

Бытко С.Ю., Миткин С.Б. Правовая информационная система «ДРАКОН-право» на базе алгоритмического визуального языка «ДРАКОН» // Проблемы и вызовы цифрового общества: тенденции развития правового регулирования цифровых трансформаций / под ред. Н. Н. Ковалёвой / в 2 т. Т. 1. — Саратов: Изд. «Саратовская гос. юридическая академия», 2020. — 456 с. — С. 322-334.

Сведения об авторах

Бытко Сергей Юрьевич — доктор юридических наук, доцент, профессор прокурорского надзора и криминологии Саратовской государственной юридической академии.

Митькин Степан Борисович — программист, разработчик программного обеспечения Правовой информационной системы «ДРАКОН-право».

7.6. Правовая информационная система «ДРАКОН-право» на базе алгоритмического визуального языка «ДРАКОН»

Цифровые технологии в сфере информатизации права

В настоящее время руководство России заявляет о необходимости внедрения цифровых технологий во все сферы жизни страны (в последнее время в этой связи часто используется термин «цифровизация»). Об этом, в частности, неоднократно заявлял Президент России В. В. Путин в ряде своих выступлений¹.

Термин «цифровые технологии» является обобщенным и конкретное его наполнение зависит от сферы применения таких технологий. Нас в данном случае интересуют вопросы их внедрения в сферу правоприменения. И, чтобы говорить о развитии, в первую очередь, нужно оценить текущее состояние дел в сфере информатизации права. И здесь мы можем говорить о значительном прогрессе. В первую очередь, речь идет о базах данных (иногда говорят об информационных банках) правовой информации, созданных и пополняемых такими компаниями, как «Гарант», «КонсультантПлюс». В настоящее время ими накоплены огромные массивы нормативных актов во всех отраслях права, проведена масштабная работа по систематизации баз данных, повышению скорости и удобства поиска (для каждого нормативного акта формируется оглавление, расставляются гиперссылки для связи между частями одного документа или между различными документами, предоставляется доступ к различным редакциям нормативного акта, научно-практическим комментариям, предоставляются сведения об источниках опубликования, дате вступления в законную силу). Усиленно развиваются такие направления, как формирование баз данных судебных решений по отдельным категориям дел, конструкторов, облегчающих подготовку различных правовых документов, правовых калькуляторов, позволяющих рассчитывать размеры пошлин, пенсий, трудового стажа и т.п.

На государственном уровне создан официальный интернет-портал правовой информации, в котором аккумулируются действующие редакции нормативных актов, информация о законопроектах, официальная электронная версия Собрания законодательства Российской Федерации.

Существенное значение для правоприменительной практики имеют информационные материалы, размещаемые на сайтах правоохранительных органов

¹ Владимир Путин: Внедрить цифровые технологии во все сферы жизни. Выступление на пленарной дискуссии на Петербургском экономическом форуме 4 июня 2017 г. [Электронный ресурс]. URL: <https://rg.ru/2017/06/04/reg-szfo/vladimir-putin-vnedrit-cifrovye-tehnologii-vo-vse-sfery-zhizni.html> (дата обращения: 24.03.2020); Путин: формирование цифровой экономики – вопрос нацбезопасности РФ [Электронный ресурс]. URL: <https://tass.ru/ekonomika/4389411> (дата обращения: 24.03.2020).

и судов, а также данные уголовной статистики. В своей совокупности эти информационные ресурсы предоставляют огромные возможности для анализа практики применения правовых норм и направлений их совершенствования.

Следует упомянуть также о так называемых автоматизированных местах, например «Аrm Следователя» – доставляющих следователю готовые шаблоны процессуальных документов, алгоритмы действий в отдельных ситуациях, возможности автоматизированного заполнения статистических карт, справочную информацию по нормативным актам и т. п.

С учетом изложенного можно сделать вывод, что процессы информатизации в правоприменительной сфере направлены в, первую очередь, на обеспечении пользователя исчерпывающей правовой информацией по интересующему вопросу.

Проблемы имеющихся правовых информационных систем

Увеличение объемов накапливаемой в базах данных правовой информации парадоксальным образом снижает эффективность работы пользователей.

