
procedure TMainForm.Button5Click(Sender: TObject);  
begin  
  DecompressFormula ( GenFormula );  
  PaintBoxGen.Repaint;  
end;
```

```
procedure TMainForm.BtnDrawAsTreeClick(Sender: TObject);  
begin  
  PaintBoxGen.Repaint;  
  PaintBoxUser.Repaint;  
end;
```

```
procedure TMainForm.Button6Click(Sender: TObject);
```

```
begin
```

```
if GenFormula <> nil then
```

```
begin
```

```
Minimize ( GenFormula, Law, SpinEdit.Value );
```

```
PaintBoxGen.Repaint;
```

```
end;
```

```
end;
```

```
end.
```

```
unit formulas;
```

```
interface
```

```
uses Math, Windows, SysUtils, Classes, Graphics;
```

```
type
```

```
TFormulaKind = ( opNot, opIntersect, opCross, opSubscr, opSimm, tmSet ); // типы операций ( и операндов )
```

```
PFormula = ^TFormula;
```

```
TFormula = record // одно действие над максимум двумя операндами либо терминальный элемент формулы - имя множества
```

```
Used : integer; // сколько раз используется в выражении
```

```
Kind : TFormulaKind; // тип: действие, множество.
```

```
Operand : array [0..1] of PFormula; // операнды, если есть
```

```
Data : integer; // поле данных - обозначение множества.
```

```

end;

APFormula = array of PFormula;

TTerminalRec = record

  Data : integer; // поле данных - обозначение множества.

  Index : array[0..1] of integer; // индексы в массивах, сгенерированных функцией TreeToVector

end;

TLaw = record

  Part : array [0..1] of PFormula;

  Terminal : array of TTerminalRec; // список терминальных символов, т.е. имен множеств + их
положение в векторной записи формул

end;

ALaw = array of TLaw;

PApplyRec = ^TApplyRec;

TApplyRec = record

  //Target : PFormula; // заменяемый фрагмент

  Target : integer;

  Part : integer; // часть формулы, используемая для замены (0 или 1)

  Law : integer; // номер использованного закона

  Terms : APFormula; // адреса терминальных элементов формулы

  Next : PApplyRec;

end;

AApApplyRec = array of TApplyRec;

APApApplyRec = array of PApplyRec;

const

```

```

FONT_SIZE : integer = 16; // размер шрифта при отрисовке формул

ALL_BRACES : boolean = true; // выделять скобками все операции, кроме отрицания

EMPTY_SET = -'∅'; // знак пустого множества

FULL_SET = ord('I'); // знак универсального множества

function DrawFormula ( Formula : PFormula; NeedBrace : boolean ) : TBitmap; // рисует формулу в
bitmap

procedure BuildFormula ( Src : string; var Result : PFormula ); // по содержащей формулу в виде
польской записи строит древовидную структуру, с которой и работаем

procedure CompressFormula ( Src : PFormula ); // удаляет повторяющиеся ветви, сохраняя смысловую
структуру связей; граф из древовидного становится циклическим

procedure DecompressFormula ( var Src : PFormula ); // восстанавливает повторяющиеся ветви, сохраняя
смысловую структуру связей; граф становится деревом.

procedure FreeFormula ( Src : PFormula ); // удаляет формулу

function BuildLaw ( Src : string ) : TLaw; // генерирует закон, т. е. формулу, используемую в
преобразованиях.

function DrawLaw ( Law : TLaw ) : TBitmap; // рисует закон ( 2 формулы и символ тождества между
ними )

procedure FreeLaw ( Law : TLaw ); // удаляет закон

function Duplicate ( Src : PFormula ) : PFormula;

function TryApplyLaw ( Src : PFormula; Law : TLaw; LawId : integer ) : PApplyRec;

function FindVariants ( Src : PFormula; Laws : ALaw ) : PApplyRec;

procedure Apply ( var Src : PFormula; const Variant : PApplyRec; Laws : ALaw );

procedure Mutate ( var Src : PFormula; Laws : ALaw );

procedure Minimize ( var Src : PFormula; Laws : ALaw; const Deep : integer ); // Deep - глубина перебора

```

```
function DrawFormulaTree ( Formula : PFormula; RightToLeft : boolean ) : TBitmap; // рисует формулу в
bitmap в виде дерева
```

```
var
```

```
    LawId, PartId : integer;
```

```
implementation
```

```
function Equal ( p1, p2 : PFormula ) : boolean;
```

```
begin
```

```
    Result := false;
```

```
    if p1 = p2 then Result := true
```

```
    else if (p1 = nil) or (p2 = nil) then Result := false
```

```
    else if p1.Kind = P2.Kind then
```

```
        if p1.Kind <> tmSet then
```

```
            Result := CompareMem ( p1, p2, SizeOf(TFormula) )
```

```
        else Result := p1.Data = p2.Data;
```

```
end;
```

```
function Count ( F : PFormula ) : integer; // кол-во элементов. Элемент - операция или имя множества.
```

```
begin
```

```
    if F = nil then Result := 0
```

```
    else Result := 1 + Count( F.Operand[0] ) + Count( F.Operand[1] );
```

```
end;
```

```
function TreeToVector ( Src : PFormula; Expand : boolean ) : APFormula; // если Expand - true, указатель
на элемент формулы встречается столько раз, сколько раз в формуле встречается сам элемент. Иначе
- только 1 раз.
```

```
var
```

P : PFormula; // составляющие формулы

i : integer;

procedure EnumElements (F : PFormula); // собирает элементы формулы в P. Использует переменные: P, i.

var

j : integer;

begin

if F <> nil then

begin

if not Expand then for j := i-1 downto 0 do if F = P[j] then Exit; // отсев повторов

P[i] := F;

Inc (i);

EnumElements (F.Operand[0]);

EnumElements (F.Operand[1]);

end;

end;

begin

SetLength (P, Count(Src));

i := 0;

EnumElements (Src);

P := Copy (P, 0, i);

Result := P;

end;

function DrawFormula (Formula : PFormula; NeedBrace : boolean) : TBitmap;

var

b1, b2 : TBitmap;


