|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт комплексной безопасности и специального приборостроения | | |
| Кафедра аппаратного, программного и математического обеспечения вычислительных систем | | |

|  |  |
| --- | --- |
| **РАБОТА ДОПУЩЕНА К ЗАЩИТЕ** | |
| Заведующий кафедрой |  |
|  | *Подпись* |
| Кулагин Владимир Петрович | |
| *ФИО* | |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г. | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| по направлению подготовки бакалавра | | 01.03.04 |  | Прикладная математика |
|  | | *Код направления подготовки* |  | *Наименование* |
| Математическое моделирование в экономике и технике | | | | |
| *направления подготовки* | | | | |
| на тему: | **Решающие деревья в принятии управленческих решений** | | | |
|  |  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | *\_\_\_\_\_\_\_\_\_\_* Котов Святослав Павлович*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | |
|  | *Подпись* | *Фамилия Имя Отчество* | |
| Шифр | \_\_\_\_150261\_\_\_\_ |  |  |
| Группа | \_ТМБО-01-15\_\_ |  |  |
|  |  |  |  |
| Руководитель  работы | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | *\_\_\_\_\_\_*д. ф.-м. н., профессор *\_\_\_\_\_\_\_* | *\_\_\_\_\_*Шамин Р. В.*\_\_\_\_* |
|  | *Подпись* | *Ученая степень, ученое звание, должность* | *ФИО* |
|  | |  |  |

Москва 2019 г.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт комплексной безопасности и специального приборостроения | | |
| Кафедра аппаратного, программного и математического обеспечения вычислительных систем | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СОГЛАСОВАНО | | |  | УТВЕРЖДАЮ | | |
| Заведующий  кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Подпись* | | |  | Директор  института \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Подпись* | | |
| \_\_\_\_\_\_Кулагин Владимир Петрович\_\_\_\_\_  *ФИО* | | |  | \_\_Снедков Александр Борисович\_\_\_\_  *ФИО* | | |
| « » | \_\_\_\_\_\_\_\_\_\_ | 20 г. |  | « » | \_\_\_\_\_\_\_\_\_\_\_ | 20\_\_\_ г. |

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы бакалавра

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_\_\_\_Котов Святослав Павлович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
|  | *Фамилия Имя Отчество* | | |
| Шифр | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_150261\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Направление подготовки | | \_\_\_\_\_01.03.04\_\_\_\_ | \_\_\_Прикладная математика\_\_\_ |
|  | | *Код направления подготовки* | *Наименование* |
| Математическое моделирование в экономике и технике | | | |
| *направления подготовки*  *Наименование направления подготовки* | | | |
| Группа | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ТМБО-01-15\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |

1. Тема выпускной квалификационной работы бакалавра:

**Решающие деревья в принятии управленческих решений**

2. Цель и задачи выпускной квалификационной работы бакалавра

Цель работы: изучение алгоритма решающих деревьев и его программная реализация для применения на предприятиях, чтобы построить оптимальные управленческие решения.

Задачи работы: рассмотреть метод решающих деревьев и разработать программное решение для принятия оптимальных управленческих решений на предприятиях.

3. Этапы выпускной квалификационной работы бакалавра:

|  |  |  |  |
| --- | --- | --- | --- |
| №  этапа | Содержание этапа выпускной квалификационной работы | Результат выполнения этапа ВКР | Срок выполнения |
| 1 | Введение | Актуальность темы | 10.05.2019 |
| 2 | Исследовательский раздел | Анализ предметной области и постановка задачи | 10.05.2019 |
| 3 | Специальный раздел | Исследование математических методов | 17.05.2019 |
| 4 | Технологический раздел | Разработка программной реализации | 17.05.2019 |
| 5 | Экономический раздел | Экономические перспективы программной реализации | 24.05.2019 |
| 6 | Заключение | Основные результаты и выводы | 24.05.2019 |

4. Перечень разрабатываемых документов и графических материалов:

4.1 Презентация PowerPoint

4.2 Раздаточный материал

5. Руководитель выпускной квалификационной работы бакалавра:

|  |  |  |  |
| --- | --- | --- | --- |
| Функциональные обязанности | Должность в Университете | Фамилия Имя Отчество | Подпись |
| Руководитель ВКР | профессор | Шамин Р. В. |  |

|  |  |
| --- | --- |
| Задание выдал | Задание принял к исполнению |
| Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Обучающийся: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *Подпис*ь | *Подпис*ь |
| «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. | «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. |

**АННОТАЦИЯ**

на выпускную квалификационную работу бакалавра на тему

**«Решающие деревья в принятии управленческих решений»**

студента МИРЭА – Российского технологического университета

Института комплексной безопасности и специального приборостроения

группы ТМБО-01-15

**Котова Святослава Павловича**

**Цель данной работы:** изучение алгоритма решающих деревьев и его программная реализация для применения на предприятиях, чтобы построить оптимальные управленческие решения.

Работа состоит из введения, четырех разделов, заключения и списков использованных источников.

Квалификационная работа содержит 47 страниц машинописного текста (без приложений), 1 таблицу, 10 рисунков, 5 приложений, 13 наименований списка использованных источников.

**Ключевые слова:** решающие деревья, принятие решений, деревья решений, расчёт решающих деревьев, построение решающих деревьев

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc11868430)

[1. Исследовательский раздел 5](#_Toc11868431)

[1.1. Основные понятия и определения метода дерева решений 5](#_Toc11868432)

[1.2. Сущность метода дерево решений 7](#_Toc11868433)

[1.3. Область применения решающих деревьев 13](#_Toc11868434)

[1.4. Алгоритм построения решающих деревьев 15](#_Toc11868435)

[1.5. Преимущества и недостатки решающих деревьев. 18](#_Toc11868436)

[2. Специальный раздел 22](#_Toc11868437)

[2.1. Постановка задачи 22](#_Toc11868438)

[2.2.Выполнение математического расчёта методом дерева решений 23](#_Toc11868439)

[3. Технологический раздел 32](#_Toc11868440)

[3.1. Описание программной реализации 32](#_Toc11868441)

[4. Экономический раздел 48](#_Toc11868442)

[4.1. Оценка экономических перспектив разработанного программного обеспечения 48](#_Toc11868443)

[ЗАКЛЮЧЕНИЕ 51](#_Toc11868444)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 52](#_Toc11868445)

[ПРИЛОЖЕНИЕ А. Листинг программного кода 54](#_Toc11868446)

[ПРИЛОЖЕНИЕ Б. Презентация 68](#_Toc11868447)

[ПРИЛОЖЕНИЕ В. Результат проверки на оригинальность текста 83](#_Toc11868448)

[ПРИЛОЖЕНИЕ Г. Заключение руководителя об оригинальности текста 86](#_Toc11868449)

[ПРИЛОЖЕНИЕ Д. Отзыв научного руководителя 87](#_Toc11868450)

# **ВВЕДЕНИЕ**

**У**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**4**

*ВКР2068717-010304-КБ5-19-18*

Студент

Котов С.П.

Руководит.

Шамин Р.В.

Консульт.

Шамин Р.В.

Н. Контр.

Галёмина Е.А.

Зав. Каф.

Кулагин В.П.

Решающие деревья в принятии управленческих решений

Лит.

Листов

ТМБО-01-15

Актуальность темы исследования обоснована тем, что метод дерева решений имеет большое значение для разработки эффективных управленческих решений. Метод решающих деревьев является в последнее время популярным, он позволяет выбрать наиболее эффективные альтернативы.

Этот метод помогает оптимизировать многие процессы в стратегии развития предприятия. Дерево решений помогает выявить доминирующие факторы, оказывающие влияние на предприятие.

Своевременная разработка и принятие правильного решения – главные задачи работы управленческого персонала любой организации. Применяют метод решающих деревьев в условиях неопределённости, когда каждое решение зависит от исхода предыдущего решения или исходов испытаний.

В России данный метод применяется не так давно, поэтому является слабо изученным. На практике исследований по данному методу недостаточно, поэтому для российских предпринимателей очень важно, чтобы метод лучше изучен, и чтобы они видели результаты его применения на практике.

Метод решающих деревьев достаточно широко применяется во многих компаниях. С помощью него можно составить прогноз для различных направлений науки, технологий, техники, для профессиональных и личных целей. Этот метод подходит практически для любого предприятия. Благодаря этому методу предприятие сможет спрогнозировать свои действия, выбрать правильную стратегию развития. Метод помогает согласовать перспективные цели и конкретные задачи на разных уровнях иерархии.

# **Исследовательский раздел**

## **Основные понятия и определения метода дерева решений**

История возникновения метода дерева решений начинается с 1957 года. Американские исследователи Ч. Черчмен и Р. Акофф впервые предложили идею метода. Предложенная схема была похожа на дерево, из-за этого метод так и назвали «дерево решений» или решающие деревья [1].

В дальнейшем многие ученые внесли свой вклад в доработку и усовершенствование метода. Работы велись в разных направлениях. Такие известные ученые как Ф. В. Тейлор, В. Врум, М. Крауз, Г. Саймон разработали статегию применения этого метода в научном управлении. М. Вебер в группе с другими учеными разработали бюрократическую теорию. Х. Файоль, Л. Урвик, Дж., Д. Лизней нашли применение методу решающих деревьев в принятии решений в организациях. Эти три направления применения метода составили три главных раздела классической теории [2]. Причем, все авторы классической теории имеют свой индивидуальный подход к разработке стратегии принятия управленческих решений.

Товары, которые предлагает организация на рынок сбыта должны быть конкурентоспособными, то есть они не должны проигрывать конкурентам по качеству. А чтобы достичь эту самую конкурентоспособность, должны быть разработаны определенные нормативы. Разработка нормативов является для стратегического менеджмента главной целью. Многие ученые разрабатывали концепции и идеи в принятии управленческих решений. Эти идеи применялись и на практике.

Каждое предприятие имеет свою цель, которая в конечном итоге должна привести к отдаче, то есть затраты должны быть оправданы. Метод решающих деревьев помогает найти все возможные варианты, которые позволят обеспечить максимальную отдачу.

Метод дерева решений может использоваться и в личных целях. Увидев наглядно на схеме свои возможности, человек может построить какие-то планы, даже увидеть личные цели в группе. Цели у человека могут быть как личными, так и профессиональными.

Метод решающих деревьев разрабатывается для относительно постоянной структуры, которая мало изменялась в течение какого-то времени. Разрабатывается не только устойчивая структура целей, но и выявляются очевидные проблемы, выбираются определенные направления [1].

Что же такое дерево решений?

Дерево решений – математическая модель, с помощью которой задаётся процесс принятия решений. При этом отображается каждое возможное решение, а также предшествующие и последующие этим решениям события или другие решения и последствия каждого конечного решения [3].

На каждое конкретное решение влияют какие-то определённые факторы. У каждого решения есть свои последствия, которые носят вероятностный характер. Метод дерева решений применяется при прогнозировании, в задачах классификации. Решения приходится принимать в условиях риска, неопределённости. Исход событий зависит от вероятностей. В таких условиях процесс принятия решений является последовательным, а метод дерева решений предполагает определять, что конкретно необходимо делать в каждой вершине дерева [3].

Дерево решений состоит из следующих элементов: дуг, узлов решений, узлов событий и конечных узлов (исходов) [3].

## **Сущность метода дерево решений**

Дерево решений – это схематическое представление проблемы принятия решений. Дерево решений предоставляет возможность руководителю учесть различные варианты действий, правильно проанализировать финансовые результаты, пересмотреть их в соответствии с приписанной им вероятностью, и на основании этого сравнить альтернативы. Неотъемлемой частью метода дерева решений является концепция ожидаемого значения.

Часто результаты одного решения влияют на последующие решения. В этом случае дерево решений выступает полезным инструментом для принятия последовательных ситуаций. Таким образом, с помощью дерева решений можно разрулить сложные ситуации [4].

Дерево решений также может называться регрессионным деревом или деревом классификации. Дерево включает в себя листья и ветки. На ветках дерева пишутся атрибуты. Целевая функция как раз напрямую зависит от этих атрибутов. Значения [целевой функции](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) записаны в листьях. В остальных узлах записываются атрибуты, по которым различаются случаи. Спускаясь по дереву до листа с соответствующим значением, можно классифицировать новый случай. Деревья решений служат для принятия решений при интеллектуальном анализе, в статистике, когда требуется анализ данных, в обучении в различных направлениях, в управленческих решениях предприятия, а также в личных целях. На входе пишутся несколько переменных для создания целевой переменной [5].

В теоретических разработках единого четкого определения понятия дерева решений не существует. Развитие теории управленческих решений, а также методов и технологий разработки стратегии управления позволили появиться на свет нескольким определениям метода решающих деревьев.

В основе дерева лежит иерархическая структура. В вершине дерева стоит цель высшего порядка, или высшего уровня. Главная цель разделяется на подцели. Дерево строится поэтапно, сверху вниз, от более высокого уровня к более низкому. Все цели должны быть согласованы между собой. Конкретизация более низкого уровня должна быть выше верхнего уровня. На самом низком уровне цель формулируется более детально [6].

В. Я Горфинкель отмечает, что название «дерево целей» напрямую связано с деревом, так как схема очень напоминает перевернутое дерево. Дерево это представляет «совокупность распределенных по уровням целей» [7]. В качестве примера можно привести такое дерево. Пусть главной целью будет достижение человеком сбалансированного питания, тогда подцель второго уровня будет правильное потребление количества белков, жиров, углеводов, микроэлементов и витаминов. Подцель третьего уровня будет формулироваться как потребление правильного количества хлеба, молока, овощей, фруктов и других продуктов.

По мнению профессора В.Я. Горфинкеля и профессора В. А. Швандера дерево решений – «структурированная, построенная по иерархическому принципу (распределенная по уровням, ранжированная) совокупность целей экономической системы, программы, плана, в которой выделены: генеральная цель ("вершина дерева"); подчиненные ей подцели первого, второго и последующего уровней ("ветви дерева")» [7].

В энциклопедическом словаре под руководством А. М. Прохорова находим такое определение: «Дерево целей и задач – развернутая, распределенная по уровням совокупность целей и задач экономической программы, построенная по логической схеме: цели программы задачи, которые надо решить для достижения этих целей, мероприятия, обеспечивающие решение задач, ресурсы, и т.д.» [8].

Хотя формулировки разные, отражают они одну суть.

В исследуемом методе используются решения, а теория разработки решений лежит в основе любого менеджмента.

Методика решающих деревьев.

Существует технология разработки решений. Прежде всего, собираются всевозможные данные, сведения, информация о данной проблеме, формулируется проблема и ставится задача. Эту работу обычно выполняет менеджер. Методика дерева решений позволяет дать точно сформулированное решение. На основании этого делается правильный выбор. Если в задаче это было оговорено, то выносится решение, в котором прямо сказано, кто и что будет делать, где и когда, с помощью чего процесс должен осуществиться.

Методы разработки принятия управленческих решений:

1) метод постановки задачи обоснования решений в условиях определенности и неопределенности;

2) «метод мозговой атаки»;

3) экспертный метод оценки;

4) метод моделирования с помощью матриц;

5) методы анализа внешней и внутренней среды предприятия;

6) графоаналитический метод Мака;

7) программно-целевой метод решения задач по скалярному критерию.