Рассмотрим в качестве примера назначение вида исправительного учреждения лицу, осужденному к наказанию в виде лишения свободы. Соответствующий запрос на естественном языке может звучать так: «Какой вид исправительного учреждения будет назначен осужденному к наказанию в виде лишения свободы». Если ввести его в строке поиска СПС «Гарант»¹, мы получим (по состоянию на 25 марта 2020 г. подборку из 15295 документов. Причем на первом месте в выдаче будет ссылка на ч. 4 ст. 80 УИК РФ «Раздельное содержание осужденных к лишению свободы в исправительных учреждениях», затем в результатах выдачи следует семь комментариев к ст. 58 УК РФ «Назначение осужденным к лишению свободы вида исправительного учреждения», затем материалы судебной практики, перемежающийся комментариями законодательства. Непосредственный источник ответа на данный вопрос, ст. 58 УК РФ, в списке результатов находится под номером 30, на второй странице результатов поиска.

Если же искать точное название статьи 58 УК РФ, то результат поиска будет гораздо «компактнее», всего 1 839 документов, причем вначале опять следует ряд комментариев законодательства, а ссылка на ст. 58 УК РФ следует лишь 9-м пунктом в списке. Если даже в списке найденных документов выбрать категорию «Акты органов власти», то и в этом случае на первом месте будет ссылка на ч. 4 ст. 80 УИК РФ, а ст. 58 УК РФ – на втором месте. Однако, следует учесть, что подобный результат достигается лишь при указании точного названия ст. 58 УК РФ, что является заведомо невозможным для неспециалиста в области права.

¹ Имеется в виду общедоступная интернет-версия на сайте www.garant.ru.

При работе с общедоступной версией СПС «Консультант плюс» результат несколько отличается. В первом варианте запроса на естественном языке результат включает в себя всего 29 документов, УК РФ находится всего на 7 месте, что значительно удобнее. В то же время по ссылке открывается не текст статьи 58, а весь Уголовный кодекс, что снижает ценность полученного ответа. Справедливости ради нужно отметить, что выдача результатов СПС «Консультант плюс» является компактной потому, что не содержит ссылок на комментарии законодательства, обильно представленные в аналогичных результатах поиска в системе «Гарант».

Впрочем, вряд ли справедливо упрекать разработчиков упомянутых правовых информационных систем в низком качестве работы. Для своих целей они достаточно совершенны и для специалиста в области права являются огромным подспорьем. Однако избыток информации зачастую препятствует быстрому получению ответа. Так, при наличии пробелов в праве или при неопределенности правовой нормы специалист вынужден обращаться к поиску в материалах судебной практики, разъяснениям в теоретических работах и т.д. Если же пользователь не обладает специальными познаниями в юриспруденции, он не сможет правильно сформулировать корректный поисковый запрос.

Большое количество документов в результатах поиска не является самой значительной проблемой. Предположим, пользователь нашел необходимую ему ст. 58 УК РФ и пытается самостоятельно определить – какой вид исправительного учреждения будет избран в интересующем его случае. Сможет ли он самостоятельно найти ответ? Полагаем, что если и сможет, то далеко не сразу, поскольку правильное применение ст. 58 УК РФ предполагает знание таких уголовно-правовых понятий, как категория преступления, форма вины, сроки погашения судимости, сроки давности привлечения к уголовной ответственности, рецидив и его виды. Отсутствие же специальной подготовки приведет к необходимости длительно разбираться с соответствующими понятиями.

Сложность задачи многократно возрастает с учетом того, что ст. 58 УК РФ в ее текущем виде содержит ряд пробелов, возникших после внесения изменений в ст. 18 УК РФ, вследствие которых при рецидиве преступлений перестали учитываться судимости за преступления небольшой тяжести. В подобных случаях необходимо изучение дополнительной информации, в частности рекомендаций, содержащихся в Постановлении Пленума Верховного Суда РФ, положения ст. 74 УИК РФ «Виды исправительных учреждений».

Конечно, квалифицированные юристы с большим стажем работы запоминают решение часто встречающихся в их практике сложных задач, однако, полагаем, и они периодически сталкиваются с проблемами, требующими изучения множества нормативных актов.

Программное обеспечения для разрешения типовых правовых ситуаций

Приведенный пример показывает перспективное направление развития информационных правовых систем, которое сегодня практически не охвачено вниманием разработчиков информационных правовых систем – разработку алгоритмов разрешения типовых правовых ситуаций, когда все нюансы взаимодействия отдельных норм права между собой реализованы алгоритмически, от пользователя не будет требоваться самостоятельного анализа текстов нормативных актов. Ему необходимо лишь отвечать на ряд простых вопросов, не требующих для ответа специальных знаний. В настоящее время готовых программных комплексов подобного рода не имеется, но уже наметились два основных подхода к их построению.