```
s : string;
```

```
function DrawString ( F : TFont; S : String ) : TBitmap;
```

```
begin
```

```
Result := TBitmap.Create;
```

```
with Result, Canvas do
```

```
begin
```

```
Brush.Style := bsSolid;
```

```
Brush.Color := clWhite;
```

```
Font.Assign ( F );
```

```
Width := TextWidth ( s ) + 1;
```

```
Height := TextHeight ( s );
```

```
TextOut ( 0, 0, s );
```

```
end;
```

```
end;
```

```
procedure AddBraces ( var B : TBitmap );
```

```
var
```

```
T, Buf : TBitmap;
```

```
w : integer;
```

```
begin
```

```
Buf := TBitmap.Create;
```

```
with Buf, Canvas do
```

```
begin
```

```
Font.Name := 'Arial';
```

```
Font.Height := B.Height;
```

```
Height := B.Height;
```

```

w := TextWidth ( '(' );
Width := B.Width + w + TextWidth ( ')' );
Brush.Style := bsSolid;
Brush.Color := clWhite;
TextOut (0, 0, '(' );
Draw ( w, 0, B );
TextOut (w + B.Width, 0, ')' );
end;
T := B; B := Buf; T.Free;
end;

```

```

function CheckBraces ( o1, o2 : PFormula ) : boolean;
begin
Result := false;
if (o1 <> nil) and (o2 <> nil) then if o2.Kind <> tmSet then
if (o2.Kind <> opNot) and (o1.Kind <> opNot) then
if (ord(o2.Kind) <= ord(o1.Kind)) or ALL_BRACES then Result := true;
end;
end;

```

```

function DrawFormulaTree ( Formula : PFormula; RightToLeft : boolean ) : TBitmap; // рисует формулу в
битмап в виде дерева
var
i, j, k, m, levels, CharSize : integer;
p : APFormula;
A, l : array of integer;
f : boolean;
pnt : array of TPoint;
s : string;

```

```

procedure Link ( i : integer );
var
  j, k : integer;
begin
  with Result, Canvas, p[i]^ do
  // if Operand[0] <> Operand[1] then
    for j := 0 to 1 do if Operand[j] <> nil then
      begin
        for k := High ( p ) downto 0 do
          if Operand[j] = p[k] then
            PolyLine ( [ Point(Pnt[i].x, Pnt[i].y + (4*j-2)),
                        Point(Pnt[k].x, Pnt[k].y + (4*j-2)) ] )
          end
        end;
      end;
    end;

begin
  Result := TBitmap.Create;
  with Result.Canvas.Font do
    begin
      Name := 'Arial';
      Size := FONT_SIZE;
      Color := clBlack;
    end;
  CharSize := Round ( Result.Canvas.TextHeight ( 'A' ) * 1.5 );

  P := TreeToVector ( Formula, false );

```

```

SetLength ( A, Length(P) );

fillchar ( A[0], Length(A)*SizeOf(Integer), 0 );

A[0] := 1; Levels := 1;

repeat

f := true;

for i := 0 to High ( P ) do if P[i] <> nil then

if A[i] > 0 then

for j := High ( P ) downto 0 do

for k := 0 to 1 do

if P[i].Operand[k] <> nil then if P[i].Operand[k] = P[j] then

if A[j] <= A[i] then

begin

A[j] := A[i]+1;

Levels := Max ( Levels, A[j] );

f := false;

end;

until f;

SetLength ( l, Levels );

fillchar ( l[0], Length(l)*SizeOf(Integer), 0 );

for i := High ( A ) downto 0 do Inc ( l[A[i]-1] );

k := 0;

for i := High ( l ) downto 0 do k := Max ( k, l[i] );

SetLength ( Pnt, Length ( P ) );

for i := Length ( l ) downto 1 do

begin

m := 0;

for j := High ( P ) downto 0 do

```

```

if A[j] = i then
begin
if RightToLeft then Pnt[j].x := Round ( (Length(l) - i + 0.5) * CharSize * 2 )
else Pnt[j].x := Round ( (i - 0.5) * CharSize * 2 ) ;
Pnt[j].y := Round((0.5 + (k - l[i-1])/2 + m) * CharSize * 1.5);
Inc ( m );
end;
end;

```

```

procedure BuildFormula ( Src : string; var Result : PFormula );

```

```

function SubRead ( s : string; var i : integer ) : PFormula ;

```

```

function Find ( c : char ) : boolean;

```

```

begin

```

```

while i < Length(s) do

```

```

if s[i] = c then begin Result := true; exit end

```

```

else Inc ( i );

```

```

Result := false;

```

```

end;

```

```

begin

```

```

Result := nil;

```

```

if i > Length(Src) then Exit;

```

```

if s[i] in [')', ';'] then Exit;

```

```

new ( Result );

```

```

fillchar ( Result^, SizeOf ( TFormula ), 0 );

```

```

begin

```

```

Result.Used := 1;

```

```

Result.Operand[0] := nil;

Result.Operand[1] := nil;

Result.Data := 0;

case s[i] of

'0' : begin Result.Kind := tmSet; Result.Data := EMPTY_SET; end;

'1' : begin Result.Kind := tmSet; Result.Data := FULL_SET; end;

'+' : Result.Kind := opIntersect;

'*' : Result.Kind := opCross;

'-' : Result.Kind := opSubscr;

'$' : Result.Kind := opSimm;

'^' : Result.Kind := opNot;

else begin Result.Kind := tmSet; Result.Data := ord(s[i]); end;

end;

Inc ( i );

if Result.Kind <> tmSet then

if Find('(') then

begin

Inc ( i );

Result.Operand[0] := SubRead ( s, i );

if Result.Kind <> opNot then

begin

if not Find(';') then Exit;

Inc ( i );

Result.Operand[1] := SubRead ( s, i );

end;

if not Find(')') then Exit;

end;

end;

```

```

    end;

end;

var
    i : integer;

begin
    i := 1;