Иногда руководство предприятия вынуждено привлекать наиболее авторитетных специалистов экономического анализа, это и есть так называемый «метод мозговой атаки».

В теории менеджмента строго прописаны законы и критерии требований логического, системного и других подходов. При построении дерева учитываются эти ограничения и требования. Также для принятия правильного решения необходимо учитывать факторы и показатели, сдерживающие развитие предприятия, а также факторы внешней и внутренней среды. Нельзя также забывать о предпочтениях руководства к риску, другими словами, важно правильно оценить, готово ли руководство к риску.

Никогда нельзя недооценивать конкурентов или других субъектов, со стороны которых возможны неожиданные суждения и реакции. Эти реакции могут сильно помешать выполнению намеченной цели.

Таким образом, принять правильное решение по методике дерева решений может только тогда, когда учитывается как можно больше факторов. А правильное решение – это то решение, которое будет самым выгодным в данной ситуации.

При разработке и принятии управленческих решений сначала чётко формируется задача, цель, чтобы ясно представлять, что надо решить. Затем формируются решения. Их, как правило, несколько альтернативных. Потом после анализа всех альтернативных решений выбирается самое оптимальное, которое лучше всего подходит для данного предприятия для решения поставленной цели.

Для конечного выбора решения нужен экономический анализ. Постановка задачи должна включать выявление проблем и их оценку. Правильная оценка финансового состояния предприятия показывает, успешно или нет идут процессы продаж на коммерческом предприятии. Важно своевременно найти все проблемы и недостатки в работе коммерческого предприятия, чтобы понять к чему могут привести эти проблемы. Тот, кому поручается разрабатывать варианты решений, несёт за это ответственность. Обычно это менеджер или само руководство непосредственно. Иногда нанимают консультанта со стороны. Исходя из реальных условий работы коммерческого предприятия, делают прогнозы возможных вариантов финансовых результатов. Перед этим оцениваются собственные ресурсы, анализируется, может ли предприятие себе позволить заёмные ресурсы, каков фактор риска взятия заёмных ресурсов. И только после этого приступают к разработке модели финансового состояния, учитывая все варианты возможного использования ресурсов.

В методе решающих деревьев для правильного формирования решения проводится анализ абсолютных показателей: чистого дохода или прибыли, рентабельности, ликвидности баланса, платёжеспособности предприятия. Проводится анализ не только финансово состояния предприятия, но и финансовой устойчивости. Для этого необходима экономическая диагностика финансового состояния организации, у которой имеется набор финансовых характеристик. Оценивается эффективность использования как собственного, так и заёмного капитала. Выявляются возможные ограничения: финансовые, ресурсные, материальные, кадровые, правовые.

Прирост прибыли прогнозируется из ожидаемого прироста продаж. Прирост прибыли — это абсолютная величина. Ещё к абсолютным величинам относятся сокращение дебиторской и кредиторской задолженности, экономия затрат.

Нельзя не включить в варианты решения – решение о бездействии. При решении о бездействии проблемная ситуация не ликвидируется сама собой.

Есть ещё относительные величины, такие как рейтинговая оценка решений в баллах. При этом учитывается значимость и вероятность реализации.

Для выбора оптимального решения используют величину показателя, который выбирается для измерения предпочтений в решении. Эта величина – критерий принятия решения. Берётся максимум этой величины, выведенной с учётом значимости выбранного решения и вероятности его реализации на данном предприятии.

Очень важно для эффективности принятия решения: «применение к системе менеджмента научных подходов и принципов, методов моделирования, автоматизация управления, мотивация качественного решения и другие». Это основные факторы, которые учитываются при принятии управленческого решения.

Интуиция, суждение, рациональность – это три момента, которые присутствуют при принятии любого управленческого решения.

Применяется системный подход к принятию решения. На входе в систему устанавливаются параметры проблемы. Это те проблемы, которые надо решить. Это могут быть, например, объёмы продаж, цены, требования покупателей. На выходе системы формируется решение, которое имеет определённую степень риска, определённую степень вероятности реализации. Результаты могут быть выражены количественно или качественно. На качество управленческого решения влияет внешняя среда (макро и микросреда). Информация, которая поступает от потребителя к тому, кто принял данное решение или кто ответственен за данное решение, является обратной связью. Обратная связь может нести негативную информацию, исходящую от потребителей. Негативная информация может быть из-за некачественного решения ответственного лица. Могут появиться какие-то новые требования потребителя, которые требуют или доработки или даже принятия нового решения.

Для «обеспечения высокого качества эффективности управленческого решения» [8] необходимо чтобы у лица, ответственного за принятие решения было как можно больше достоверной информации по параметрам, входа, выхода, внешней среды. К разработке принятия решения необходим научный подход по изучению экономических законов. Важно применять методы функционально-системного анализа прогнозирования и моделирования. Каждое решение должно быть экономически обосновано.

Структурированная система построения дерева решений позволяет сопоставить варианты решений, выбрать из множества вариантов решений. Важна правовая обоснованность в принятии решения, автоматизация процесса разработки и реализации решений, процесса сбора и обработки информации. Должна быть разработана функциональная система ответственности для мотивации принятия качественного решения. Решение должно быть эффективным и должен присутствовать механизм для реализации решения.

У организации может быть не одна цель, а несколько. Цели могут иметь индивидуальный характер и сложную взаимосвязь. Тогда применяется более сложная специальная модель. Она называется модель дерева целей. Такая специальная модель учитывает иерархию целей. Между целями могут быть конфликтные связи, иерархические и взаимосвязи.

## **Область применения решающих деревьев**

В настоящее время область применения решающих деревьев достаточно широка. Все решаемые задачи могут быть объединены в следующие три класса:

1) описание данных;

2) классификация;

3) регрессия.

Описание данных подразумевает информацию о данных может быть записана в неудобной для анализа форме. В дереве же решений можно хранить эту же самую информацию в компактной форме. При этом точность описания объектов сохраняется.

С задачами классификации, то есть, отнесения объектов к одному из заранее известных классов, деревья решений хорошо справляются.

Деревья решений позволяют установить зависимость целевой переменной от независимых переменных, если целевая переменная имеет непрерывные значения.

При использовании решающих деревьев появляется ряд преимуществ. Так, например, это:

1)быстрый процесс обучения;

2)высокая точность прогноза;

3)понятная классификационная модель;

4)сосредоточенность правил в областях, где специалисту трудно формализовать свои знания;

5)формулировка правил на естественном языке.

Таким образом, для каждого специалиста метод решающих деревьев может быть хорошим помощником при обработке и анализе данных, как на теории, так и на практике.

Деревья решений применимы в различных областях, особенно там, где высока вероятность ошибки правильного принятия решения, и высока цена этой ошибки. Незаменим этот метод при анализе данных. Кроме того, методы построения деревьев решений уже включены в состав некоторых пакетов, предназначенных для интеллектуального анализа данных. Для руководителя или аналитика дерево решений – прекрасный инструмент, хорошее подспорье при анализе данных для принятия выгодного решения.

Итак, деревья решений могут применяться для решения поставленных задач в таких областях:

1) банковское дело;

2) медицина;

3) молекулярная биология;

4) промышленность;

5) сельское хозяйство;

6) торговля;

7) личные цели.

Банковское дело подразумевает, что прежде, чем выдать клиенту кредит, банк с помощью дерева решений может проанализировать кредитоспособность клиента. Оценка кредитоспособности клиентов является для банков очень важной составляющей.

Схема решающих деревьев помогает провести правильную диагностику различных заболеваний.

С помощью решающих деревьев возможно, например, провести анализ строения аминокислот.

С помощью решающих деревьев возможно выявление различных недостатков и дефектов производства. А также возможно провести испытания без разрушений и риска. Например, проверка качества сварки.

Своевременный правильный выбор того или иного решения помогает правильно вести хозяйство, с минимальными затратами и убытками получать максимальную прибыль.

В данной работе будет рассмотрено применение решающих деревьев в сельском хозяйстве. Метод позволит выбрать оптимальный путь решения ведения хозяйства, принятие важных решений в технологии производства, справиться с поставленной задачей по увеличению чистого дохода.

Для удовлетворения спроса населения, важно знать, какие товары на рынке пользуются наибольшим спросом. Зная спрос, можно планировать выпускать наиболее реализуемые товары, то есть ориентироваться на покупателя.

Иногда очень важно сделать правильный личный выбор. Помочь в этом может опять же схема решающих деревьев.

Необходимо иметь обозримое число вариантов развития проекта, только тогда возможно применить метод решающих деревьев. Аналитику или лицу, ответственному за составление дерева решений необходимо предоставить как можно больше достоверной информации, чтобы он мог проанализировать как можно больше вариантов развития. Отбраковывая неперспективные варианты, он выберет наиболее эффективный вариант.

## **Алгоритм построения решающих деревьев**

Для построения дерева решений важна последовательность сбора данных. Данные могут собираться по следующей схеме:

1)определение состава фаз жизненного цикла проекта;

2)определение продолжительности фаз жизненного цикла проекта;

3)определение ключевых событий, которые могут повлиять на дальнейшее развитие проекта;

4)определение времени наступления ключевых событий;

5)формулировка всех возможных решений, которые могут быть приняты в результате наступления каждого ключевого события;

6)определение вероятности принятия каждого решения;

7)определение стоимости каждого этапа осуществления проекта (стоимости работ между ключевыми событиями) в текущих ценах.

На основании полученных данных строится дерево решений. Структура дерева содержит узлы, представляющие собой ключевые события. Ключевые события – это точки принятия решений. Ветви – работы по реализации проекта.

При построении дерева решений рассчитываются вероятность каждого варианта развития событий.

Метод дерева решений не работает, если отсутствует правильная достоверная информация или предоставляется недостоверная информация.

Дерево решений – это графическое изображение процесса принятия решений, в котором отражены альтернативные решения, альтернативные состояния среды, соответствующие вероятности и выигрыши для любых комбинаций альтернатив и состояний среды.

Рисуют деревья слева направо (допускается вариант сверху вниз).

Места, где принимаются решения, обозначают квадратами (), места появления исходов кругами (), возможные решения пунктирными линиями (), возможные исходы сплошными линиями ().

Выделяют следующие этапы построения дерева решений:

1. первоначально обозначают ключевую проблему; ‘то будет вершина дерева;
2. для каждого момента определяют все возможные варианты дальнейших событий, которые могут оказать влияние на ключевую проблему; это будут исходящие от вершины дуги дерева;
3. обозначают время наступления событий;
4. каждой дуге прописывают денежную и вероятностную характеристики;
5. проводят анализ полученных результатов. [9].

Для каждой дуги дерева могут быть определены числовые характеристики, например величина прибыли по проекту и вероятность ее получения. В этом случае оно помогает учесть все возможные варианты действия и соотнести с ними финансовые результаты. Для формулирования сценариев развития проекта необходимо располагать достоверной информацией с учетом вероятности и времени наступления событий. Затем переходят к сравнению альтернатив.[9].

Для каждой альтернативы рассчитывается ожидаемая стоимостная оценка

(), максимальная из сумм оценок выигрышей, умноженных на вероятность реализации выигрышей, для всех возможных вариантов [10].

Стоимостная оценка рассчитывается в каждом узле по формуле 1.4.1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| , | | | | (1.4.1) |
| где |  | – | стоимостная оценка узла, | |
|  |  | – | высокий спрос, | |
|  |  | – | низкий спрос, | |
|  |  | – | прибыль (убыток) при высоком спросе, | |
|  |  | – | прибыль (убыток) при низком спросе, | |

Дерево решений – графический метод, позволяющий увязать точки принятия решения, возможные стратегии , их последствия  с возможными факторами, условиями внешней среды.

Построение дерева решений начинается с более раннего решения. Затем изображаются возможные действия и последствия каждого действия или события. Далее снова принимается решение или выбор направления действия. И так до тех пор, пока все логические последствия результатов не будут исчерпаны.

Дерево решений строится с помощью пяти элементов:

1) момент принятия решения;

2) точка возникновения события;

3) связь между решениями и событиями;

4) вероятность наступления события (сумма вероятностей в каждой точке должна быть равна 1);

5) ожидаемое значение (последствия) – количественное выражение каждой альтернативы, расположенное в конце ветви.

Простейшее решение представляет собой выбор из двух вариантов: "да" или "нет" [11].

## **Преимущества и недостатки решающих деревьев.**

Из преимуществ можно выделить:

1)простота интерпретации и наглядность;

2)возможность работы как с категориями, так и с количественными значениями;

3)универсальность в плане решения задач и классификации, и регрессии;

4)возможность работы с пропусками в данных (пустыми значениями атрибутов). В этом случае деревья решений могут заполнить пропуски наиболее вероятным значением;

5)эффективность при небольших наборах данных.

6)явное описание их конструкции

Из недостатков можно выделить:

1)сложность контроля размера дерева; деревья получаются или очень маленькими или очень большими;

2)нестабильность процесса; небольшие изменения в узле на верхнем уровне ведут к изменениям во всем дереве ниже;

3)неадекватность разделения на классы в сложных случаях [12];

4)решающие деревья представляют собой белый ящик, так как после того, как мы построим решающее дерево, наглядно видно полное описание работы решающего дерева; поэтому решающие деревья – это способ решения задачи классификации и регрессии; этот способ хотя и оригинальный, но эффективный.

Двоичное дерево является ориентированным графом [13]. У каждого графа из каждой вершины выходит не более двух дуг. Соответственно, в каждую вершину входит не более одной дуги. Помимо всего этого, существует одна единственная вершина, в которую не входит ни одна дуга. Эта вершина является корнем дерева. Поэтому существует единственный путь из корня дерева до любой вершины дерева. Таким образом, из каждой вершины исходит либо две дуги, либо не исходит ни одной. Если из вершины не исходит ни одной дуги, то такие вершины называют листьями.

Существует такая важная для дерева характеристика, как глубина или высота. Глубина или высота дерева – это максимальная длина пути в этом дереве.

Решающие деревья применяют для представления упорядоченных наборов конечной длины (формула 1.5.1):

|  |  |
| --- | --- |
|  | (1.5.1) |

Если рассматривать подмассивы массива , то удобно сохранять нумерацию массива  (формула 1.5.2):

|  |  |
| --- | --- |
|  | (1.5.2) |

Существует такое определение как предикат для определения понятия решающего дерева. Если  – это некоторой массив, на котором определена функция (формула 1.5.3):

|  |  |
| --- | --- |
| , | (1.5.3) |

то эта функция  называется предикатом. Смысл предиката состоит в том, что для каждого элемента массива  предикат имеет значение Истина или Ложь. В этом и состоит смысл предиката. Можно вместо  использовать значение , а вместо  — значение . При фиксированном значении  предикат  представляет собой высказывание.

Приведем пример. Массив  состоит из натуральных чисел. Значение функции  равно  для нечетных чисел и  для четных чисел. Тогда Функция  является предикатом, который отвечает на вопрос: «Является ли число  нечетным?».

Решающее дерево применяется для решения задачи описания (представления) массива , на котором можно задать множество предикатов. При этом каждый предикат  делит массив  на два подмассива  и . Это происходит следующим образом (формулы 1.5.4 и 1.5.5):

|  |  |
| --- | --- |
| , | (1.5.4)  (1.5.5) |

В вершинах, которые не являются листьями решающего дерева, располагаются предикаты, а на дугах подмассивы массива .