Первый из них базируется на использовании нейронных сетей (так называемого «искусственного интеллекта»), принцип работы которых состоит в распознавании в потоке входных данных объектов (образцов, образов), узнавать которые нейронную сеть обучают заранее. Второй подход состоит в создании программ, в которых заранее предусмотрены все возможные решения правовой проблемы в зависимости.

Оба подхода имеют свои особенности, преимущества и недостатки, поэтому остановимся на них подробнее.

Нейронные сети

Нейронные сети получили такое название постольку, поскольку разработчики попытались симитировать работу биологических нейронных сетей. В последнее время нейросети получили широкое распространение в сфере распознавания образов (например, в системах автоматической фиксации нарушений правил дорожного движения), речи или текста на естественном языке. Программе предлагается для обучения большое число заранее подготовленных образцов (например, изображений какого-либо животного, отдельных символов алфавита). После обучения нейросеть способна самостоятельно узнавать соответствующие образы на предлагаемых ей изображениях. Несмотря на то, что осмысленной деятельностью компьютера при решении подобных задач назвать нельзя¹, вероятность правильного ответа очень высока. Так, «робосудья», модель, обученная на отдельных видах налоговых споров способна предсказать результаты их рассмотрения в суде с вероятностью 93 %². При этом сами раз-

¹ Например, если программа обучена распознавать определенный символ, который отсутствует на предлагаемом изображении, она, скорее всего подберет какой-то другой, отдаленно похожий на образец, которому она обучена.

² Как создали робосудью и почему не стоит ждать скорых революций? [Электронный ресурс]. URL: [https://vc.ru/tribuna/46093-kak-sozdali-robosudyu-i-pochemu-ne-stoit-zhdat](https://vc.ru/tribuna/46093-kak-sozdali-robosudyu-i-pochemu-ne-stoit-zhdat-revoluciy)

работчики отмечают ряд особенностей нейросетей, которые ограничивают области их применения:

- высокая эффективность лишь на узких задачах, решению которых нейросеть заранее обучена;
- нейросеть не понимает, что важно для людей, а что неважно, поэтому ее можно обучить вынесению как правосудных, так и неправосудных решений;
- обучение нейросети требует трудоемкой подготовки данных, поскольку текстов судебных актов на «естественном» языке машина не понимает.

С точки зрения права одним из главных недостатков нейросетей является то, что мы не всегда можем доверять принятому ей решению. Как отмечают некоторые исследователи, нейросеть в редких случаях может генерировать различные решения для похожих, с точки зрения человека, наборов данных. Поэтому ее решение в строгом смысле слова не является доказательным¹. Другим недостатком нейронных сетей, на наш взгляд, является то, что получаемое решение является не алгоритмическим, не очевидным. Поясним на примере. В марте 2016 г. состоялся матч по игре го между Ли Седодем, одним из сильнейших игроков в го и программой AlphaGo, действующей для вычисления следующего хода нейронные сети. Игра го интересна тем, что требует от игрока не только логики, но и интуиции, творческого мышления. В силу этого создать сильную программу для этой игры на основе традиционного алгоритмического программирования считается невозможным. По результатам матча нейросетевые алгоритмы показали свою исключительную эффективность: программа победила со счетом 4:1. В ходе матча комментаторы часто не понимали логики игры машины и, даже, не всегда могли понять, на чьей стороне перевес: настолько необычно и нестандартно играла программа. Во время матча приглашенный программист, один из разработчиков программы, Тор Греспел, на просьбу прокомментировать игру машины заявил: «хотя мы и программировали эту машину, мы не знаем, какой ход она сделает»².

Таким образом, даже если решение нейросетевого алгоритма будет правильным, понять, как оно принято не всегда представляется возможным. Подобный способ принятия решений плохо совместим с судебными процедурами, в которых участники процесса, обосновывают свою позицию опираясь на нормы права, основываясь на правилах логики, уясняют содержание правовой

skoryh-revolyuciy (дата обращения: 26.03.2020).

¹ James M. The Flaw Lurking In Every Deep Neural Net [Электронный ресурс]. URL: <https://www.i-programmer.info/news/105-artificial-intelligence/7352-the-flaw-lurking-in-every-deep-neural-net.html> (дата обращения: 26.03.2020).