    Result := SubRead ( Src, i );

end;

procedure CompressFormula ( Src : PFormula );

var
    P : APFormula;
    A : array of integer;
    i, j, n : integer;
    flag : boolean;

begin
    P := TreeToVector ( Src, false );
    n := Length ( P );
    SetLength ( A, n );
    Dec(n);
    for i := n downto 0 do A[i] := i;

repeat
    flag := true;
    for i := 1 to n do
        if A[i] = i then

```

```

for j := i+1 to n do
  if A[j] = j then
    if Equal(P[i], P[j]) then
      begin
        A[j] := i;
        flag := false;
      end;
    if not flag then
      for i := 0 to n do
        if (P[i] <> nil) and (A[i] <> i) then
          begin
            for j := 0 to n do if i <> j then if P[j] <> nil then with P[j]^ do
              begin
                if Operand[0] = P[i] then
                  begin Operand[0] := P[A[i]]; if P[A[i]] <> nil then Inc ( P[A[i]].Used ) end;
                if Operand[1] = P[i] then
                  begin Operand[1] := P[A[i]]; if P[A[i]] <> nil then Inc ( P[A[i]].Used ) end;
              end;
            Dispose ( P[i] );
            P[i] := nil;
          end;
        until flag;

      P := nil;
      A := nil;
    end;

```



```

function BuildLaw ( Src : string ) : TLaw;

var
i, j, k, m : integer;
n : array[0..1] of integer;
P : array [0..1] of APFormula;
f : boolean;

begin
with Result do
begin
i := Pos ('=', Src );
BuildFormula ( Copy ( Src, 1, i-1 ), Part[0] );
BuildFormula ( Copy ( Src, i+1, Length(Src)-i ), Part[1] );

for i := 0 to 1 do
begin
CompressFormula ( Part[i] ); // чтобы терминальные символы попадались по одному разу.
P[i] := TreeToVector ( Part[i], false );
n[i] := 0;
for j := High ( P[i] ) downto 0 do if P[i][j].Kind = tmSet then Inc ( n[i] );
end;

Terminal := nil;
SetLength ( Terminal, n[0] + n[1] ); // с запасом - потом уменьшим.
m := 0;

for i := 0 to 1 do

```

```

for j := High ( P[i] ) downto 0 do with P[i][j]^ do
  if Kind = tmSet then
    begin
      f := true;
      for k := m-1 downto 0 do
        if Terminal[k].Data = Data then
          begin
            f := false;
            Terminal[k].Index[i] := j;
            break;
          end;
        if f then
          begin
            Terminal[m].Data := Data;
            Terminal[m].Index[i] := j;
            Terminal[m].Index[1-i] := -1;
            Inc ( m );
          end;
        end;
      end;
    end;

Terminal := Copy ( Terminal, 0, m );

P[0] := nil;
P[1] := nil;

end;
end;

```

```

function DrawLaw ( Law : TLaw ) : TBitmap;

var

b1, b2 : TBitmap;

begin

Result := TBitmap.Create;

b1 := DrawFormula ( Law.Part[0], false );

b2 := DrawFormula ( Law.Part[1], false );

with Result, Canvas do

begin

Brush.Style := bsSolid;

Brush.Color := clWhite;

Font.Name := 'Symbol';

Font.Size := FONT_SIZE;

Width := b1.Width + b2.Width + TextWidth ( #$ba );

Height := Max ( Max ( b1.Height, b2.Height ), TextHeight ( #$ba ) );

Draw ( 0, Height-b1.Height, b1 );

TextOut ( b1.Width, Height - TextHeight ( #$ba ), #$ba );

Draw ( Width - b2.Width, Height-b2.Height, b2 );

end;

b1.Free;

b2.Free;

end;

procedure FreeLaw ( Law : TLaw );

begin

with Law do

begin

```

```

FreeFormula ( Part[0] );
FreeFormula ( Part[1] );
Terminal := nil;
end;
end;

procedure AddRes ( Src : PApplyRec; var R : PApplyRec );
var
P : PApplyRec;
begin
if R = nil then R := Src
else
begin
P := R; while P.Next <> nil do P := P.Next; P.Next := Src;
end;
end;

function TryApplyLaw ( Src : PFormula; Law : TLaw; LawId : integer ) : PApplyRec;
var
T : APFormula;
Part : integer;
AP : APFormula;

function CheckTerminal ( F1, F2 : PFormula ) : boolean;
var
i : integer;
begin

```

```

Result := false;

with Law do

for i := High ( Terminal ) downto 0 do with Terminal[i] do

if Data = F2.Data then

begin

if (F2.Data = EMPTY_SET) or (F2.Data = FULL_SET) then

if (F1.Kind <> F2.Kind) or (F1.Data <> F2.Data) then

begin Result := false; Exit end;

if T[i] <> nil then Result := T[i] = F1

else

begin

T[i] := F1;

Result := true;

Exit;

end;

end;

end;
end;

```

```

function Check ( F1, F2 : PFormula ) : boolean;

begin

if (F1 = nil) then Result := F2 = nil

else if F2.Kind <> tmSet then

begin

if F1.Kind = F2.Kind then

Result := Check ( F1.Operand[0], F2.Operand[0] ) and

Check ( F1.Operand[1], F2.Operand[1] )

else Result := false;

```

```

end

else

  Result := CheckTerminal(F1, F2);

end;

function CheckAll ( {F1, }F2 : PFormula ) : PApplyRec;

var

  R : PApplyRec;

  i : integer;

begin

  Result := nil;

  for i := High ( AP ) downto 0 do

    if AP[i] <> nil then

      begin

        fillchar ( T[0], Length(T)*SizeOf(T[0]), 0 );

        if Check ( AP[i], F2 ) then

          begin

            New ( R );

            R.Target := i;

            R.Part := Part;

            R.Law := LawId;

            R.Terms := Copy ( T, 0, Length(T) );

            R.Next := nil;

            AddRes ( R, Result );

          end;

        end;

      end;

    end;

  end;

end;

```

```

begin
  Result := nil;
  CompressFormula ( Src );
  SetLength ( T, Length ( Law.Terminal ) );

  AP := TreeToVector ( Src, true );

  for Part := 0 to 1 do
    AddRes ( CheckAll ( {Src,} Law.Part[Part] ), Result );
    T := nil;