Предикат в корневой вершине дерева применяется ко всему массиву . Левая дуга, которая выходит из каждой вершины, помечается массивом , а правая массивом , на которые делит предикат, в этой вершине входящий массив. На листьях мы имеем подмассив массива .

Если массив  конечный, то решающее дерево строится таким образом, что листья этого дерева будут содержать по одному элементу.

Предикаты выбираются таким образом, чтобы глубина этого дерева была минимальной. Для этого используются методы минимизации энтропии или максимизировании информации.

Если мы имеем дело с дискретными данными, тогда каждый элемент множества  имеет частоту встречаемости или вероятность в массиве . Если каждый элемент  имеет вероятность , то информационная энтропия Шеннона определяется по формуле 1.5.6:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| , | | | | (1.5.6) |
| где |  | – | двоичный алгоритм | |

Будем считать, что в наших вычислениях . Поскольку , то величина .

Для случая, когда элементы  являются действительными числами, то в качестве меры энтропии можно выбрать выборочную дисперсию (формула 1.5.7):

|  |  |
| --- | --- |
| , | (1.5.7) |

где (формула 1.5.8)

|  |  |
| --- | --- |
|  | (1.5.8) |

Вернемся к построению решающего дерева. Пусть у нас есть массив , которое необходимо разделить на два подмассива с помощью одного из предиката . Обозначим эти подмассивы  и . Пусть множество всех доступных предикатов обозначено через . Тогда  будет наиболее информативным, если (формула 1.5.9):

|  |  |
| --- | --- |
|  | (1.5.9) |

Это означает, что наиболее информативный предикат – это такой предикат, который минимизирует сумму энтропии двух подмассивов. Далее, эту операцию повторять до тех пор, пока в энтропия будет положительной. В случае, когда массив  такой, что , то это означает, что массив  либо пуст, либо состоит из одинаковых элементов.

Таким образом, мы можем получить более компактное описание всего массива.

# **Специальный раздел**

## **Постановка задачи**

Имеется молодой агрономический питомник по производству саженцев плодовых и ягодных культур площадью  га.

У питомника есть еще неосвоенная площадь  га.

Питомник имеет прибыль  млн при спросе на продукцию  и  млн. при спросе .

У питомника есть два варианта повышения прибыли: за счет увеличения земли и за счет внедрения в производство интенсивных технологий.

Если увеличить количество эксплуатируемой земли в  раза и работать по старой технологии, то выход продукции возрастет в  раза, а соответственно и прибыль в  раза.

Если выбрать второй вариант, то требуются разовые вложения  млн. на новую технику (срок эксплуатации техники  лет), но и прибыль тогда возрастет до  млн. в первый год при спросе  и до  млн. в последующие  года.

Спрос на саженцы во второй год будет, как и в первом, а вот в третий и четвёртый годы есть большая вероятность, что он упадет до .

В пятый год спрос вернется к своим прежним цифрам. На увеличенной территории можно также работать по интенсивной технологии, предварительно вложившись в такую же сумму, как и на прежней территории.

Прибыль тогда в первый год будет  млн. при спросе . При спросе  есть риск, что будет убыток  млн.

На второй год прибыль возрастет до  млн. с той же вероятностью, а риск уменьшится до  млн.

В последующие  года вероятностный спрос на продукцию уменьшится до , в пятый год спрос будет таким же, как и в первый год. Показатели прибыли и убытка в последующие три года останутся прежними.

Перед главным агрономом питомника стоит выбор между этими вариантами.

Надо решить задачу методом решающих деревьев. Надо учитывать, что деятельность питомника направлена на длительное существование, поэтому нужен наиболее перспективный путь на ближайшие  лет. Также надо выяснить, на какой год окупятся вложения, если примется решение внедрения интенсивных технологий.

## **Выполнение математического расчёта методом дерева решений**

Составим дерево решений для первого года (Рисунок 2.2.1).

Нарисовав дерево решений, определим наиболее эффективную последовательность действий. Опираться будем на ожидаемые доходы.

Приведем расчеты по каждому году отдельно, поскольку каждый год меняется спрос на продукцию, и прибыль отличается в первый и последующие годы. Срок эксплуатации техники пять лет, и в условии задачи надо найти наиболее перспективный путь на ближайшие 5 лет.

**Расчёт за 1 год**

|  |
| --- |
|  |
| Рисунок 2.2.1 – Построение дерева решений для определения стратегии развития питомника для получения оптимальной прибыли за 1 год |

Вычислим ожидаемую стоимостную оценку в узле (формула 2.2.1), учитывая, что прибыль составит  млн. при высоком спросе, но и есть риск потерять  млн. при низком спросе . Поскольку это интенсивные технологии, учтём вложения в производство  млн.

|  |  |
| --- | --- |
|  | (2.2.1) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.2). Вложений здесь не требуется. Доход составит  млн. с вероятностью  и  млн. с вероятностью.

|  |  |
| --- | --- |
|  | (2.2.2) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.3), учитывая, что прибыль составит  млн. при высоком спросе, но и есть риск потерять  млн. при низком спросе . Поскольку это интенсивные технологии требуются затраты в  млн.

|  |  |
| --- | --- |
|  | (2.2.3) |

Вычислим ожидаемую стоимостную оценку в узле (формула 2.2.4). Это тот вариант, если мы не будем применять интенсивные технологии, и не будем увеличивать площадь, то есть будем работать по старой схеме. Прибыль тогда у нас будет по условию задачи  млн. при спросе  и  млн. при спросе .

|  |  |
| --- | --- |
|  | (2.2.4) |

Найдем оценочную стоимость в узле  (формула 2.2.5). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.5) |

Поэтому в узле  отбрасываем возможное решение «интенсивные технологии».

Найдем оценочную стоимость в узле  (формула 2.2.6). Она равна максимальной оценочной стоимости узлов  и 

|  |  |
| --- | --- |
|  | (2.2.6) |

Поэтому в узле  отбрасываем возможное решение «использовать интенсивные технологии».

Мы подошли к узлу , теперь нам предстоит принять самое главное решение: увеличивать площадь эксплуатации или нет. Для этого посчитаем стоимостную оценку узла  (формула 2.2.7). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.7) |

Поэтому в узле  отбрасываем возможное решение «не увеличивать площадь».

Итак, по результатам первого года у нас получилось, что целесообразно площадь увеличить и не использовать интенсивные технологии.

Это если мы хотим получить очень быструю отдачу. Но как правило в сельском хозяйстве нельзя ориентироваться на показатели первого года. Максимальная отдача, которую можно получить, наступает не ранее четвёртого или пятого года. Поэтому, если мы хотим получить максимальную отдачу, а не быструю, надо проанализировать работу питомника в остальные годы, тем более что этого требует условие задачи.

**Расчёт за 2 год**

Построим дерево решений для второго года (Рисунок 2.2.2):

|  |
| --- |
|  |
| Рисунок 2.2.2 – Построение дерева решений для определения стратегии развития питомника для получения оптимальной прибыли за 2 год |
|  |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.8), учитывая, что прибыль составит уже  млн. при высоком спросе , но и есть риск потерять  млн. при низком спросе . Затрат на вложение уже не будет, так как техника покупалась на пять лет.

|  |  |
| --- | --- |
|  | (2.2.8) |

Для узла  ничего не поменялось (формула 2.2.9):

|  |  |
| --- | --- |
|  | (2.2.9) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.10), учитывая, что прибыль составит теперь  млн. при высоком спросе, но и есть риск потерять  млн. при низком спросе . Затрат на технику не будет.

|  |  |
| --- | --- |
|  | (2.2.10) |

Для узла  ничего не поменялось (формула 2.2.11):

|  |  |
| --- | --- |
|  | (2.2.11) |

Найдем оценочную стоимость в узле  (формула 2.2.12). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.12) |

Поэтому в узле  отбрасываем возможное решение «не использовать интенсивные технологии».

Найдем оценочную стоимость в узле  (формула 2.2.13). Она равна максимальной оценочной стоимости узлов C и D.

|  |  |
| --- | --- |
|  | (2.2.13) |

Поэтому в узле  отбр2асываем возможное решение «не использовать интенсивные технологии».

Мы подошли к узлу , теперь нам предстоит решить: увеличивать площадь эксплуатации или нет. Для этого посчитаем стоимостную оценку узла  (формула 2.2.14). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.14) |

Поэтому в узле  отбрасываем возможное решение «не увеличивать площадь».

Итак, по результатам второго года у нас получилось, что целесообразно площадь увеличить и использовать интенсивные технологии.

Как мы видим, на второй год решение поменялось в сторону использования интенсивных технологий. Кроме того, мы видим, что затраты в

 млн. на технику, приобретённой в первый год, окупятся уже на второй год, так как прибыль составит более  млн., если не сработает фактор риска.

**Расчёт за 3 год**

Построим дерево решений для третьего года (Рисунок 2.2.3):

|  |
| --- |
|  |
| Рисунок 2.2.3 – Построение дерева решений для определения стратегии развития питомника для получения оптимальной прибыли за 3 год |
|  |

Будем делать расчёты, исходя из условия задачи, что спрос на продукцию упадет до , а вероятность дохода и убытков останется прежней.

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.15):

|  |  |
| --- | --- |
|  | (2.2.15) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.16):

|  |  |
| --- | --- |
|  | (2.2.16) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.17):

|  |  |
| --- | --- |
|  | (2.2.17) |

Вычислим ожидаемую стоимостную оценку в узле  (формула 2.2.18):

|  |  |
| --- | --- |
|  | (2.2.18) |

Найдем оценочную стоимость в узле  (формула 2.2.19). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.19) |

Поэтому в узле  отбрасываем возможное решение «не использовать интенсивные технологии».

Найдем оценочную стоимость в узле  (формула 2.2.20). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.20) |

Поэтому в узле  отбрасываем возможное решение «не использовать интенсивные технологии».

Мы подошли к узлу , теперь нам предстоит решить: увеличивать площадь эксплуатации или нет. Для этого посчитаем стоимостную оценку узла  (формула 2.2.21). Она равна максимальной оценочной стоимости узлов  и .

|  |  |
| --- | --- |
|  | (2.2.21) |

Поэтому в узле  отбрасываем возможное решение «не увеличивать площадь».

Итак, по результатам третьего года у нас получилось, что целесообразно площадь увеличить и использовать интенсивные технологии.

**Расчёт за 4 год**

Построим дерево решений для четвёртого года (Рисунок 2.2.4):

|  |
| --- |
|  |
| Рисунок 2.2.4 – Построение дерева решений для определения стратегии развития питомника для получения оптимальной прибыли за 4 год |
|  |

Результаты будут такими же, как и у третьего года, так как ничего не изменилось, поэтому по результатам четвёртого года целесообразно площадь увеличить и использовать интенсивные технологии.

**Расчёт за 5 год**

Построим дерево решений для пятого года (Рисунок 2.2.5):

|  |
| --- |
|  |
| Рисунок 2.2.5 – Построение дерева решений для определения стратегии развития питомника для получения оптимальной прибыли за 5 год |
|  |

На пятый год спрос опять увеличится, как и во втором году, поэтому результаты будут такими же, как и во второй год. Значит, по результатам пятого года целесообразно площадь увеличить и использовать интенсивные технологии.

Таким образом, проанализировав оценочную стоимость за все пять лет, можно принять правильное решение, дающее максимальную отдачу: увеличить площадь эксплуатации земли и при этом использовать интенсивные технологии.

Максимальная отдача будет наблюдаться во второй и пятый годы.

# **Технологический раздел**

## **Описание программной реализации**

В данном разделе описана работа программы, код которой приведён в Приложении А. Программа написана в среде разработки QT на языке С++. Выполнена в консольном режиме.

Программа реализует метод решающих деревьев. Точнее говоря, в автоматическом режиме решает задачу предыдущего раздела о прибыли питомника по входным данным.

На вход программа получает расход на интенсивные технологии, которые учитываются только в 1 год, прогнозируемые вероятности и прибыли за 5 лет. Для удобства значения переменных были переименованы (Рисунки 3.1.1, а и 3.1.1, б).

|  |
| --- |
|  |
| Рисунок 3.1.1, а – Переменные решающего дерева (a – оригинальные названия переменных) |

|  |
| --- |
|  |
| Рисунок 3.1.1, б – Переменные решающего дерева (б – названия переменных в программе) |

Ввод данных выглядит следующим образом. Это цикл рассчитанный на 5 итераций (от 0 до 4). Итерации символизируют года (Листинг 3.1.1).

cout<<TA04.toLocal8Bit().data(); //TA04

cin>>T; //T

for (int i=0; i<=4; i++)

{

int j=i; //j

cout<<endl;

cout<<"-------------------------------------"<<endl;

//---

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().

data()<<TA13.toLocal8Bit().data(); //TA05, TA09, TA13 - j

cin>>A[j]; //A0, A1, A2, A3, A4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().

data()<<TA21.toLocal8Bit().data(); //TA06, TA09, TA21 - j

cin>>P0[j]; //P01, P02, P03, P04, P05

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().

data()<<TA14.toLocal8Bit().data(); //TA07, TA09, TA14 - j

cin>>B[j]; //B0, B1, B2, B3, B4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().

data()<<TA22.toLocal8Bit().data(); //TA08, TA09, TA22 - j

cin>>P1[j]; //P10, P11, P12, P13, P14

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().

data()<<TA15.toLocal8Bit().data(); //TA05, TA10, TA15 - j

cin>>C[j]; //C0, C1, C2, C3, C4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().

data()<<TA23.toLocal8Bit().data(); //TA06, TA10, TA23 - j

cin>>P2[j]; //P20, P21, P22, P23, P24

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().

data()<<TA16.toLocal8Bit().data(); //TA07, TA10, TA16 - j

cin>>D[j]; //D0, D1, D2, D3, D4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().

data()<<TA24.toLocal8Bit().data(); //TA08, TA10, TA24 - j

cin>>P3[j]; //P30, P31, P32, P33, P34

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().

data()<<TA17.toLocal8Bit().data(); //TA05, TA11, TA17 - j

cin>>E[j]; //E0, E1, E2, E3, E4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().

data()<<TA25.toLocal8Bit().data(); //TA06, TA11, TA25 - j

cin>>P4[j]; //P40, P42, P42, P43, P44

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().

data()<<TA18.toLocal8Bit().data(); //TA07, TA11, TA18 - j

cin>>F[j]; //F0, F1, F2, F3, F4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().

data()<<TA26.toLocal8Bit().data(); //TA08, TA11, TA26 - j

cin>>P5[j]; //P50, P51, P52, P53, P54

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().

data()<<TA19.toLocal8Bit().data(); //TA05, TA12, TA19 - j

cin>>G[j]; //G0, G1, G2, G3, G4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().

data()<<TA27.toLocal8Bit().data(); //TA06, TA12, TA27 - j

cin>>P6[j]; //P60, P61, P62, P63, P64

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().

data()<<TA20.toLocal8Bit().data(); //TA07, TA12, TA20 - j

cin>>H[j]; //H0, H1, H2, H3, H4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().