² URL: <https://www.wired.com/2016/03/googles-ai-wins-pivotal-game-two-match-go-grandmaster/> [Электронный ресурс]. URL: <https://www.wired.com/2016/03/googles-ai-wins-pivotal-game-two-match-go-grandmaster/> (дата обращения: 25.05.2020).

нормы используя различные методы толкования для того, чтобы убедить судью в своей правоте.

Впрочем, сказанное не означает, что нейросети не должны применяться в информационных правовых системах. Полагаем, что они будут незаменимы при обработке неструктурированной текстовой информации (например, фабул уголовных дел), вычленивая элементы, имеющие правовое значение, поиске судебных прецедентов, рассмотрении вопросов, базирующихся на оценочных признаках и в ряде других, слабо алгоритмизируемых направлениях.

Алгоритмизация норм права

Другой подход состоит в разработке программ, предлагающих разрешение правовых вопросов на основе алгоритмического программирования, когда программист заранее описывает какой результат будет выдаваться пользователю в зависимости от полученных входных данных.

В отличие от нейросетей данный подход реализует доказательное решение правовой задачи, поскольку каждый шаг алгоритма поддается объективной проверке путем анализа исходного кода программы и сопоставления его с логикой применяемых правовых норм. Доказательность, проверяемость предлагаемых решений является, на наш взгляд, главным достоинством традиционного алгоритмического подхода, делающая его предпочтительным для создания информационных правовых систем нового поколения. Если алгоритм строго следует логике правовой нормы, то результативность решения правовой проблемы равна 100 %, в то время как нейросети абсолютной точности не гарантируют.

Однако и на этом пути разработчиков ожидают немалые трудности. Прежде всего, стоит отметить высокую трудоемкость программирования, требующую тесного взаимодействия специалистов в той или иной отрасли права и программистов (что, впрочем, справедливо и для нейронных сетей), частые изменения законодательства, требующие правки множества уже созданных алгоритмов и таблиц исходных данных, пробелы в законодательстве и т. д. Подробное описание всех проблем, с которыми могут столкнуться разработчики информационных систем, заслуживает отдельной статьи. Здесь же хотелось остановиться на том, каким нам видится преодоление основной проблемы разработчиков – трудоемкости программирования и сложности взаимодействия юристов и программистов.

Происхождение языка ДРАКОН

Язык ДРАКОН был разработан в Федеральном космическом агентстве (Научно-производственном центре автоматизации и приборостроения имени академика Н. А. Пилюгина, г. Москва) и Российской академии наук (Институт прикладной математики имени академика М. В. Келдыша, г. Москва).

Автором данного языка является В.Д. Паронджанов, опубликовавший несколько книг, посвященных ДРАКОНу¹. Технические подробности и особенности языка широко освещены в указанных и других его монографиях. Здесь же хотелось отметить такие его ключевые особенности, как

- визуальное представление алгоритмов, кардинально упрощающих их понимание (“сверхвысокая понимаемость” в терминологии В.Д. Паронджанова);
- возможность написания программ на языке, приближенном к обычной речи и способность преобразовывать визуальные алгоритмы языка «ДРАКОН» в обычные программные коды.

Наглядность и простота языка «ДРАКОН» позволила писать программы, управляющие автоматизированной системой управления кораблем «Буран» не программистам, а инженерам, непосредственно проектирующим эту систему. Таким образом было достигнуто упрощение программирования, исключены ошибки, возникающие при передаче информации от узкого специалиста в сфере автоматизированных систем управления к программисту (последний был фактически исключен из процесса программирования), а также ошибки программирования, порождаемые усталостью, рассеянием внимания, характерным для написания программ традиционными способами. Одновременно возросла читаемость программ и легкость проверки кода на предмет правильности реализации заложенных в нем идей. Достоинства языка ДРАКОН делают его применение возможным в различных отраслях знания, в том числе, и в праве.