    AP := nil;
  end;

function ListToVector ( R : PApplyRec ) : AApplyRec;
var
  i : integer;
  P : PApplyRec;
begin
  P := R;
  i := 0;
  while P <> nil do begin Inc ( i ); P := P.Next end;
  SetLength ( Result, i );
  P := R;
  i := 0;
  while P <> nil do begin Result[i] := P; Inc ( i ); P := P.Next end;

```

```
end;
```

```
function FindVariants ( Src : PFormula; Laws : ALaw ) : PApplyRec;
```

```
var
```

```
  i : integer;
```

```
begin
```

```
  CompressFormula ( Src );
```

```
  Result := nil;
```

```
  for i := 0 to High ( Laws ) do
```

```
    AddRes ( TryApplyLaw ( Src, Laws[i], i ), Result ); // изучаем все возможные варианты модификации  
    исходной формулы на основании всех законов, строим из вариантов список...
```

```
end;
```

```
procedure Apply ( var Src : PFormula; const Variant : PApplyRec; Laws : ALaw );
```

```
var
```

```
  i, j, k, l, NewSet : integer;
```

```
  F, Ft : PFormula;
```

```
  AF : APFormula;
```

```
begin
```

```
  with Variant^ do
```

```
    begin
```

```
      NewSet := ord ( 'A' );
```

```
      F := Duplicate ( Laws[Law].Part[1-Part] );
```

```
      AF := TreeToVector ( Src, true );
```

```
      for j := High ( AF ) downto 0 do with AF[j]^ do
```



```

if Kind = tmSet then

  if Data <> FULL_SET then

    if Data >= NewSet then NewSet := Data+1;

if NewSet = FULL_SET then Inc ( NewSet );

AF := nil;

AF := TreeToVector ( F, false );

for j := High ( Terms ) downto 0 do

begin

  k := Laws[Law].Terminal[j].Index[1-Part];

  if k > -1 then

    if Terms[j] <> nil then

      begin

        for l := High ( AF ) downto 0 do if l <> k then if AF[l] <> nil then

          begin

            if AF[l].Operand[0] = AF[k] then AF[l].Operand[0] := Duplicate ( Terms[j] );

            if AF[l].Operand[1] = AF[k] then AF[l].Operand[1] := Duplicate ( Terms[j] );

          end;

          Ft := AF[k];

          FreeFormula ( AF[k] );

          if k = 0 then AF[k] := Duplicate ( Terms[j] ) else AF[k] := nil;

          for l := High ( AF ) downto 0 do

            if AF[l] = Ft then AF[l] := AF[k]; // для удобства отладки

          end

        else if (AF[k].Data <> FULL_SET) and (AF[k].Data <> EMPTY_SET) then

          begin

            AF[k].Data := NewSet;

```

```

    Inc ( NewSet );

end;

end;

Ft := AF[0];

AF := nil;

DecompressFormula ( Src );

AF := TreeToVector ( Src, true );

for i := High ( AF ) downto 0 do

begin

    if AF[i].Operand[0] = AF[Target] then AF[i].Operand[0] := Ft;

    if AF[i].Operand[1] = AF[Target] then AF[i].Operand[1] := Ft;

end;

FreeFormula(AF[Target]);

if Src = AF[Target] then Src := Ft;

AF[Target] := nil;

AF := nil;

end;

end;

procedure Minimize ( var Src : PFormula; Laws : ALaw; const Deep : integer ); // Deep - глубина перебора

procedure Find ( var S : PFormula; const D : integer );

var

```

```

i, j, l, k : integer;

P : PApplyRec;

A : APApplyRec;

F, R : PFormula;

begin

DecompressFormula ( S );

P := FindVariants ( S, Laws );

A := ListToVector ( P );

j := Count ( S );

l := -1;

R := nil;

for i := High ( A ) downto 0 do

begin

F := Duplicate ( S );

Apply ( F, A[i], Laws );

if D > 1 then Find ( F, D-1 );

k := Count ( F );

if k < j then

begin

j := k;

l := i;

if R <> nil then FreeFormula ( R );

R := F;

end

else FreeFormula ( F );

end;

```

```
if R <> nil then
begin
FreeFormula ( S );
S := R;
end;

for i := High ( A ) downto 0 do Dispose ( A[i] );
A := nil;
end;

begin
Find ( Src, Deep );
end;

LawId := A[i].Law;
PartId := A[i].Part;

Apply ( Src, A[i], Laws );

for i := High ( A ) downto 0 do Dispose ( A[i] ); // освобождаем память от списка вариантов.
A := nil;
end;