data()<<TA28.toLocal8Bit().data(); //TA08, TA12, TA28 - j

cin>>P7[j]; //P70, P71, P72, P73, P74

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

system("pause");

}

Листинг 3.1.1 – Цикл ввода данных

Входные данные отправляются во второй цикл – цикл решения (Листинг 3.1.2). Он реализует алгоритм решения, описанный в предыдущем разделе. Он так же состоит из 5 итераций (от 0 до 4), которые символизизируют года. Условно он делится на 2 части – решение и ответ. Решение описывает сам алгорим. Оно кодирует данные набором технических переменных.

for (int k=0; k<5; k++)

{

//---

j=k; //j

//---

if (P0[j]>=P1[j]) {PT0[j]=1;} //P0-PT0=1 (A-P1)

else if (P1[j]> P0[j]) {PT0[j]=2;} //P1-PT0=2 (B-P2)

//---

if (P2[j]>=P3[j]) {PT1[j]=1;} //P2-PT1=1 (C-P3)

else if (P3[j]> P2[j]) {PT1[j]=2;} //P3-PT1=2 (D-P4)

//---

if (P4[j]>=P5[j]) {PT2[j]=1;} //P4-PT2=1 (E-P5)

else if (P5[j]> P4[j]) {PT2[j]=2;} //P5-PT2=2 (F-P6)

//---

if (P6[j]>=P7[j]) {PT3[j]=1;} //P6-PT3=1 (G-P7)

else if (P7[j]> P6[j]) {PT3[j]=2;} //P7-PT3=2 (Н-P8)

//---

if (j==0) {T0[j]=A[j]\*P0[j]+B[j]\*P1[j]-T;} //T0-1 (A-B)

else {T0[j]=A[j]\*P0[j]+B[j]\*P1[j];} //T0-2\_5 (A-B)

T1[j]=C[j]\*P2[j]+D[j]\*P3[j]; //T1-1\_5 (C-D)

if (j==0) {T2[j]=E[j]\*P4[j]+F[j]\*P5[j]-T;} //T2-1 (E-F)

else {T2[j]=E[j]\*P4[j]+F[j]\*P5[j];} //T2-2\_5 (E-F)

T3[j]=G[j]\*P6[j]+H[j]\*P7[j]; //T3-1\_5 (G-H)

//---

if (T0[j]>=T1[j]) {S0[j]=T0[j]; ST0[j]=1;} //S0,T0-ST0=1 (A-B)

else if (T1[j]> T0[j]) {S0[j]=T1[j]; ST0[j]=2;} //S0,T1-ST0=2 (C-D)

//---

if (T2[j]>=T3[j]) {S1[j]=T2[j]; ST1[j]=1;} //S1,T2-ST1=1 (E-F)

else if (T3[j]> T2[j]) {S1[j]=T3[j]; ST1[j]=2;} //S1,T3-ST1=2 (G-H)

//---

if (S0[j]>=S1[j]) {PT2[j]=0; PT3[j]=0; ST1[j]=0; RT[j]=1; R[j]=S0[j];}

//R,RT=1 (A-D)

else if (S1[j]> S0[j]) {PT0[j]=0; PT1[j]=0; ST0[j]=0; RT[j]=2; R[j]=S1[j];}

//R,RT=2 (E-H)

//---

if (ST0[j]==1) {PT1[j]=0;} //PT0=1-2,PT1=0 (A-P1,B-P2)

else if (ST0[j]==2) {PT0[j]=0;} //PT1=1-2,PT0=0 (C-P3,D-P4)

//---

if (ST1[j]==1) {PT3[j]=0;} //PT2=1-2,PT3=0 (E-P5,F-P6)

else if (ST1[j]==2) {PT2[j]=0;} //PT3=1-2,PT2=0 (G-P7,H-P8)

Листинг 3.1.2 – Цикл Решения. Часть 1. Решение

Ответ также состоит из двух частей – Общей и Финальной. Общая часть выводит на экран значения оценок дерева решений (Листинг 3.1.3).

cout<<endl;

cout<<TA29.toLocal8Bit().data()<<j+1<<TA30.toLocal8Bit().data(); //TA29, TA30

cout<<"-----------------"<<endl;

cout<<endl;

cout<<TA31.toLocal8Bit().data()<<TA32.toLocal8Bit().

data()<<TA36.toLocal8Bit().data()<<T0[j]<<endl; //TA31,TA32,TA36 - T0

cout<<TA31.toLocal8Bit().data()<<TA33.toLocal8Bit().

data()<<TA37.toLocal8Bit().data()<<T1[j]<<endl; //TA31,TA32,TA37 - T1

cout<<TA31.toLocal8Bit().data()<<TA34.toLocal8Bit().

data()<<TA38.toLocal8Bit().data()<<T2[j]<<endl; //TA31,TA32,TA38 - T2

cout<<TA31.toLocal8Bit().data()<<TA35.toLocal8Bit().

data()<<TA39.toLocal8Bit().data()<<T3[j]<<endl; //TA31,TA35,TA39 - T3

cout<<endl;

cout<<TA40.toLocal8Bit().data()<<TA41.toLocal8Bit().

data()<<TA43.toLocal8Bit().data()<<S0[j]<<endl; //TA40,TA41,TA43 - S0

cout<<TA40.toLocal8Bit().data()<<TA42.toLocal8Bit().

data()<<TA44.toLocal8Bit().data()<<S1[j]<<endl; //TA40,TA42,TA44 - S1

cout<<endl;

cout<<TA45.toLocal8Bit().data()<<TA46.toLocal8Bit().

data()<<R[j]<<endl; //TA45,TA46 - R

cout<<endl;

Листинг 3.1.3 – Цикл Решения. Часть 2. Ответ – общая часть

Финальная часть ответа выбирается по значениям технических переменных, которые были присвоены в части решения (Листинг 3.1.4).

if ((RT[j]==1) && (ST0[j]==1) && (ST1[j]==0) && (PT0[j]==1) && (PT1[j]==0)

&& (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA51.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P0[j]<<endl; //TA47,TA51,TA67 - P0

cout<<TA48.toLocal8Bit().data()<<TA52.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P1[j]<<endl; //TA48,TA52,TA67 - P1

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA59.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<A[j]<<endl; //TA49,TA59,TA67 - A

cout<<TA50.toLocal8Bit().data()<<TA60.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<B[j]<<endl; //TA50,TA60,TA67 - B

cout<<endl;

cout<<TA68.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA68,TA72

cout<<endl;

DTRpSpTP1A(); //Ветвь A - Высокий спрос

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==1) && (ST1[j]==0) && (PT0[j]==2) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA51.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P0[j]<<endl; //TA47,TA51,TA67 - P0

cout<<TA48.toLocal8Bit().data()<<TA52.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P1[j]<<endl; //TA48,TA52,TA67 - P1

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA59.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<A[j]<<endl; //TA49,TA59,TA67 - A

cout<<TA50.toLocal8Bit().data()<<TA60.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<B[j]<<endl; //TA50,TA60,TA67 - B

cout<<endl;

cout<<TA68.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA68,TA73

cout<<endl;

DTRpSpTP2B(); //Ветвь B - Низкий спрос

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==2) && (ST1[j]==0) && (PT0[j]==0) &&

(PT1[j]==1) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA53.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P2[j]<<endl; //TA47,TA53,TA67 - P2

cout<<TA48.toLocal8Bit().data()<<TA54.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P3[j]<<endl; //TA48,TA54,TA67 - P3

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA61.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<C[j]<<endl; //TA49,TA61,TA67 - C

cout<<TA50.toLocal8Bit().data()<<TA62.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<D[j]<<endl; //TA50,TA62,TA67 - D

cout<<endl;

cout<<TA69.toLocal8Bit().data()<<TA72.toLocal8Bit().

data()<<endl; //TA69,TA72

cout<<endl;

DTRpSmTP3C(); //Ветвь С - Высокий спрос

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==2) && (ST1[j]==0) && (PT0[j]==0) &&

(PT1[j]==2) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA53.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P2[j]<<endl; //TA47,TA53,TA67 - P2

cout<<TA48.toLocal8Bit().data()<<TA54.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P3[j]<<endl; //TA48,TA54,TA67 - P3

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA61.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<C[j]<<endl; //TA49,TA61,TA67 - C

cout<<TA50.toLocal8Bit().data()<<TA62.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<D[j]<<endl; //TA50,TA62,TA67 - D

cout<<endl;

cout<<TA69.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA69,TA73

cout<<endl;

DTRpSmTP4D(); //Ветвь D - Низкий спрос

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==1) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==1) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA55.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P4[j]<<endl; //TA47,TA55,TA67 - P4

cout<<TA48.toLocal8Bit().data()<<TA56.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P5[j]<<endl; //TA48,TA56,TA67 - P5

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA63.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<E[j]<<endl; //TA49,TA63,TA67 - E

cout<<TA50.toLocal8Bit().data()<<TA64.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<F[j]<<endl; //TA50,TA64,TA67 - F

cout<<endl;

cout<<TA70.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA70,TA72

cout<<endl;

DTRmSpTP5E(); //Ветвь E - Высокий спрос

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==1) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==2) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA55.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P4[j]<<endl; //TA47,TA55,TA67 - P4

cout<<TA48.toLocal8Bit().data()<<TA56.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P5[j]<<endl; //TA48,TA56,TA67 - P5

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA63.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<E[j]<<endl; //TA49,TA63,TA67 - E

cout<<TA50.toLocal8Bit().data()<<TA64.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<F[j]<<endl; //TA50,TA64,TA67 - F

cout<<endl;

cout<<TA70.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA70,TA73

cout<<endl;

DTRmSpTP6F(); //Ветвь F - Низкий спрос

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==2) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==1))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA57.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P6[j]<<endl; //TA47,TA57,TA67 - P6

cout<<TA48.toLocal8Bit().data()<<TA58.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P7[j]<<endl; //TA48,TA58,TA67 - P7

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA65.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<G[j]<<endl; //TA49,TA65,TA67 - G

cout<<TA50.toLocal8Bit().data()<<TA66.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<H[j]<<endl; //TA49,TA66,TA67 - H

cout<<endl;

cout<<TA71.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA71,TA72

cout<<endl;

DTRmSmTP7G(); //Ветвь G - Высокий спрос

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==2) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==2))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA57.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P6[j]<<endl; //TA47,TA57,TA67 - P6

cout<<TA48.toLocal8Bit().data()<<TA58.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<P7[j]<<endl; //TA48,TA58,TA67 - P7

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA65.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<G[j]<<endl; //TA49,TA65,TA67 - G

cout<<TA50.toLocal8Bit().data()<<TA66.toLocal8Bit().

data()<<TA67.toLocal8Bit().data()<<H[j]<<endl; //TA49,TA66,TA67 - H

cout<<endl;

cout<<TA71.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA71,TA73

cout<<endl;

DTRmSmTP8H(); //Ветвь H - Низкий спрос

cout<<endl;

//---

}

//---

if (j==0) {cout<<TA74.toLocal8Bit().data()<<endl;} //TA74

if (j!=4) {system("pause");}

//---

}

Листинг 3.1.4 – Цикл Решения. Часть 2. Ответ – финальная часть

Разобравшись с устройством кода, мы можем приступить к запуску программы. Сначала нас встретит меню программы (Рисунок 3.1.2). Пункт A – отвечает за решение задачи, код которой мы рассмотрели, пункт B – отвечает за справку, в нём мы увидим условие задачи, пункт X – отвечает за выход из программы. Нас интересует решение задачи, поэтому выберем пункт A.

|  |
| --- |
|  |
| Рисунок 3.1.2 – Меню программы |

Выбрав пункт A, мы будем должны ввести входные данные. Будем использовать значения из предыдущего раздела (Рисунок 3.1.3).

|  |
| --- |
|  |
| Рисунок 3.1.3 – Ввод данных |

Когда мы введём все необходимые данные, программа по алгоритму решающего дерева рассчитает и выдаст оптимальное решение (Рисунок 3.1.4).

|  |
| --- |
|  |
| Рис. 3.1.4 – Вывод ответа. Расчёт за 1 год |
|  |

Как мы видим, программа справилась со своей задачей. Мы получили такой же ответ, как и при ручном решении. Программа правильно определила ветвь. Она полностью готова к практическому применению.

# **Экономический раздел**

## **Оценка экономических перспектив разработанного программного обеспечения**

Целью и задачей данной выпускной квалификационной работы является: составление алгоритма решающих деревьев и его программная реализация для применения на предприятиях, чтобы построить оптимальные управленческие решения.

Предметом исследования данной работы является конкретное сельскохозяйственное предприятие, перед которым стоит выбор наиболее перспективного дальнейшего развития, который в конечном итоге приведёт к максимальной прибыли и быстрому сроку окупаемости.

Опираясь на ожидаемые доходы, построив дерево решений и разработав программное обеспечение, мы определили наиболее эффективную последовательность действий. В ходе работы был проведён экономический анализ всех альтернативных решений.

За основу экономического анализа была взята стоимостная оценка каждого узла дерева решений. Стоимостная оценка () для каждой альтернативы берется максимальная, состоящая из сумм оценок выигрышей, которые необходимо умножить на вероятность реализации выигрышей, для всех возможных вариантов.

Для построения схемы решающих деревьев и составления программного продукта потребуется ряд затрат.

Затраты будут рассчитываться по формуле 4.1.1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| , | | | | (4.1.1) |
| где |  | – | затраты на построение дерева решений и разработку программного обеспечения, | |
|  |  | – | затраты на электроэнергию, | |
|  |  | – | затраты на амортизацию оборудования, | |
|  |  | – | затраты на сотрудника предприятия, который занимается построением дерева решений и разработкой программного обеспечения | |

Мы использовали бесплатные программы, поэтому затраты на используемое для разработки программное обеспечение не учитывается.

Затраты на электроэнергию рассчитываются по формуле 4.1.2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| , | | | | (4.1.2) |
| где |  | – | затраты на электроэнергию, | |
|  |  | – | Общая трудоёмкость, | |
|  |  | – | стоимость электроэнергии (в  году цена за  кВт/ч составляет  руб, | |
|  |  | – | Суммарная мощность оборудования (современного ПК) составляет от  кВт | |
|  |  | – | стоимостная оценка узла, | |

Составим таблицу трудоёмкости (Таблица 4.1.1).

|  |  |  |
| --- | --- | --- |
| Таблица 4.1.1 – Трудоёмкость | | |
| **№ п/п** | **Этап** | **Трудоёмкость, час** |
| 1 | Постановка и описание задачи | 10 |
| 2 | Составление и построение дерева решений | 30 |
| 3 | Написание программы | 24 |
| 4 | Отладка программы | 6 |
| 5 | Оформление документов | 24 |

Общая трудоёмкость составила (формулы 4.1.3 и 4.1.4):

|  |  |
| --- | --- |
| ч | (4.1.3) |
| руб | (4.1.4) |

Затраты на амортизацию оборудования будем рассчитывать по формуле 4.1.5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| , | | | | (4.1.5) |
| где |  | – | балансовая стоимость ПК, | |
|  |  | – | время устаревания оборудования | |

Балансовая стоимость оборудования, на котором написано программное обеспечение для нашей задачи составляет  рублей, время до устаревания оборудования в среднем считается равным  годам ( дней), время разработки составило  часа, следовательно, получим (формула 4.1.6):

|  |  |
| --- | --- |
| руб | (4.1.6) |

Рассчитаем затраты на сотрудника предприятия, который занимается построением дерева решений и разработкой программного обеспечения. Часовая ставка главного агронома составляет  рублей в час, тогда его заработанная плата за проект составит (формула 4.1.7):

|  |  |
| --- | --- |
| руб | (4.1.7) |

Ставка страховых взносов с 1 января 2012 года составляет 30%. Затраты на отчисления страховых взносов составят (формула 4.1.8):

|  |  |
| --- | --- |
| руб | (4.1.8) |

Заработная плата с учетом страховых взносов составит (формула 4.1.9):

|  |  |
| --- | --- |
| руб | (4.1.9) |

Затраты на построение дерева решений и разработку программного обеспечения составят (формула 4.1.10):

|  |  |
| --- | --- |
| руб | (4.1.10) |

Если округлить, то (формула 4.1.11):

|  |  |
| --- | --- |
| руб | (4.1.11) |

Кроме того, метод решающих деревьев наглядно показывает, когда наступит срок окупаемости у конкретного предприятия. В нашем случае срок окупаемости наступит на второй год, если не сработает фактор риска.