Основы языка ДРАКОН

Дракон-схема состоит из вертикально-ориентированных областей, в терминологии языка «ДРАКОН» – веток, отображающих законченные смысловые последовательности (см. рис 1). Первой выполняется самая левая ветка. Значки, расположенные на ветке, являются стандартными, имеющими одинаковое значение в любой схеме. Каждая ветка начинается сверху со значка «Имя ветки». Ниже расположено тело ветки – основная логическая последовательность, характерная для описываемой структуры. Значки в теле ветки следуют в направлении сверху-вниз. В самом низу находится значок «Адрес», в котором содержится имя следующей ветки, на которую должен быть осуществлен переход. Тело ветки содержит значки двух типов: приказы и вопросы. Значки-приказы предписывают совершение действий. Например, значок “Действие” с текстом “Рецидив = опасный рецидив” дает команду программе “считать

¹ См.: *Паронджанов В.Д.* Занимательная информатика. М. 2007; Его же. «Как улучшить работу ума. Алгоритмы без программистов - это очень просто!» М., 2001; Его же. «Почему мудрец похож на обезьяну. М., 2007; Его же. «Дружелюбные алгоритмы понятны каждому». М., 2010 и др.

данное преступление совершенным при опасном рецидиве”. Значки-вопросы осуществляют ветвление в алгоритме в зависимости от значений переменных. Например, значок “Вопрос” с текстом “Форма вины == умысел?” предписывает выбор дальнейшего пути в алгоритме в зависимости от того, является ли преступление умышленным или неосторожным.

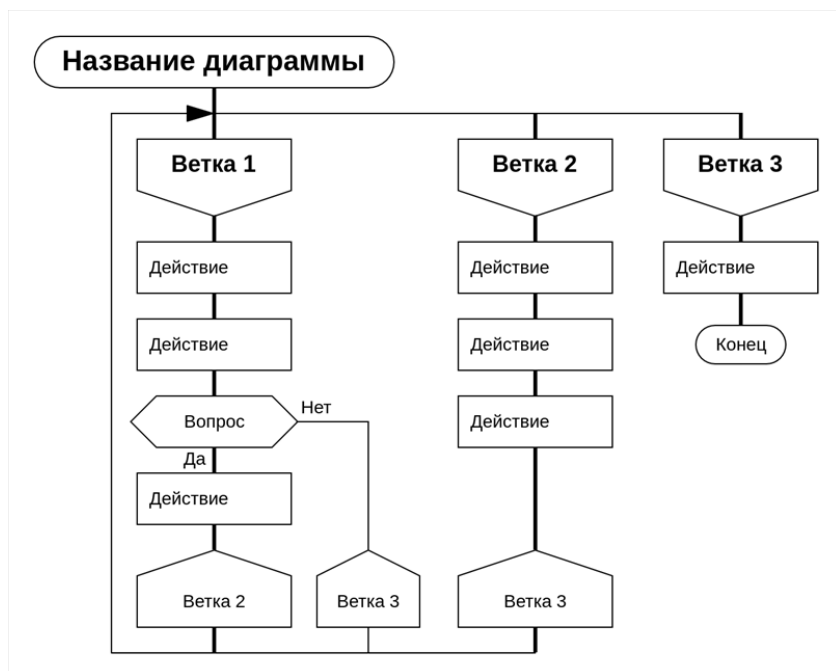
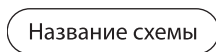


Рис. 1. Дракон-схема, состоящая из нескольких веток.

Наиболее часто используемые значки языка «ДРАКОН» выглядят следующим образом:



Название схемы

Название схемы – размещается всегда вверху слева



Ветка 1

Имя ветки – Название ветки алгоритма. Отражает в сжатом виде смысл выполняемых в данной ветви действий.



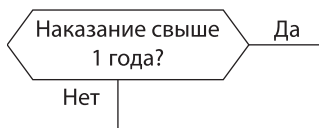
Ветка 2

Адрес – имя ветки, на которую нужно перейти.

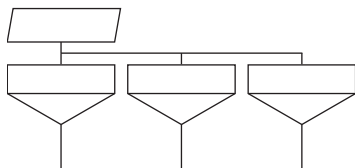


Действие

Действие – значок-приказ, содержащий простую команду.



Вопрос – значок, содержащий вопрос, на который можно ответить “да” или “нет”. В зависимости от ответа, значок Вопрос определяет направление дальнейшего движения по алгоритму.



Выбор – значок, содержащий вопрос, на который можно дать несколько ответов. В верхнем четырехугольнике указывается имя переменной. В фигурах под ним задаются возможные значения переменной и соответствующие значениям пути выполнения алгоритма.

В языке ДРАКОН принят следующий порядок построения схемы: наиболее желательная для пользователя последовательность событий размещается на вертикальной прямой в левой части ветки алгоритма. Худшие, менее предпочтительные варианты ответвляются вправо. Это правило называется “Чем правее, тем хуже”. Данное правило позволяет моментально увидеть, какие пути через алгоритм благоприятны для пользователя (влекут желаемые для него правовые последствия) а какие – нет.