end.
```

2.8 Согласие автора на указание сведений об авторе, указанных в заявлении

<p>Дата поступления</p> <p><i>(заполняется Федеральной службой по интеллектуальной собственности)</i></p>	<p>В Федеральную службу по интеллектуальной собственности Бережковская наб., д. 30, корп. 1, г. Москва, Г-59, ГСП-3, 125993, Российская Федерация</p>
<p>Согласие автора на указание сведений об авторе, указанных в заявлении</p>	
<p>Заявка № _____ <i>(указывается при наличии регистрационного номера заявки)</i></p> <p>на государственную регистрацию: <input checked="" type="checkbox"/> Программы для ЭВМ <input type="checkbox"/> Базы данных, государственная регистрация которой осуществляется в соответствии с пунктом 4 статьи 1259 Кодекса <input type="checkbox"/> Базы данных, государственная регистрация которой осуществляется в соответствии с пунктом 3 статьи 1334 Кодекса <i>(Отметить знаком «X» вид результата интеллектуальной деятельности)</i></p> <p>Название: <u>« Программный комплекс по классической теории множеств»</u></p> <p>_____</p> <p><i>(указывается в соответствии с графой 1 заявления о государственной регистрации программы для ЭВМ или базы данных)</i></p>	
<p>Правообладатель (и) (Заявитель)(и) <i>(указываются фамилия, имя, отчество (последнее – при наличии), место жительства физического лица, наименование, место нахождения, основной государственный регистрационный номер (ОГРН) и идентификационный номер налогоплательщика (ИНН) юридического лица)</i></p> <p style="text-align: center;"><u>Федеральное государственное бюджетное образовательное учреждение высшего образования «Приамурский государственный университет имени Шолом-Алейхема»</u></p> <p>679015, Российская Федерация, Еврейская автономная область, г. Биробиджан, ул. Широкая, д. 70а (RU)</p> <p><i>Подтверждаю согласие на указание обо мне, как авторе, следующих сведений в графе 7А заявления на государственную регистрацию данной программы для ЭВМ или базы данных.</i></p> <p>7А. СВЕДЕНИЯ ОБ АВТОРЕ:</p> <p>Фамилия имя отчество: Седова Нелли Алексеевна</p> <p>Дата рождения: число: <u>17</u> месяц: <u>10</u> год: <u>1981</u> Гражданство: <u>Российская Федерация</u></p> <p>Автор согласен с обработкой указанных персональных данных, необходимой для исполнения полномочий федеральных органов исполнительной власти, участвующих в предоставлении государственных услуг, предусмотренных Федеральным законом от 27 июля 2010 года № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг», включая регистрацию субъекта персональных данных на едином портале государственных и муниципальных услуг и (или) региональных порталах государственных и муниципальных услуг и в течение срока действия исключительного права на регистрируемый объект.</p> <p>Место постоянного жительства, включая указание страны:</p> <p>690001, Российская Федерация, Приморский край, г. Владивосток, ул. Стаса, 122, кв. 435 (RU)</p> <p>Краткое описание творческого вклада автора при создании регистрируемой программы для ЭВМ или базы данных: Написание программного комплекса, отладка программного комплекса.</p> <p>При публикации сведений о государственной регистрации программы для ЭВМ или базы данных автор просит: <i>(отметить [X])</i></p> <p><input checked="" type="checkbox"/> упоминать его под своим именем <input type="checkbox"/> не упоминать его (анонимно)</p>	

упоминать его под псевдонимом: _____

Подпись автора:

(подпись должна быть расшифрована)

Подпись(и) правообладателя(ей) или его (их) представителя(ей)

Ректор ФГБОУ ВО «Приамурский государственный университет имени Шолом-Алейхема»

Н.Г. Баженова

М.П.

«14» сентября 2017 г.

(от имени юридического лица заявление подписывается руководителем организации или иным лицом, уполномоченным на это в установленном законодательством Российской Федерации порядке, с указанием его должности, подпись удостоверяется печатью юридического лица при наличии печати. Подпись любого лица должна быть расшифрована с указанием фамилии и инициалов и даты подписания заявления)

2.9 Согласие на обработку персональных данных

В Федеральную службу
по интеллектуальной собственности
Бережковская наб., д. 30, корп. 1,
г. Москва, Г-59, ГСП-3, 125993,
Российская Федерация

Название программы для ЭВМ или базы данных _____

Программный комплекс по по классической теории множеств

№ заявки _____
(указывается при наличии регистрационного номера заявки)

Согласие на обработку персональных данных

Ф. И. О. субъекта персональных данных **Седова Нелли Алексеевна**

Адрес места жительства **Россия, Приморский край, г. Владивосток, ул. Стаса, 612, кв. 435**

Документ, удостоверяющий личность субъекта персональных данных, дата его выдачи и выдавший орган **паспорт серия 11 11 № 222222, выдан 14.11.2006 Дальнереченский ГОВД Приморского края**

Подтверждаю согласие на обработку моих персональных данных, предусмотренную частью 3 статьи 3 Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных», в целях предоставления Федеральной службой по интеллектуальной собственности государственной услуги в соответствии с Федеральным законом от 27 июля 2010 г. № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг».

Мне известно, что в случае отзыва согласия на обработку персональных данных Федеральная служба по интеллектуальной собственности вправе продолжить обработку персональных данных без моего согласия в соответствии с частью 2 статьи 9, пунктом 4 части 1 статьи 6 Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных».

Подпись

/ Седова Н.А. /
(Ф. И. О. субъекта персональных данных)

Дата _____

2.10 Сопроводительное письмо

(оформляется на бланке вуза)

В Федеральную службу по интеллектуальной
собственности
Бережковская наб., 30, корп. 1, Москва,
Г-59, ГСП-3, 123993

Направляю Вам на регистрацию программу для ЭВМ «Программный комплекс по классической теории множеств», правообладателем исключительного права на которую (заявителем) является Федеральное государственное бюджетное образовательное учреждение высшего образования «Приамурский государственный университет имени Шолом-Алейхема».

Комплектность заявки указана в приложении.

Приложение:

1. Заявление	на	1	л.	в	1	экз.
2. Дополнение к заявлению	на	1	л.	в	1	экз.
3. Распечатка исходного текста программы	на	19	л.	в	1	экз.
4. Реферат	на	1	л.	в	2	экз.
5. Материалы аудиовизуальных отображений, порождаемых программой для ЭВМ	на	7	л.	в	1	экз.