# **ЗАКЛЮЧЕНИЕ**

Дерево решений – вполне типичный пример алгоритма так называемого индуктивного обучения «по прецедентам», когда на основе значений атрибутов исходных данных строится некоторая решающая функция, оцениваются параметры модели и т.п. То есть в этом случае прогнозную модель конструирует не пользователь, а алгоритм – автоматически на основе исходных данных. Пользователь же, выбирая метод (в нашем случае дерево решений), определяет некий класс моделей, которые могут быть построены и обучены с помощью данного алгоритма.

В результате проделанной работы, можно сделать вывод, что метод дерева решений имеет огромный потенциал, в том числе в сфере управления предприятиями и экономикой, его удобно применять и довольно легко рассчитывать. Самое главное – дерево решений наглядно показывает все стратегии и помогает сделать правильный выбор. Данный метод определённо стоит внедрять на предприятиях. Пример с агрономическим питомником прекрасно это показывает.

Программное обеспечение было разработано на языке C++. Оно позволяет автоматизировать расчёт выигрышной стратегии и поступить наиболее выгодно. Разработка программы имеет довольно низкие затраты, по сравнению с прибылью которую можно получить.

Исчисление при построении дерева решений и разработке программы идет на тысячи, а прибыль и риск потерь питомника, рассмотренного в работе, исчисляются миллионами, поэтому очень выгодно перед проектированием управленческих задач нанять специалиста, который грамотно разработает дерево управленческих решений. Для российских предпринимателей очень важно, чтобы данный метод был лучше изучен и чаще внедрялся на практике.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Райзберг Б.А., Лозовский Л.Ш., Стародубцева Е.Б. Современный экономический словарь. — 5-е изд., перераб. и доп. — М.: НОРМА, 2006
2. Радугин А. А., Радугин К. А. Введение в менеджмент: социология организаций и управления — учебное пособие. — Воронеж: Высшая школа предпринимателей, 2005
3. Метод дерева решений и другие методы на основе графов [Электронный ресурс] // Function-x, 2019.

URL: <https://function-x.ru/graphs4_modeling_decision_tree_game_tree.html>

(дата посещения: 23.05.2019)

1. Дерево Решений. Сущность метода «дерево решений» [Электронный ресурс] // Studwood, 2019.

URL: <https://studwood.ru/2151494/menedzhment/derevo_resheniy> (дата

посещения: 23.05.2019)

1. Дерево решений [Электронный ресурс] // Wikipedia, 2019.

URL: <https://ru.wikipedia.org/wiki/Дерево_решений> (дата посещения:

23.05.2019)

1. А.Н. Азрилиян. Новый экономический словарь. – 3-е изд. – М.: Институт новой экономики, 2008
2. В.Я. Горфинкель, В.А. Швандер. Экономика предприятия: Учебник для вузов.– 4- е изд., перераб. и доп. – М.: ЮНИТИ – ДАНА, 2007
3. Прохоров А.М. Энциклопедический словарь экономии и права. М.: ИНФРА – М, 2009
4. Метод «дерево решений» [Электронный ресурс] //Studfiles, 2019.

URL: <https://studfiles.net/preview/5663272/page:36/> (дата посещения

23.05.2019)

1. Дерево решений: оценка альтернатив в управлении [Электронный ресурс] // Elitarium, 2019. URL: <http://www.elitarium.ru/derevo-reshenij-ocenka-informacija-alternativa-neopredelennost-variant-dejstvie-vybor/> (дата посещения: 23.05.2019)
2. Дерево решений – Методы принятия управленческих решений [Электронный ресурс] // Studme, 2019.

URL: <https://studme.org/31896/menedzhment/derevo_resheniy> (дата посещения

23.05.2019)

1. Использование деревьев решений в задачах прогнозной аналитики [Электронный ресурс] // Prognoz, 2019.

URL: <http://www.prognoz.ru/blog/platform/decision-tree-in-predictive-analytics/>

(дата посещения: 23.05.2019)

1. Шамин Р. В. Практическое руководство по машинному обучению. — М.: Научный канал Lector.ru, 2019. [Электронный ресурс] // AI.Lector, 2019. URL: <http://shamin.ru/link/ai/shamin.pdf> (дата посещения: 23.05.2019)

# **ПРИЛОЖЕНИЕ А.**

**Листинг программного кода**

Приведена сокращённая версия листинга исходников из-за большого объёма.[[1]](#footnote-1)

#include <QtCore>

#include <iostream>

#include <iomanip>

#include <QCoreApplication>

#ifndef DECISIONTREECONSOLE\_H

#define DECISIONTREECONSOLE\_H

using namespace std;

void caseA(); //Cодержимое пункта A

#endif // DECISIONTREECONSOLE

Листинг A.1 – DTFunc.h. Заголовочный файл функций программы

#include DTFunc.h

void caseA() //Содержимое пункта A

{//A. Ввод данных и расчёт оптимальной ветви

//---

//Шапка

cout<<TA00.toLocal8Bit().data(); //TA00

cout<<"-------------------------------------"<<endl;

cout<<endl;

//---

cout<<TA01.toLocal8Bit().data(); //TA01

cout<<TA02.toLocal8Bit().data(); //TA02

cout<<endl;

cout<<endl;

cout<<TA03.toLocal8Bit().data(); //TA03

cout<<endl;

cout<<TA75.toLocal8Bit().data(); //TA75

cout<<endl;

cout<<TA04.toLocal8Bit().data(); //TA04

cin>>T; //T

for (int i=0; i<=4; i++)

{

int j=i; //j

cout<<endl;

cout<<"-------------------------------------"<<endl;

//---

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().data()

<<TA13.toLocal8Bit().data(); //TA05, TA09, TA13 - j

cin>>A[j]; //A0, A1, A2, A3, A4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().data()

<<TA21.toLocal8Bit().data(); //TA06, TA09, TA21 - j

cin>>P0[j]; //P01, P02, P03, P04, P05

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().data()

<<TA14.toLocal8Bit().data(); //TA07, TA09, TA14 - j

cin>>B[j]; //B0, B1, B2, B3, B4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA09.toLocal8Bit().data()

<<TA22.toLocal8Bit().data(); //TA08, TA09, TA22 - j

cin>>P1[j]; //P10, P11, P12, P13, P14

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().data()

<<TA15.toLocal8Bit().data(); //TA05, TA10, TA15 - j

cin>>C[j]; //C0, C1, C2, C3, C4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().data()

<<TA23.toLocal8Bit().data(); //TA06, TA10, TA23 - j

cin>>P2[j]; //P20, P21, P22, P23, P24

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().data()

<<TA16.toLocal8Bit().data(); //TA07, TA10, TA16 - j

cin>>D[j]; //D0, D1, D2, D3, D4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA10.toLocal8Bit().data()

<<TA24.toLocal8Bit().data(); //TA08, TA10, TA24 - j

cin>>P3[j]; //P30, P31, P32, P33, P34

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().data()

<<TA17.toLocal8Bit().data(); //TA05, TA11, TA17 - j

cin>>E[j]; //E0, E1, E2, E3, E4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().data()

<<TA25.toLocal8Bit().data(); //TA06, TA11, TA25 - j

cin>>P4[j]; //P40, P42, P42, P43, P44

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().data()

<<TA18.toLocal8Bit().data(); //TA07, TA11, TA18 - j

cin>>F[j]; //F0, F1, F2, F3, F4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA11.toLocal8Bit().data()

<<TA26.toLocal8Bit().data(); //TA08, TA11, TA26 - j

cin>>P5[j]; //P50, P51, P52, P53, P54

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA05.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().data()

<<TA19.toLocal8Bit().data(); //TA05, TA12, TA19 - j

cin>>G[j]; //G0, G1, G2, G3, G4

cout<<endl<<endl;

cout<<TA06.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().data()

<<TA27.toLocal8Bit().data(); //TA06, TA12, TA27 - j

cin>>P6[j]; //P60, P61, P62, P63, P64

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

cout<<TA07.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().data()

<<TA20.toLocal8Bit().data(); //TA07, TA12, TA20 - j

cin>>H[j]; //H0, H1, H2, H3, H4

cout<<endl<<endl;

cout<<TA08.toLocal8Bit().data()<<j+1<<TA12.toLocal8Bit().data()

<<TA28.toLocal8Bit().data(); //TA08, TA12, TA28 - j

cin>>P7[j]; //P70, P71, P72, P73, P74

cout<<endl;

//---

cout<<endl;

cout<<"-------------------------------------"<<endl;

cout<<endl;

system("pause");

}

//---

for (int k=0; k<5; k++)

{

//---

j=k; //j

//---

if (P0[j]>=P1[j]) {PT0[j]=1;} //P0-PT0=1 (A-P1)

else if (P1[j]> P0[j]) {PT0[j]=2;} //P1-PT0=2 (B-P2)

//---

if (P2[j]>=P3[j]) {PT1[j]=1;} //P2-PT1=1 (C-P3)

else if (P3[j]> P2[j]) {PT1[j]=2;} //P3-PT1=2 (D-P4)

//---

if (P4[j]>=P5[j]) {PT2[j]=1;} //P4-PT2=1 (E-P5)

else if (P5[j]> P4[j]) {PT2[j]=2;} //P5-PT2=2 (F-P6)

//---

if (P6[j]>=P7[j]) {PT3[j]=1;} //P6-PT3=1 (G-P7)

else if (P7[j]> P6[j]) {PT3[j]=2;} //P7-PT3=2 (Н-P8)

//---

if (j==0) {T0[j]=A[j]\*P0[j]+B[j]\*P1[j]-T;} //T0-1 (A-B)

else {T0[j]=A[j]\*P0[j]+B[j]\*P1[j];} //T0-2\_5 (A-B)

T1[j]=C[j]\*P2[j]+D[j]\*P3[j]; //T1-1\_5 (C-D)

if (j==0) {T2[j]=E[j]\*P4[j]+F[j]\*P5[j]-T;} //T2-1 (E-F)

else {T2[j]=E[j]\*P4[j]+F[j]\*P5[j];} //T2-2\_5 (E-F)

T3[j]=G[j]\*P6[j]+H[j]\*P7[j]; //T3-1\_5 (G-H)

//---

if (T0[j]>=T1[j]) {S0[j]=T0[j]; ST0[j]=1;} //S0,T0-ST0=1 (A-B)

else if (T1[j]> T0[j]) {S0[j]=T1[j]; ST0[j]=2;} //S0,T1-ST0=2 (C-D)

//---

if (T2[j]>=T3[j]) {S1[j]=T2[j]; ST1[j]=1;} //S1,T2-ST1=1 (E-F)

else if (T3[j]> T2[j]) {S1[j]=T3[j]; ST1[j]=2;} //S1,T3-ST1=2 (G-H)

//---

if (S0[j]>=S1[j]) {PT2[j]=0; PT3[j]=0; ST1[j]=0; RT[j]=1; R[j]=S0[j];}

//R,RT=1 (A-D)

else if (S1[j]> S0[j]) {PT0[j]=0; PT1[j]=0; ST0[j]=0; RT[j]=2; R[j]=S1[j];}

//R,RT=2 (E-H)

//---

if (ST0[j]==1) {PT1[j]=0;} //PT0=1-2,PT1=0 (A-P1,B-P2)

else if (ST0[j]==2) {PT0[j]=0;} //PT1=1-2,PT0=0 (C-P3,D-P4)

//---

if (ST1[j]==1) {PT3[j]=0;} //PT2=1-2,PT3=0 (E-P5,F-P6)

else if (ST1[j]==2) {PT2[j]=0;} //PT3=1-2,PT2=0 (G-P7,H-P8)

//---

cout<<endl;

cout<<TA29.toLocal8Bit().data()<<j+1<<TA30.toLocal8Bit().data(); //TA29, TA30

cout<<"-----------------"<<endl;

cout<<endl;

cout<<TA31.toLocal8Bit().data()<<TA32.toLocal8Bit().data()

<<TA36.toLocal8Bit().data()<<T0[j]<<endl; //TA31,TA32,TA36 - T0

cout<<TA31.toLocal8Bit().data()<<TA33.toLocal8Bit().data()

<<TA37.toLocal8Bit().data()<<T1[j]<<endl; //TA31,TA32,TA37 - T1

cout<<TA31.toLocal8Bit().data()<<TA34.toLocal8Bit().data()

<<TA38.toLocal8Bit().data()<<T2[j]<<endl; //TA31,TA32,TA38 - T2

cout<<TA31.toLocal8Bit().data()<<TA35.toLocal8Bit().data()

<<TA39.toLocal8Bit().data()<<T3[j]<<endl; //TA31,TA35,TA39 - T3

cout<<endl;

cout<<TA40.toLocal8Bit().data()<<TA41.toLocal8Bit().data()

<<TA43.toLocal8Bit().data()<<S0[j]<<endl; //TA40,TA41,TA43 - S0

cout<<TA40.toLocal8Bit().data()<<TA42.toLocal8Bit().data()

<<TA44.toLocal8Bit().data()<<S1[j]<<endl; //TA40,TA42,TA44 - S1

cout<<endl;

cout<<TA45.toLocal8Bit().data()<<TA46.toLocal8Bit().data()<<R[j]<<endl;

//TA45,TA46 - R

cout<<endl;

//---

if ((RT[j]==1) && (ST0[j]==1) && (ST1[j]==0) && (PT0[j]==1) && (PT1[j]==0)

&& (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA51.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P0[j]<<endl; //TA47,TA51,TA67 - P0

cout<<TA48.toLocal8Bit().data()<<TA52.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P1[j]<<endl; //TA48,TA52,TA67 - P1

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA59.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<A[j]<<endl; //TA49,TA59,TA67 - A

cout<<TA50.toLocal8Bit().data()<<TA60.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<B[j]<<endl; //TA50,TA60,TA67 - B

cout<<endl;

cout<<TA68.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA68,TA72

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==1) && (ST1[j]==0) && (PT0[j]==2) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA51.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P0[j]<<endl; //TA47,TA51,TA67 - P0

cout<<TA48.toLocal8Bit().data()<<TA52.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P1[j]<<endl; //TA48,TA52,TA67 - P1