Язык ДРАКОН имеет ряд преимуществ по сравнению с другими визуальными языками.

Во-первых, правила языка ДРАКОН тщательно продуманы с целью улучшить читаемость программ. Благодаря этим правилам дракон-схемы имеют опрятный и единообразный вид. Опрятность означает устранение зрительного шума из диаграмм. Примеры зрительного шума, характерные для традиционных блок-схем и отсутствующие в языке ДРАКОН: произвольный, случайный размер значков, пересечения линий, лишние стрелки, линии, нарисованные под углом или имеющие изгибы. В языке ДРАКОН значки имеют стандартную ширину, пересечения линий запрещены, вместо стрелок применяются простые прямые линии, либо вертикальные, либо горизонтальные. Единообразие достигается, в том числе, упорядочением значков сверху-вниз и упорядочением ветвления слева-направо.

Во-вторых, язык ДРАКОН имеет уникальные особенности, которых нет в других визуальных языках, например, разбиение больших алгоритмов на ветки и правило “Чем правее, тем хуже”.

Информационная правовая система “Дракон-Право”

Практика показала, что свойства языка ДРАКОН позволяют описывать процессы, далеко выходящие за рамки программирования. Специалисты в различных отраслях используют визуальные алгоритмы на этом языке для описания сложных процессов в области медицины, электроники, управленческой деятельности и т. п.

Авторы статьи, проанализировав накопленный опыт использования языка ДРАКОН, предприняли попытку применить его при создании информационной правовой автоматизированной системы “ДРАКОН-Право”. В рамках проведенного исследования было создано подмножество языка, которое содержит набор элементов, достаточных для описания нормативных актов и решения правовых задач. Типичный пример программы на этом языке приведен на рис. 2

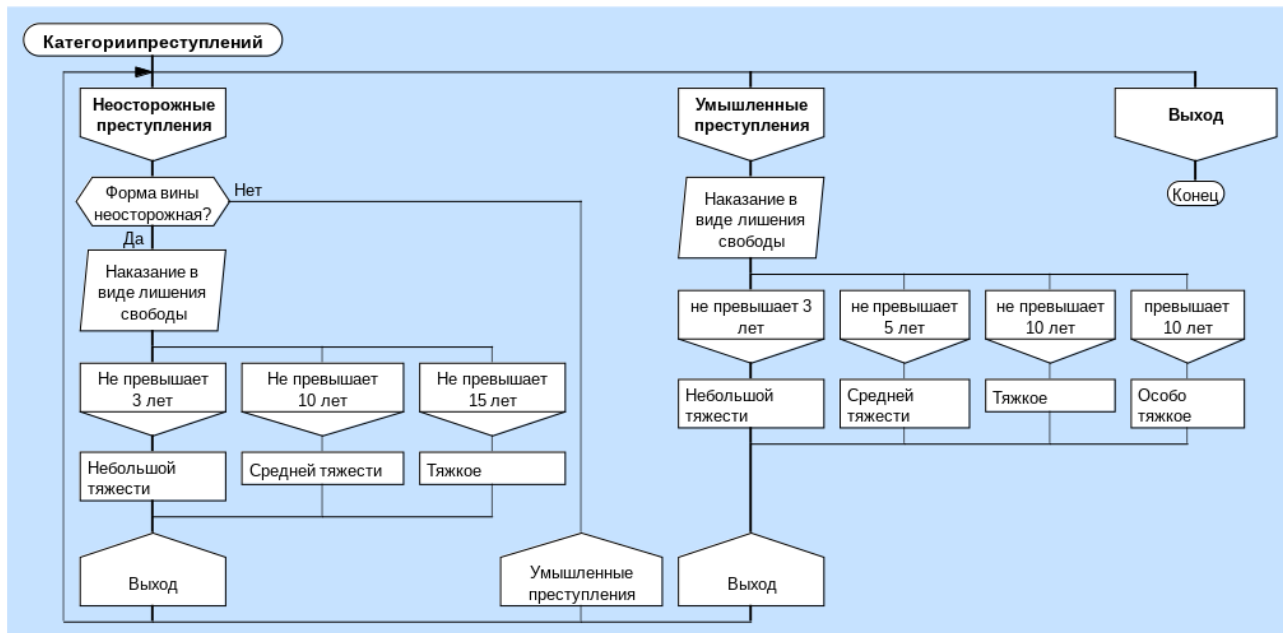


Рис. 2. Программа определения категории преступления на языке «ДРАКОН-Право»