6. Согласие на обработку персональных данных (приложение №3)	на	3	л.	в	1	экз.
7. Согласие авторов на указание сведений об авторах, указанных в заявлении (приложение №4)	на	3	л.	в	1	экз.
8. Платежный документ об уплате госпошлины	на	1	л.	в	1	экз.

Всего на 36 листах + 1 платежный документ.

Свидетельство прошу выслать по почте.

Ректор ПГУ им. Шолом-Алейхема

Н.Г. Баженова

14 сентября 2017 г.

Р.И.Баженов

(42622) 4-67-32

2.11 Справочная информация

Справочное пособие составлено на базе нормативных материалов (не является официальным документом)

версия 02.10.2015

Памятка составлена с учётом изменений, внесённых в нормативные документы, по состоянию на 01.10.2015

Консультации по уплате госпошлины: тел.: (499) 240-63-54; (499) 240-24-64; (495) 531-66-05

E-mail: otd69@rupto.ru или mfomichev@rupto.ru или otd6905@rupto.ru

ПАМЯТКА ОБ УПЛАТЕ ГОСУДАРСТВЕННОЙ ПОШЛИНЫ

Государственная пошлина уплачивается при обращении в уполномоченный федеральный орган исполнительной власти по интеллектуальной собственности (Федеральная служба по интеллектуальной собственности - Роспатент) за совершением действий по государственной регистрации программы для ЭВМ (ПрЭВМ), базы данных (БД) и топологии ИМС (ТИМС), а также за регистрацию перехода исключительного права по договору отчуждения и без договора на указанные результаты интеллектуальной деятельности (для ТИМС – дополнительно регистрация предоставления права использования).

Государственная пошлина (госпошлина) уплачивается заявителем в размере и порядке установленном Налоговым Кодексом Российской Федерации (НК РФ). Организация уплачивает госпошлину в безналичной форме путём перевода суммы госпошлины со счёта заявителя на счёт получателя платежа. Расчетные (платежные) документы, подтверждающие факт уплаты государственной пошлины, при ее уплате законным или уполномоченным представителем заявителя должны содержать указание на то, что уплата государственной пошлины осуществляется от имени заявителя и из денежных средств заявителя.

Государственная пошлина уплачивается только в рублях РФ.

Внимание: 1. Госпошлина должна быть уплачена до подачи заявки на регистрацию или до подачи документов на регистрацию договора, перехода права, залога - ст. 333.18 НК РФ.

2. Заявке на регистрацию ПрЭВМ, БД ТИМС присваивается входящий номер Роспатента при условии подтверждения факта уплаты госпошлины в размере и порядке, установленном Налоговым кодексом Российской Федерации.

«Приложение № 2

УТВЕРЖДЕНО приказом Роспатента от 28.03.2014 № 44

Реквизиты счета для уплаты государственных пошлин

за совершение действий по официальной регистрации программы для электронных

вычислительных машин, базы данных и топологий интегральных микросхем

Для учета поступлений федерального бюджета Российской Федерации, уплачиваемых в соответствии со статьей 333.30 Налогового кодекса Российской Федерации (далее - НК РФ):

по КБК 168 1 08 09000 01 1000 110 - «Государственная пошлина за совершение уполномоченным федеральным органом исполнительной власти действий по официальной регистрации программы для электронных вычислительных машин, базы данных и топологий интегральных микросхем»

При заполнении распоряжений о переводе денежных средств в уплату платежей в бюджетную систему Российской Федерации плательщики руководствуются Приказом Минфина России от 12 ноября 2013 г. N 107н «Об утверждении Правил указания информации в реквизитах распоряжений о переводе денежных средств в уплату платежей в бюджетную систему Российской Федерации» следующим образом:

поле 13 «Банк» получателя	Операционный департамент Банка России г. Москва 701
поле 14 «БИК» банка получателя	044501002
поле 15 «Корреспондентский счет» банка получателя	не заполняется
поле 16 «Наименование» получателя	Межрегиональное операционное УФК (Федеральная служба по интеллектуальной собственности)
поле 17 «Счет» получателя	40101810500000001901
поле 22 «Код»	указывается идентификатор начисления
поле 61 «ИНН» получателя	7730176088
поле 103 «КПП» получателя	773001001
поле 104 КБК получателя	16810809000011000110
поле 105 ОКТМО получателя	45318000
поле 108	указывается идентификатор сведений о физическом лице в качестве основных идентификаторов сведений о физическом лице используются: - паспорт гражданина Российской Федерации; - страховой номер индивидуального лицевого счета застрахованного лица в системе персонифицированного учета Пенсионного фонда Российской Федерации (СНИЛС); - водительское удостоверение.
поля 106, 107, 109 и 110	проставляется ноль «0»
поле 24 «Назначение платежа»	указывается подпункт пункта 1 статьи 333.30 НК РФ и

	название объекта регистрации (программы для ЭВМ, базы данных и топологии интегральной микросхемы) в объеме, достаточном для идентификации, а также имя или наименование заявителя.
СПРАВОЧНО: код ОКПО	00038971
ОГРН (администратора платежей)	1047730015200 от 24.06.04

Наличие в расчетном документе незаполненных полей **не допускается** (кроме поля 15).

Распоряжение о переводе денежных средств, подтверждающее уплату государственной пошлины, должно относиться только к одному заявителю (плательщику) и одному объекту регистрации. В том случае если заявителями являются несколько лиц, то государственная пошлина уплачивается ими самостоятельно в равных долях в соответствии с пунктом 2 статьи 333.30 НК РФ».

Уплата государственной пошлины при подаче одним заявителем нескольких заявок осуществляется отдельно по каждой заявке.

В соответствии с положениями статей 333.16 – 333.18 и 45 НК РФ госпошлину должен уплатить заявитель самостоятельно! В том случае если заявителями являются несколько лиц, то госпошлина уплачивается ими в равных долях! (пункт 2 статьи 333.18 НК РФ) Если за совершением действия, предусмотренного пунктом 1 статьи 330.30, обращаются **организации и физические лица, являющиеся правообладателями на ПрЭВМ, или БД, или ТИМС, размер доли госпошлины, уплачиваемой каждым плательщиком, определяется пропорционально количеству плательщиков на основании пункта 2 статьи 333.30 НК РФ исходя из установленных размеров для организаций и физических лиц. Таким образом, при подаче заявки несколькими заявителями каждый заявитель должен самостоятельно осуществить уплату своей доли госпошлины**