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA59.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<A[j]<<endl; //TA49,TA59,TA67 - A

cout<<TA50.toLocal8Bit().data()<<TA60.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<B[j]<<endl; //TA50,TA60,TA67 - B

cout<<endl;

cout<<TA68.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA68,TA73

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==2) && (ST1[j]==0) && (PT0[j]==0) &&

(PT1[j]==1) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA53.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P2[j]<<endl; //TA47,TA53,TA67 - P2

cout<<TA48.toLocal8Bit().data()<<TA54.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P3[j]<<endl; //TA48,TA54,TA67 - P3

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA61.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<C[j]<<endl; //TA49,TA61,TA67 - C

cout<<TA50.toLocal8Bit().data()<<TA62.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<D[j]<<endl; //TA50,TA62,TA67 - D

cout<<endl;

cout<<TA69.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA69,TA72

cout<<endl;

//---

}

else if ((RT[j]==1) && (ST0[j]==2) && (ST1[j]==0) && (PT0[j]==0) &&

(PT1[j]==2) && (PT2[j]==0) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA53.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P2[j]<<endl; //TA47,TA53,TA67 - P2

cout<<TA48.toLocal8Bit().data()<<TA54.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P3[j]<<endl; //TA48,TA54,TA67 - P3

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA61.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<C[j]<<endl; //TA49,TA61,TA67 - C

cout<<TA50.toLocal8Bit().data()<<TA62.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<D[j]<<endl; //TA50,TA62,TA67 - D

cout<<endl;

cout<<TA69.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA69,TA73

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==1) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==1) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA55.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P4[j]<<endl; //TA47,TA55,TA67 - P4

cout<<TA48.toLocal8Bit().data()<<TA56.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P5[j]<<endl; //TA48,TA56,TA67 - P5

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA63.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<E[j]<<endl; //TA49,TA63,TA67 - E

cout<<TA50.toLocal8Bit().data()<<TA64.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<F[j]<<endl; //TA50,TA64,TA67 - F

cout<<endl;

cout<<TA70.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA70,TA72

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==1) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==2) && (PT3[j]==0))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA55.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P4[j]<<endl; //TA47,TA55,TA67 - P4

cout<<TA48.toLocal8Bit().data()<<TA56.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P5[j]<<endl; //TA48,TA56,TA67 - P5

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA63.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<E[j]<<endl; //TA49,TA63,TA67 - E

cout<<TA50.toLocal8Bit().data()<<TA64.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<F[j]<<endl; //TA50,TA64,TA67 - F

cout<<endl;

cout<<TA70.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA70,TA73

cout<<endl;

cout<<endl;

//---

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==2) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==1))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA57.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P6[j]<<endl; //TA47,TA57,TA67 - P6

cout<<TA48.toLocal8Bit().data()<<TA58.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P7[j]<<endl; //TA48,TA58,TA67 - P7

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA65.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<G[j]<<endl; //TA49,TA65,TA67 - G

cout<<TA50.toLocal8Bit().data()<<TA66.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<H[j]<<endl; //TA49,TA66,TA67 - H

cout<<endl;

cout<<TA71.toLocal8Bit().data()<<TA72.toLocal8Bit().data()<<endl; //TA71,TA72

cout<<endl;

}

else if ((RT[j]==2) && (ST0[j]==0) && (ST1[j]==2) && (PT0[j]==0) &&

(PT1[j]==0) && (PT2[j]==0) && (PT3[j]==2))

{

//---

cout<<TA47.toLocal8Bit().data()<<TA57.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P6[j]<<endl; //TA47,TA57,TA67 - P6

cout<<TA48.toLocal8Bit().data()<<TA58.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<P7[j]<<endl; //TA48,TA58,TA67 - P7

cout<<endl;

cout<<TA49.toLocal8Bit().data()<<TA65.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<G[j]<<endl; //TA49,TA65,TA67 - G

cout<<TA50.toLocal8Bit().data()<<TA66.toLocal8Bit().data()

<<TA67.toLocal8Bit().data()<<H[j]<<endl; //TA49,TA66,TA67 - H

cout<<endl;

cout<<TA71.toLocal8Bit().data()<<TA73.toLocal8Bit().data()<<endl; //TA71,TA73

cout<<endl;

cout<<endl;

//---

}

if (j==0) {cout<<TA74.toLocal8Bit().data()<<endl;} //TA74

if (j!=4) {system("pause");}

//---

}

//---

}

Листинг А.2 – DTFunc.cpp. Файл реализации функций программы

#include "DTFunc.h" //подключение заголовочного файла функций

программы и необходимых библиотек

//---

//Главная функция программы

int main(int argc, char \*argv[])

{

QCoreApplication app(argc, argv);

caseA(); //Содержимое пункта A

return app.exec();

}

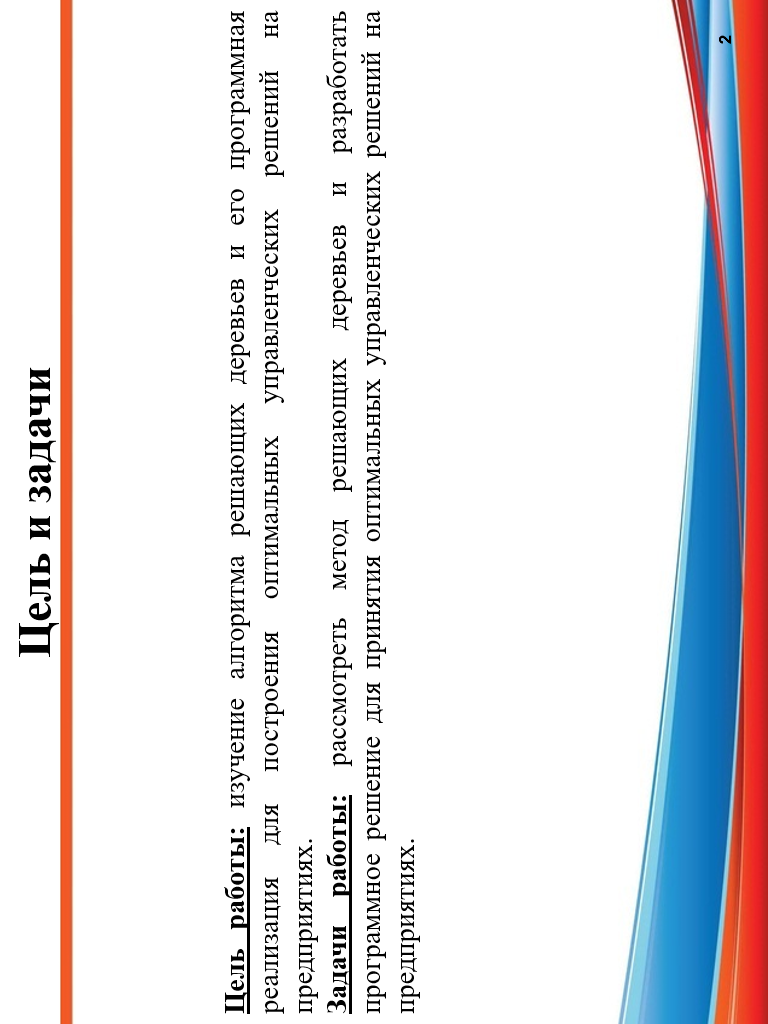
//---

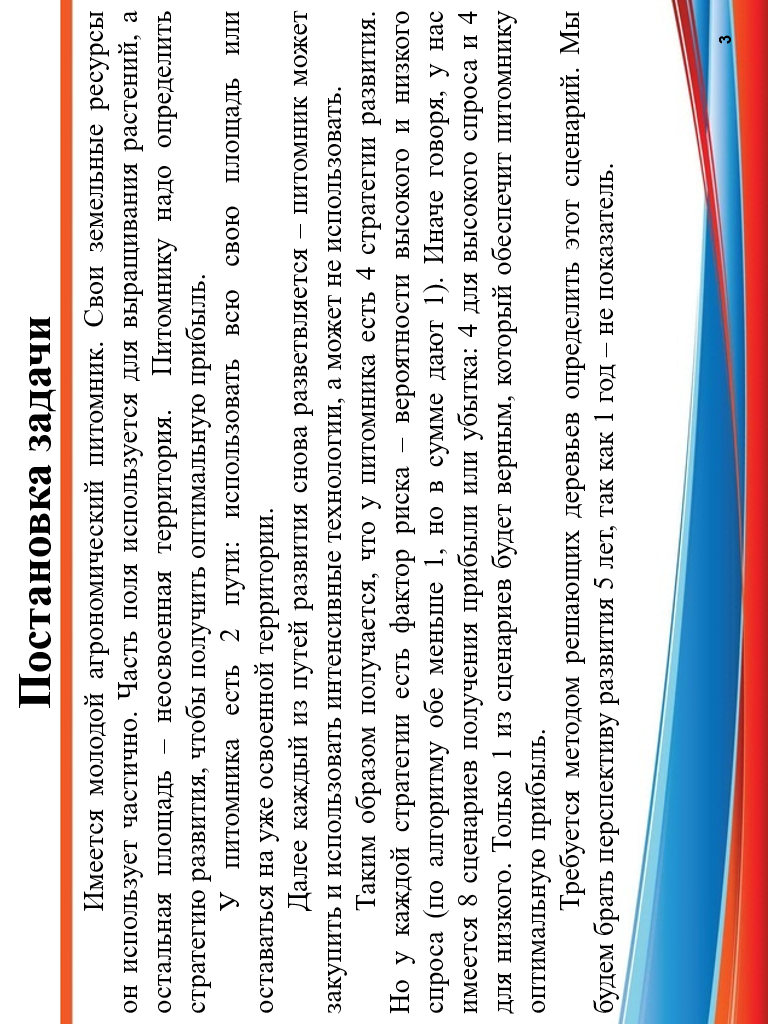
Листинг А.3 – main.cpp. Главный файл программы