Как работает ДРАКОН-право

Специалист в области права строит диаграммы подобного вида в специальной программе, редакторе схем на языке ДРАКОН. Редактор дракон-схем позволяет быстро создавать новые алгоритмы и легко редактировать уже имеющиеся. Кроме того, редактор дракон-схем обеспечивает автоматическое следование правилам языка ДРАКОН. Далее диаграммы автоматически преобразуются в алгоритмы, пригодные для запуска на компьютере, и загружаются на сайт автоматизированной информационно-правовой системы «ДРАКОН-Право», расположенной по адресу drakon-pravo.ru/. Там указанные алгоритмы выполняются по запросу пользователя.

На рис. 2 представлена небольшая схема, дающая ответ только на один простой вопрос. Система «ДРАКОН-Право» позволяет вызывать одни дракон-схемы внутри других дракон-схем и таким образом значительно расширять круг решаемых задач. Например, результаты выполнения алгоритма, представленного на рис. 1 могут быть переданы в алгоритм, определяющий вид рецидива преступлений, который, в свою очередь, может быть вызван из алгоритма, определяющего вид исправительного учреждения, назначаемого осужденному к лишению свободы, и т.д.

При выполнении алгоритма система «ДРАКОН-Право» взаимодействует с пользователем в форме вопросов и ответов. Алгоритм определяет, какие именно вопросы система задаст пользователю. Эти вопросы, как правило, не требуют для ответа специальной квалификации. В результате выполнения программы пользователю выдается отчет по интересующему его вопросу с комментарием и ссылкой на конкретную статью нормативного акта по анализируемой проблеме. Пользователь не обязан вслепую верить системе, так как ссылки на нормативные акты дают пользователю возможность понять, на основании чего система пришла к тому или иному выводу. Для анализа всегда доступно и визуальное изображение алгоритма (дракон-схемы), позволяющее проверить его соответствие положениям нормы права.

Преимущества системы «Дракон-Право»

Таким образом правовая информационная система «ДРАКОН-Право» характеризуется следующими ключевыми особенностями:

1. Простота и скорость использования автоматизированной правовой системы «Дракон-Право» для пользователей системы. Взаимодействие с программой осуществляется путем ответа на простые вопросы.

2. Получаемые решения являются полностью доказательными. Пользователю выдается ответ со ссылкой на конкретные статьи нормативных актов по его запросу. При необходимости правильность ответа может быть проверена путем анализа визуального алгоритма (дракон-схемы).

3. Легкость и быстрота создания юридических алгоритмов. Создание схем в системе «ДРАКОН-Право» не требует знания какого-либо языка программирования и не требует досконального знания всех правил языка ДРАКОН. От составителя требуется лишь понимание логики правовых норм и институтов.

4. Простота сопровождения написанных программ. После внесения изменения в правовые нормы, разработчик схем может быстро вносить правки в соответствующие алгоритмы. Ему не приходится разбираться в ранее написанном коде, как программисту на традиционном языке программирования.

5. «ДРАКОН-Право» позволяет выявлять пробелы в нормативных актах. Особенности языка таковы, что попытка создать схему, содержащую пробел, вызывает ошибку. С точки зрения права это означает, что если в норме права указывается характеристика какого-либо параметра, то беспробельная норма должна реализовывать применение права при всех вариантах значения. Например, использование при назначении вида исправительного учреждения таких характеристик субъекта, как пол, отбывание ранее наказания в виде лишения свободы, совершение преступления при рецидиве и т. д., предполагает, что правовая норма предусматривает ответ на любую комбинацию указанных характеристик.

Указанная особенность языка «Дракон-Право» позволяет использовать его в нормотворческом процессе, при создании гарантировано беспробельных норм права.

6. Связанность алгоритмов между собой. Разработчик юридических алгоритмов на языке «ДРАКОН-Право» видит схемы, связанные с той схемой, над которой он работает в настоящий момент. Это позволяет быстрее выявить все алгоритмы, подлежащие изменению, для приведения базы знаний в соответствие с изменениями законодательства.

7. Исключается риск внесения ошибок в алгоритм, характерный для традиционного взаимодействия программиста, который реализует некий процесс в программном коде, со специалистом, разъясняющего программисту особенности этого процесса.