Допускается уплата госпошлины законным или уполномоченным представителем налогоплательщика. В силу [пункта 3 статьи 29 НК РФ](#) уполномоченный представитель плательщика госпошлины - организации осуществляет свои полномочия на основании доверенности, выдаваемой в порядке, установленном гражданским законодательством Российской Федерации. Уполномоченный представитель налогоплательщика – физ. лица осуществляет свои полномочия на основании **нотариально удостоверенной доверенности** или доверенности, приравненной к нотариально удостоверенной в соответствии с гражданским законодательством РФ. В доверенности должно быть указано полномочие представителя на уплату госпошлины из средств заявителя.

В соответствии с определением Конституционного суда РФ от 22.01.2014 № 41-О, «представительство в налоговых правоотношениях означает совершение представителем действия от имени и за счет собственных средств налогоплательщика - представляемого лица, следовательно, платежные документы на уплату налога должны исходить от налогоплательщика и быть подписаны им самим, а уплата соответствующих сумм должна производиться за счет средств налогоплательщика, находящихся в его свободном распоряжении, то есть за счет его собственных средств». При этом на факт признания обязанности налогоплательщика по уплате налога исполненной не влияет то, в какой форме - безналичной или наличной - происходит уплата денежных средств, важно, чтобы из представленных платежных документов можно было четко установить, **что, соответствующая сумма госпошлины уплачена именно правообладателем и именно за счет его собственных денежных средств. Таким образом, в графе «Плательщик» платёжного документа**

при уплате госпошлины представителем должно быть указано, например, «Иванов И.И. за ООО «Альфа» из собственных средств ООО «Альфа».

Государственная пошлина за государственную регистрацию ПрЭВМ, БД, ТИМС
(ст. 333.30 п. 1 Налогового Кодекса РФ)

1) за государственную регистрацию ПрЭВМ, БД и ТИМС соответственно в Реестре ПрЭВМ, Реестре БД и Реестре ТИМС, включая выдачу заявителю свидетельства о государственной регистрации ПрЭВМ, БД и ТИМС, а также публикацию сведений о зарегистрированной ПрЭВМ, БД и ТИМС в официальном бюллетене	Для организации		Для физического лица	
	4500 руб.		3000 руб.	
Размер долей госпошлины для нескольких заявителей рассчитывается путём деления суммы госпошлины на количество заявителей, например,				
	Для заявителей – юридических лиц		Для заявителей – физических лиц	
Количество заявителей	Общая сумма госпошлины	Доля каждого заявителя к уплате	Общая сумма госпошлины	Доля каждого заявителя к уплате
Два заявителя	4500 руб.	2250 руб.	3000 руб.	1500 руб.
Три заявителя	4500 руб.	1500 руб.	3000 руб.	1000 руб.
и так далее				
Иные действия, связанные с государственной регистрацией ПрЭВМ, БД и ТИМС				
ст. 333.30 п. 1			Для юр. лица	Для физ. лица
2) за внесение изменений в документы и материалы к заявке на регистрацию ПрЭВМ, БД и ТИМС до публикации в официальном бюллетене			1200 руб.	1200 руб.
3) за внесение по инициативе заявителя изменений в депонированные документы и материалы и выдачу заявителю нового свидетельства о государственной регистрации ПрЭВМ, БД и ТИМС до публикации в официальном бюллетене			2500 руб.	1200 руб.
7) За выдачу дубликата свидетельства о государственной регистрации ПрЭВМ, БД и ТИМС ст. 333.30 п. 1, 7)			1300 руб.	1300 руб.

Государственные пошлины за государственную регистрацию перехода исключительного права на ПрЭВМ, БД и ТИМС по договору и без договора и регистрацию предоставления права использования уплачиваются по сумме подпунктов 4) и 6) п. 1 ст. 333.30]

4) за гос. регистрацию договора об отчуждении исключительного права на зарегистрированную ПрЭВМ или БД, об отчуждении или о залоге исключительного права на зарегистрированную ТИМС, лицензионного договора о предоставлении права использования	5000 руб. + 2500 руб. за <u>каждую</u> предусмотренную договором ПрЭВМ, БД и ТИМС
--	---

зарегистрированной ТИМС, а также за внесение изменений в указанные документы и их государственную регистрацию.	
б) за регистрацию в Реестре ПрЭВМ, Реестре БД и Реестре ТИМС сведений об изменении обладателя исключительного права на основании зарегистрированного договора или иного правоустанавливающего документа и за публикацию указанных сведений в официальном бюллетене	2600 руб.

В том случае если заявителями при регистрации отчуждения исключительного права являются несколько лиц, то государственная пошлина уплачивается ими самостоятельно в равных долях (п. 2 ст. 333.30 НК РФ).

Справочно: Госпошлина за регистрацию отчуждения исключительного права по договору с одним объектом составляет 10100 руб.

Государственные пошлины за государственную регистрацию перехода исключительного права без договора уплачиваются по сумме подпунктов 5) и 6) п. 1 ст. 333.30.

5) за гос. регистрацию перехода исключительного права на зарегистрированную ПрЭВМ, БД и ТИМС к другим лицам без договора	800 руб.
б) за регистрацию в Реестре ПрЭВМ, Реестре БД и Реестре ТИМС сведений об изменении обладателя исключительного права на основании зарегистрированного договора или иного правоустанавливающего документа и за публикацию указанных сведений в официальном бюллетене	2600 руб.

При заполнении платёжного документа необходимо руководствоваться положениями приказа Минфина от 24.11.2004 № 106н (в редакции Приказа Минфина от 30.12.2010 № 197н). Обязательно указывается: 1) плательщик и его реквизиты, 2) получатель платежа и его реквизиты, 3) в назначении платежа указывается: Госпошлина, название ПрЭВМ (БД, ТИМС, договора), пункты НК РФ, в соответствии с которыми уплачивается госпошлина, а также указывается: «НДС не облагается».

При уплате госпошлины организацией в платёжном документе в поле 104 обязательно указывается КБК, в поле 105 указывается ОКТМО, в поле 24 указывается назначение платежа. В полях 106-110 ПД в соответствии с требованиями Приказа Минфина от 24.11.2004 № 106н (в редакции Приказа Минфина от 30.12.2010 № 197н) проставляется ноль «0». Наличие в ПД незаполненных полей не допускается.

Образец заполнения полей в платёжном поручении при уплате госпошлины за регистрацию ПрЭВМ организацией (для других объектов вместо ПрЭВМ указывается БД или ТИМС):

Поле 104	Поле 105	Поле 106	Поле 107	Поле 108	Поле 109	Поле 110
16810809000011000110	45318000	0	0	0	0	0

Поле 24 (назначения платежа)

1.1.333.30 НК РФ. Госпошлина за государственную регистрацию ПрЭВМ «[Название объекта регистрации указывается в объёме, достаточном для идентификации]». НДС не облагается.