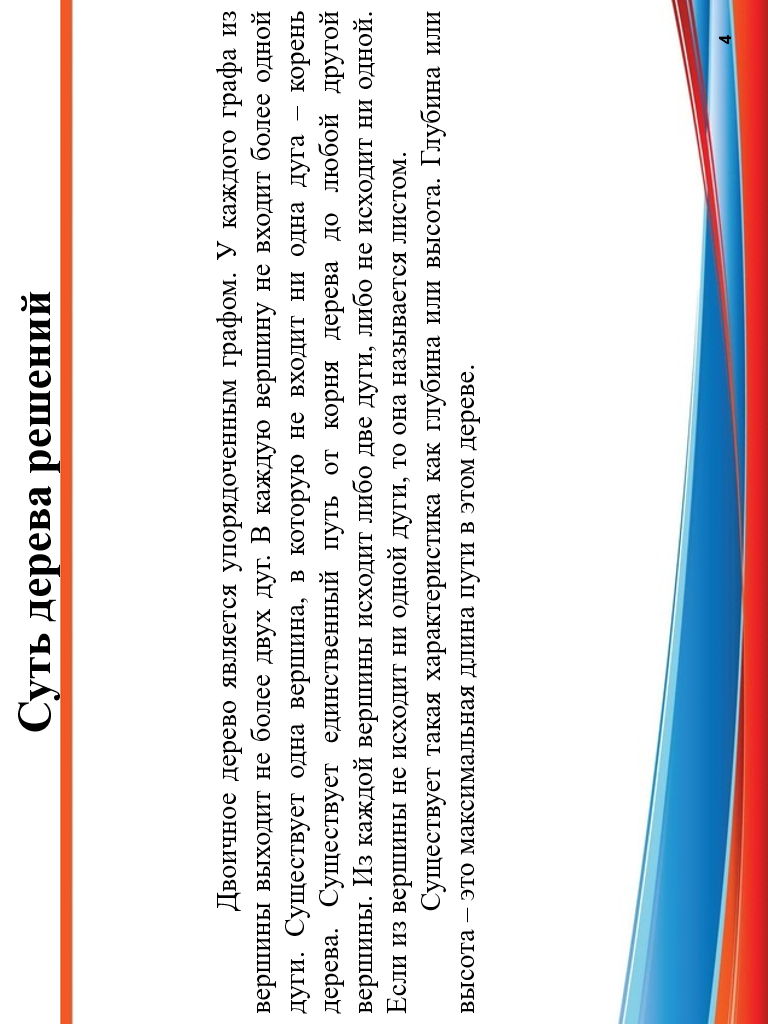
# **ПРИЛОЖЕНИЕ Б.**

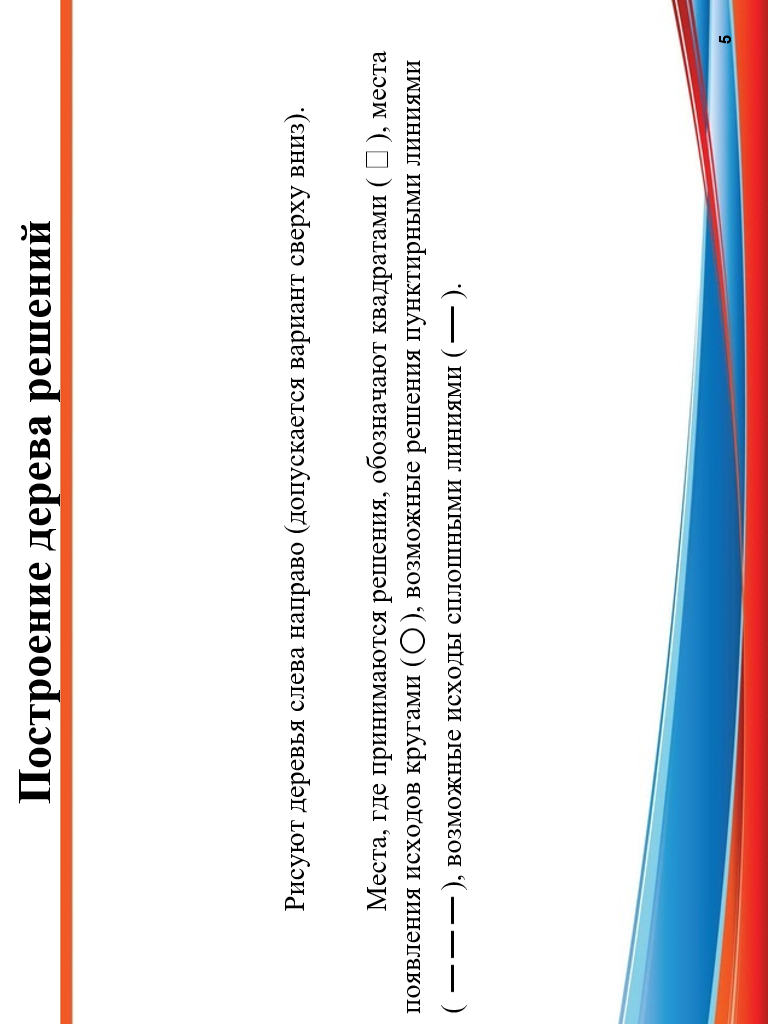
**Презентация**

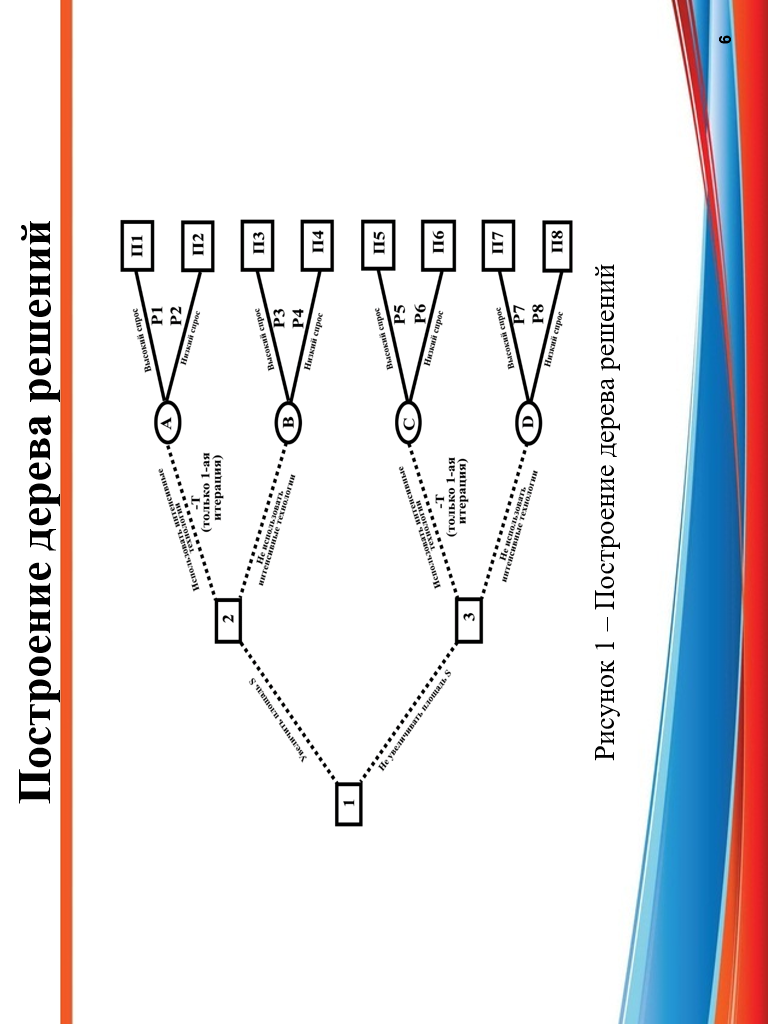


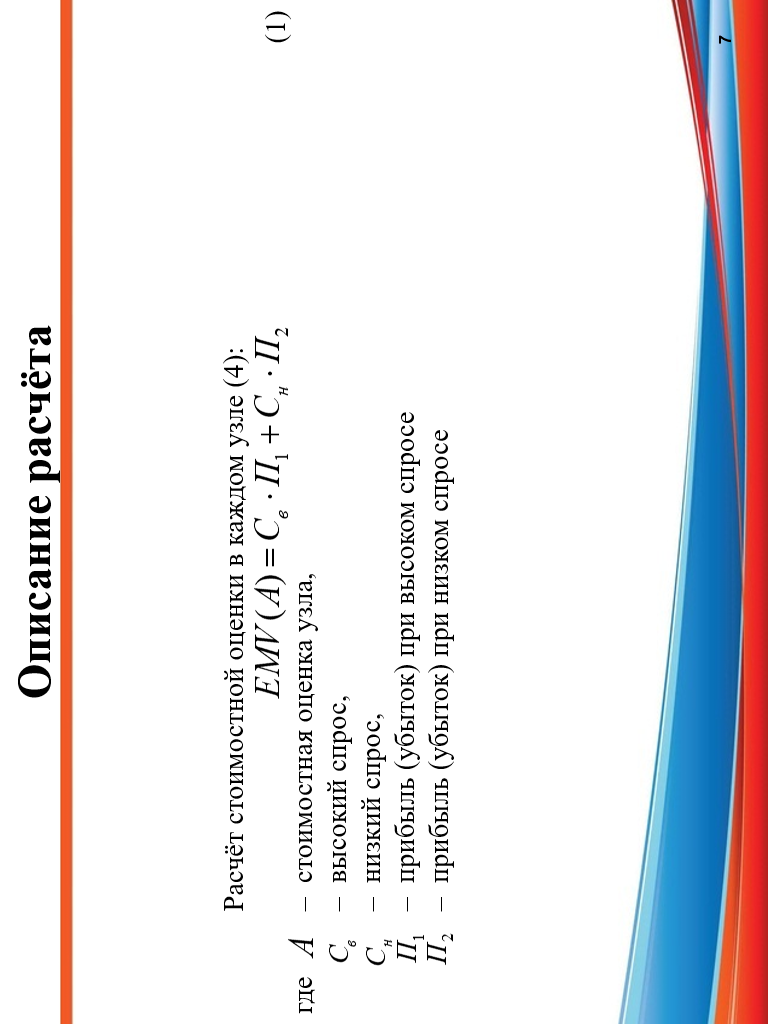




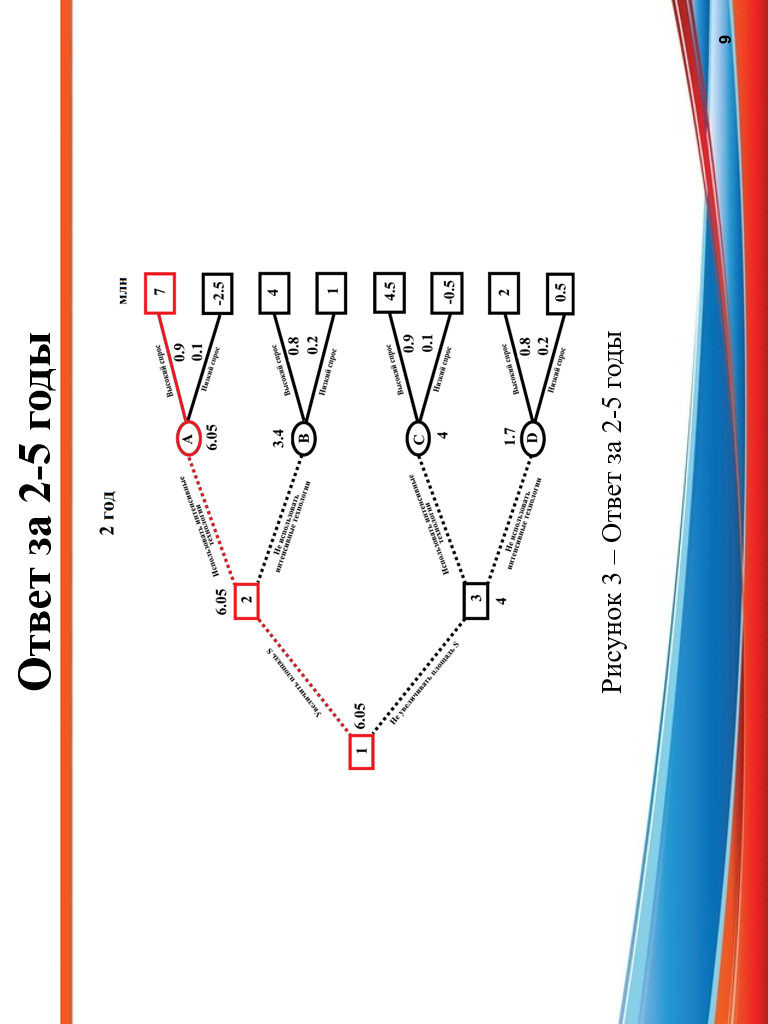


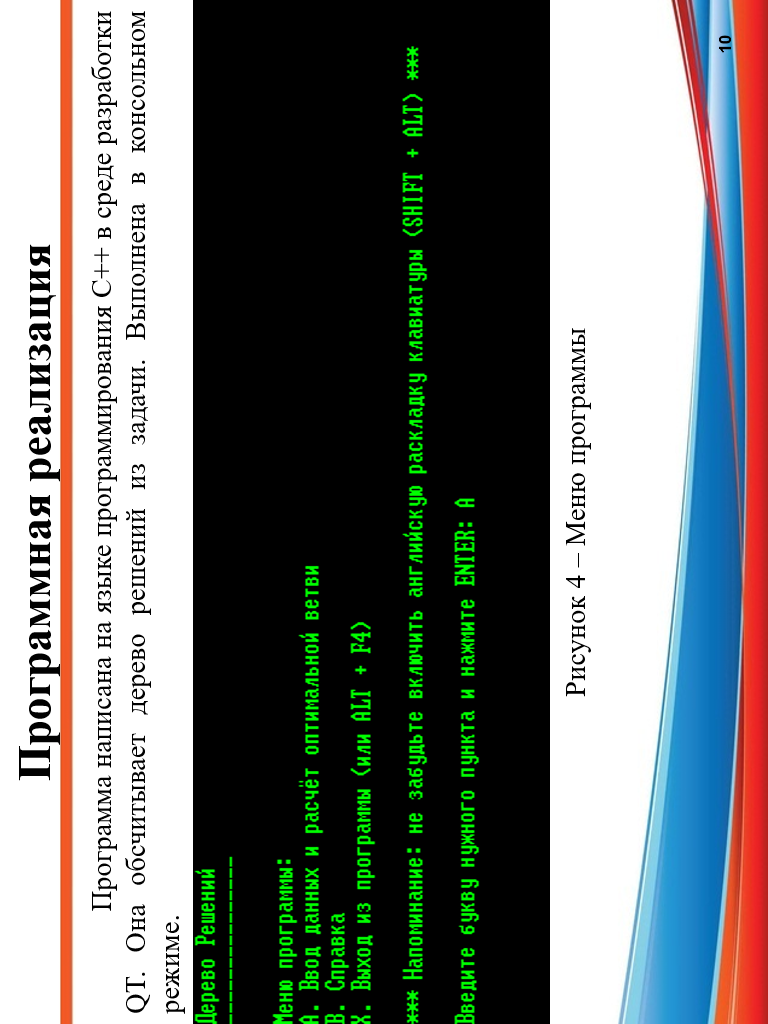


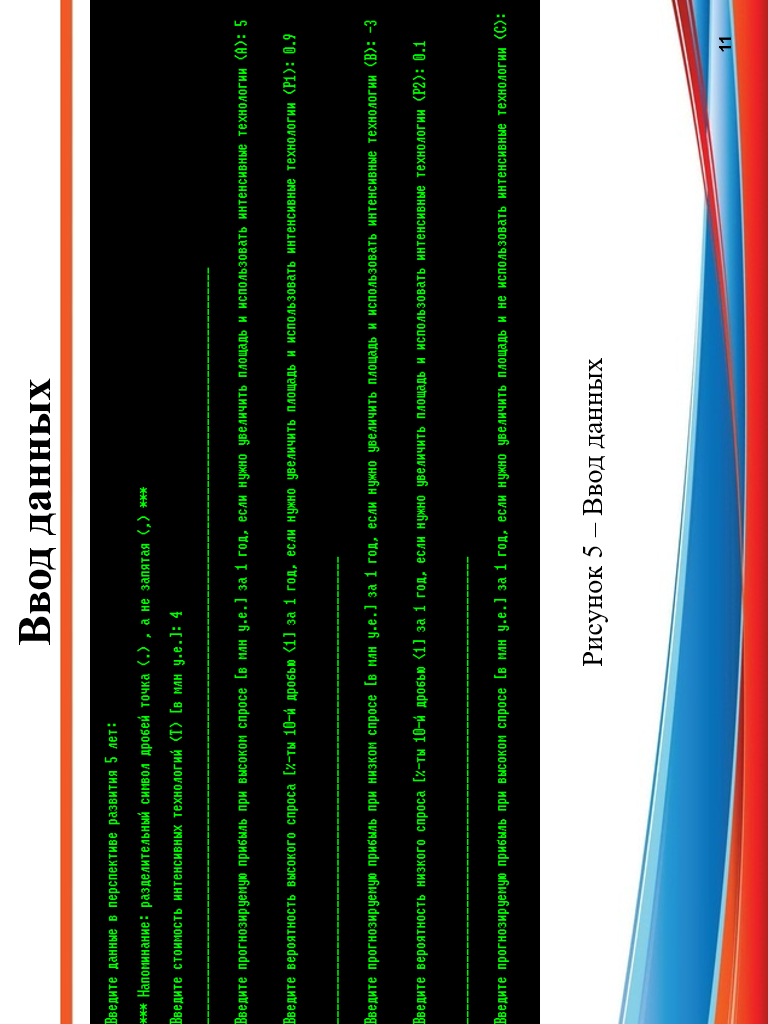