Льготы по уплате государственной пошлины

(льготы установлены Федеральным законом Российской Федерации от 27 декабря 2009 г. № 374-ФЗ "О внесении изменений в статью 45 части первой и в главу 25³ части второй Налогового кодекса Российской Федерации ...")

1. В соответствии с **подпунктом 4 пункта 1 статьи 333.35 НК РФ** от уплаты госпошлины, освобождаются федеральные органы государственной власти, органы государственной власти субъектов Российской Федерации и органы местного самоуправления при их обращении за совершением юридически значимых действий, установленных настоящей главой (25³) Кодекса.

2. В соответствии с **подпунктом 14 пункта 1 статьи 333.35 НК РФ** от уплаты госпошлины, освобождаются физическое лицо - гражданин Российской Федерации, являющийся единственным автором ПрЭВМ, БД, ТИМС и правообладателем на нее, испрашивающим свидетельство о регистрации на свое имя, в случае, если такое физическое лицо является ветераном Великой Отечественной войны, инвалидом, учащимся (воспитанником) образовательных учреждений (независимо от их форм собственности), - за совершение действий, предусмотренных подпунктами **1-3, 5 и 6 пункта 1 статьи 333.30 настоящего Кодекса**.

Льгота, предусмотренная настоящим подпунктом, предоставляется также коллективу авторов, правообладателей, **каждый член которого** является инвалидом, либо участником Великой Отечественной войны, либо инвалидом Великой Отечественной войны.

3. Основанием для предоставления льгот физическим лицам, перечисленным в подпункте 14 пункта 1 статьи 333.35 НК РФ, является **удостоверение установленного образца**. Льготы, предусмотренные подпунктом 14 пункта 1 статьи 333.35 НК РФ, предоставляются **по ходатайству автора** (авторов). Основанием для предоставления льготы являются копии соответствующих документов: **удостоверения ветерана Великой Отечественной войны (участника войны), справки медико-социальной экспертизы, документа, выданного образовательным учреждением**. Ходатайство о предоставлении указанных льгот подается вместо документа, подтверждающего уплату госпошлины, если льготой является освобождение от ее уплаты, или вместе с указанным документом.

Возврат или зачёт государственной пошлины

Порядок возврата или зачета государственной пошлины осуществляется в случаях и порядке, предусмотренных положениями статьи 333.40 Налогового кодекса Российской Федерации.

**Реквизиты счета для уплаты государственных пошлин
 за совершение действий по официальной регистрации программы для электронных
 вычислительных машин, базы данных и топологий интегральных микросхем**

Для учета поступлений федерального бюджета Российской Федерации, уплачиваемых в соответствии со статьей 333.30 Налогового кодекса Российской Федерации (далее - НК РФ):

по КБК 168 1 08 09000 01 1000 110 - «Государственная пошлина за совершение уполномоченным федеральным органом исполнительной власти действий по официальной регистрации программы для электронных вычислительных машин, базы данных и топологий интегральных микросхем».

При заполнении распоряжений о переводе денежных средств в уплату платежей в бюджетную систему Российской Федерации плательщики руководствуются Приказом Минфина России от 12 ноября 2013 г. № 107н «Об утверждении Правил указания информации в реквизитах распоряжений о переводе денежных средств в уплату платежей в бюджетную систему Российской Федерации» следующим образом:

поле 13 «Банк» получателя	Операционный департамент Банка России г. Москва 701
поле 14 «БИК» банка получателя	044501002
поле 15 «Корреспондентский счет» банка получателя	не заполняется
поле 16 «Наименование» получателя	Межрегиональное операционное УФК (Федеральная служба по интеллектуальной собственности)
поле 17 «Счет» получателя	40101810500000001901
поле 22 «Код»	указывается идентификатор начисления
поле 61 «ИНН» получателя	7730176088
поле 103 «КПП» получателя	773001001
поле 104	168108090000011000110
поле 105 ОКТМО	45318000
поле 108	указывается идентификатор сведений о физическом лице в качестве основных идентификаторов сведений о физическом лице используются: - паспорт гражданина Российской Федерации; - страховой номер индивидуального лицевого счета застрахованного лица в системе персонализированного учета Пенсионного фонда Российской Федерации (СНИЛС); - водительское удостоверение.
поля 106, 107, 109 и 110	проставляется ноль «0»
поле 24 «Назначение платежа»	указывается подпункт пункта 1 статьи 333.30 НК РФ и название объекта регистрации (программы для ЭВМ, базы данных и топологии интегральной микросхемы) в объеме, достаточном для идентификации, а также имя или наименование заявителя.
СПРАВОЧНО: код ОКПО	00038971
ОГРН (администратора платежей)	1047730015200 от 24.06.04

Наличие в расчетном документе незаполненных полей **не допускается** (кроме поля 15).

Распоряжение о переводе денежных средств, подтверждающее уплату государственной пошлины, должно относиться только к одному заявителю (плательщику) и одному объекту регистрации. В том случае если заявителями являются несколько лиц, то государственная пошлина уплачивается ими самостоятельно в равных долях в соответствии с пунктом 2 статьи 333.30 НК РФ.

Список литературы

1. Федеральная служба по интеллектуальной собственности (Роспатент). URL: <http://www.rupto.ru/about>
2. Программы для ЭВМ, БД / ФИПС. URL: http://www1.fips.ru/wps/wcm/connect/content_ru/ru/soft/
3. Открытые реестры / ФИПС. URL: <http://www1.fips.ru/wps/portal/Registers/>
4. Образцы заявок, заявлений и ходатайств и примеры их заполнения / ФИПС.
URL: http://www1.fips.ru/wps/wcm/connect/content_ru/ru/file_library/obr_zaya#7

Учебное издание

Руслан Иванович Баженов, Ольга Юрьевна Пронина

КАК СТУДЕНТУ ЗАРЕГИСТРИРОВАТЬ СОБСТВЕННУЮ ПРОГРАММУ ДЛЯ ЭВМ В РОСПАТЕНТЕ

Учебно-методическое пособие

Печатается в авторской редакции
Технический редактор Р.И. Баженов

Подписано в печать
Формат издания 60х90 1/16.
Усл. печ. л. 4 . Уч.-изд. л. 1,39
Тираж 300 экз. Заказ № ____/2017

Издательство Федерального государственного бюджетного образовательного учреждения
высшего образования

«Приамурский государственный университет имени Шолом-Алейхема»
679015, г. Биробиджан, ул. Широкая, 70-А

Типография Федерального государственного бюджетного образовательного учреждения
высшего образования

«Приамурский государственный университет имени Шолом-Алейхема»
679015, г. Биробиджан, ул. Широкая, 70-А