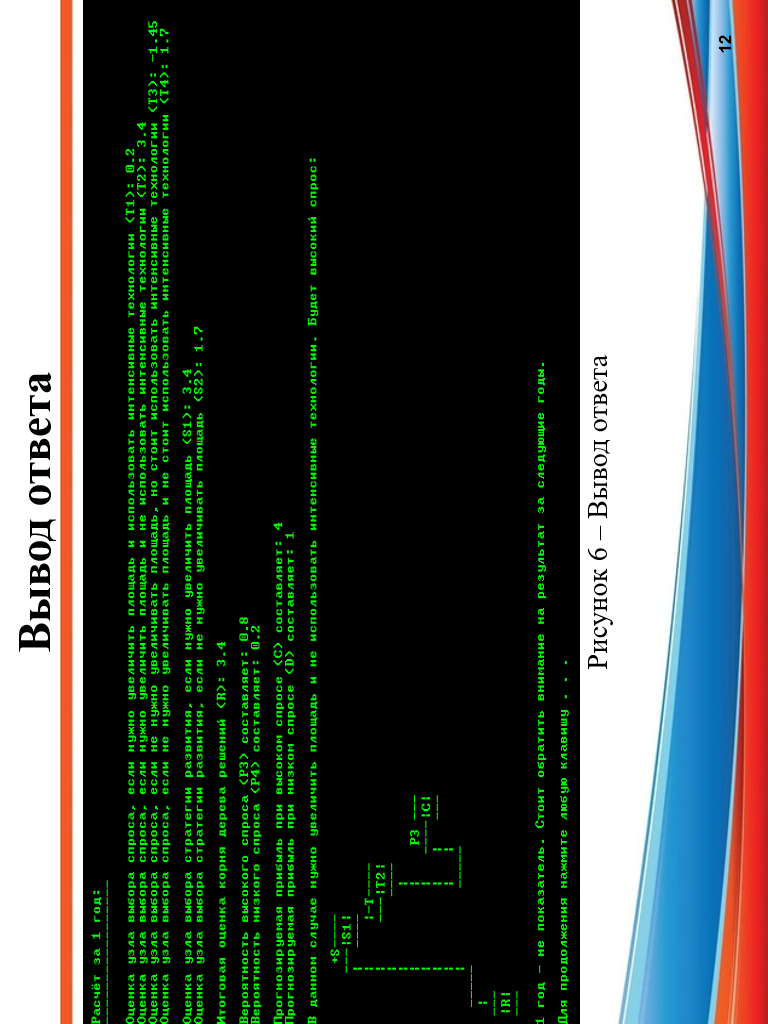


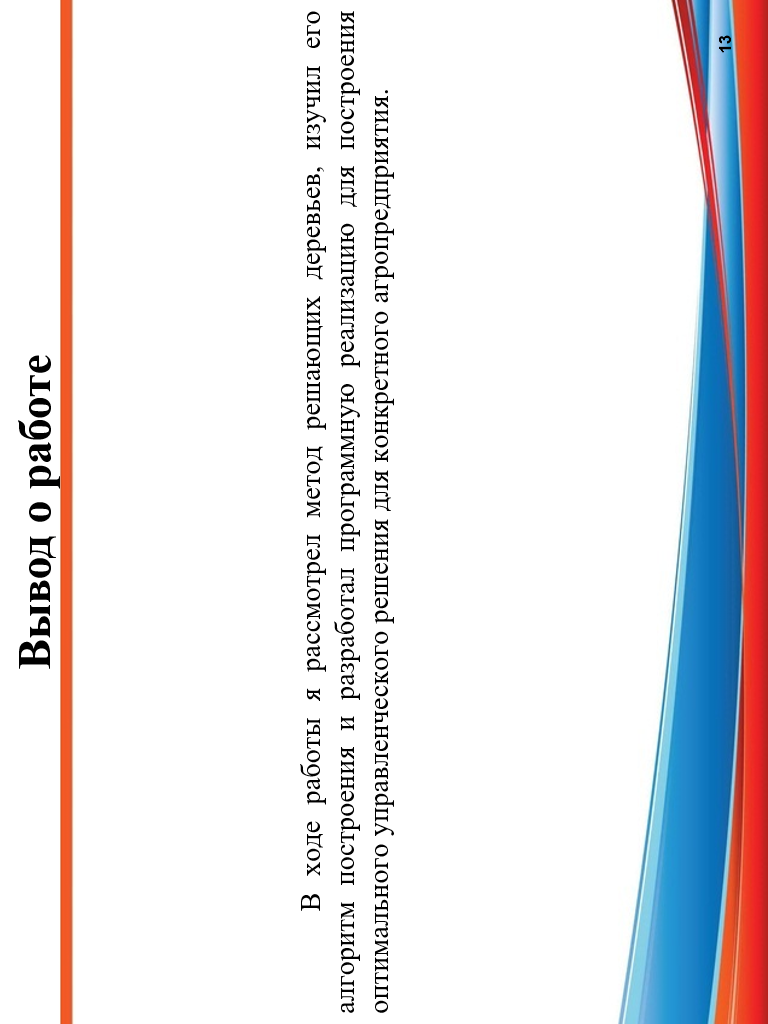


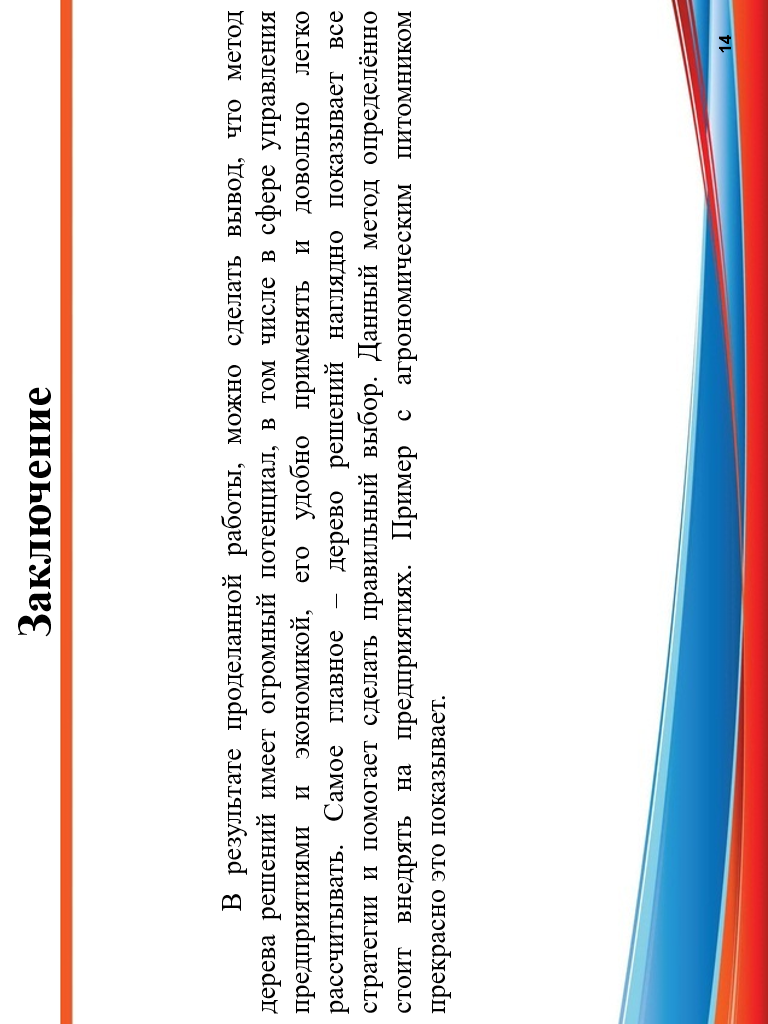


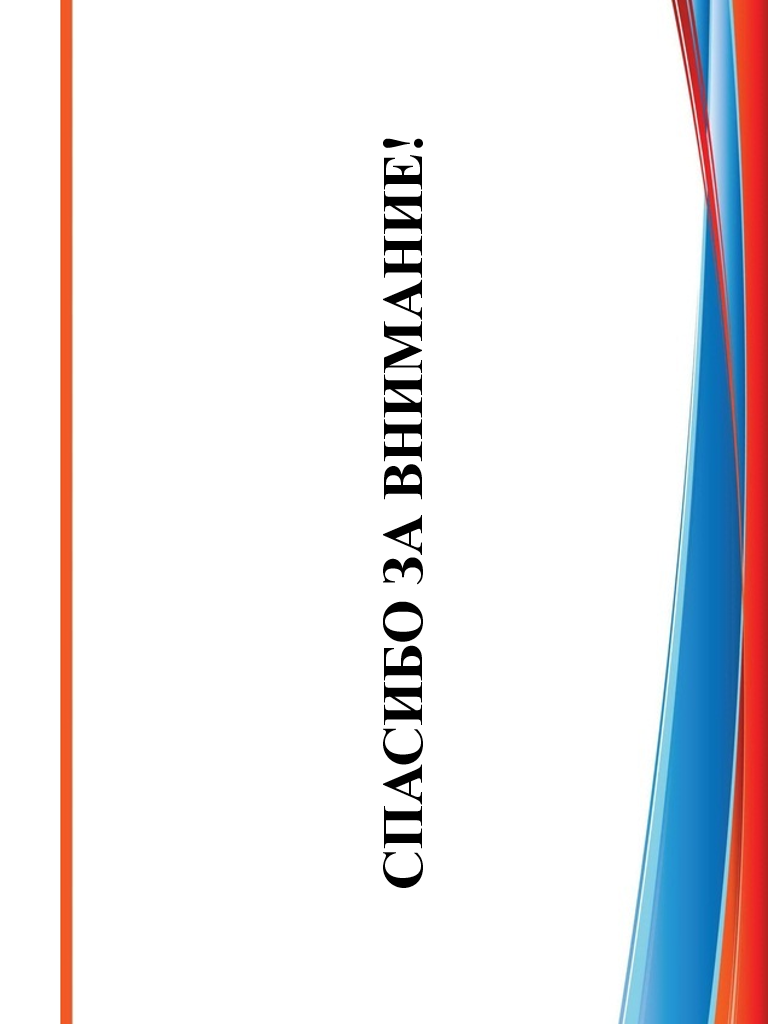






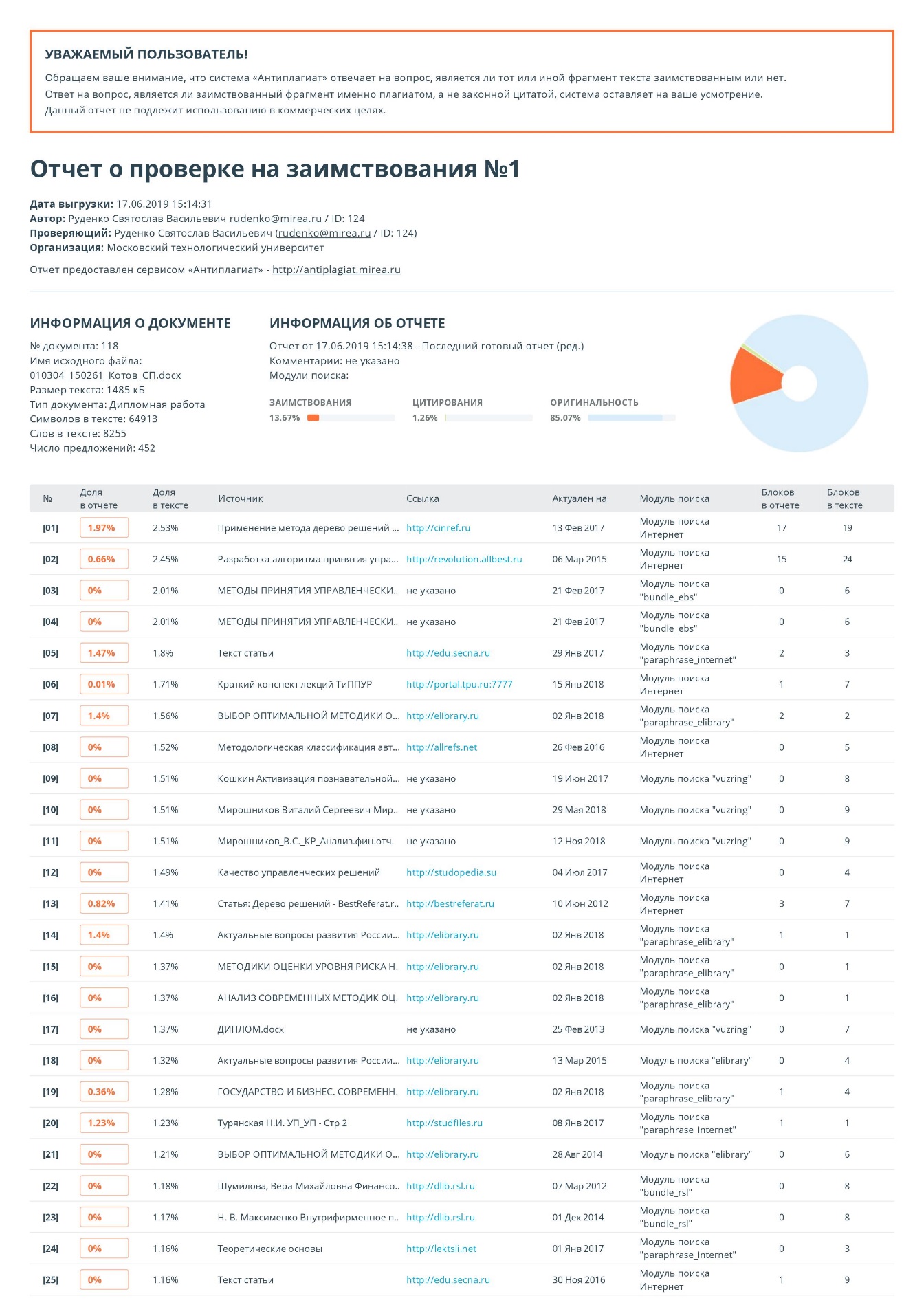




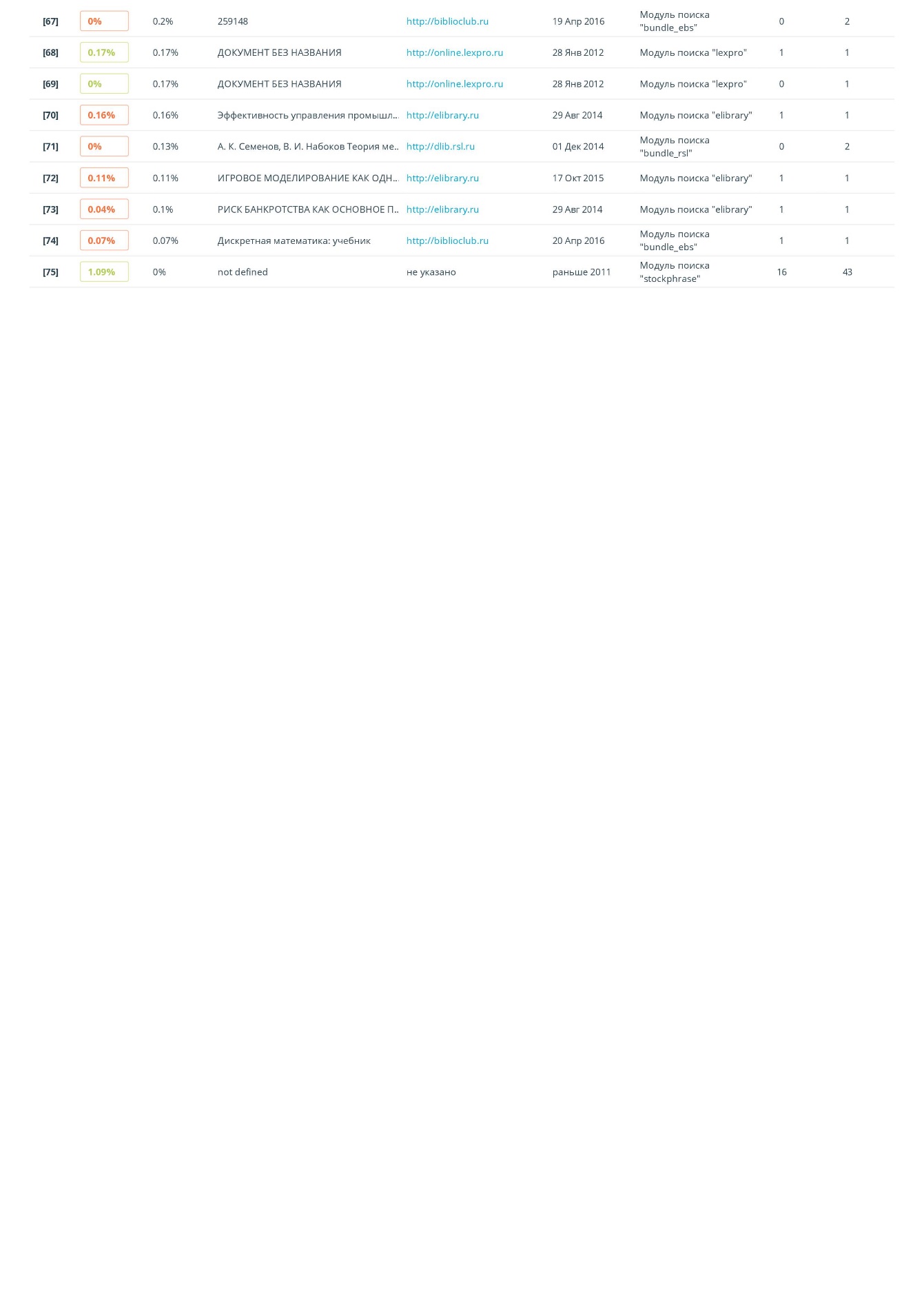


# **ПРИЛОЖЕНИЕ В.**

**Результат проверки на оригинальность текста**







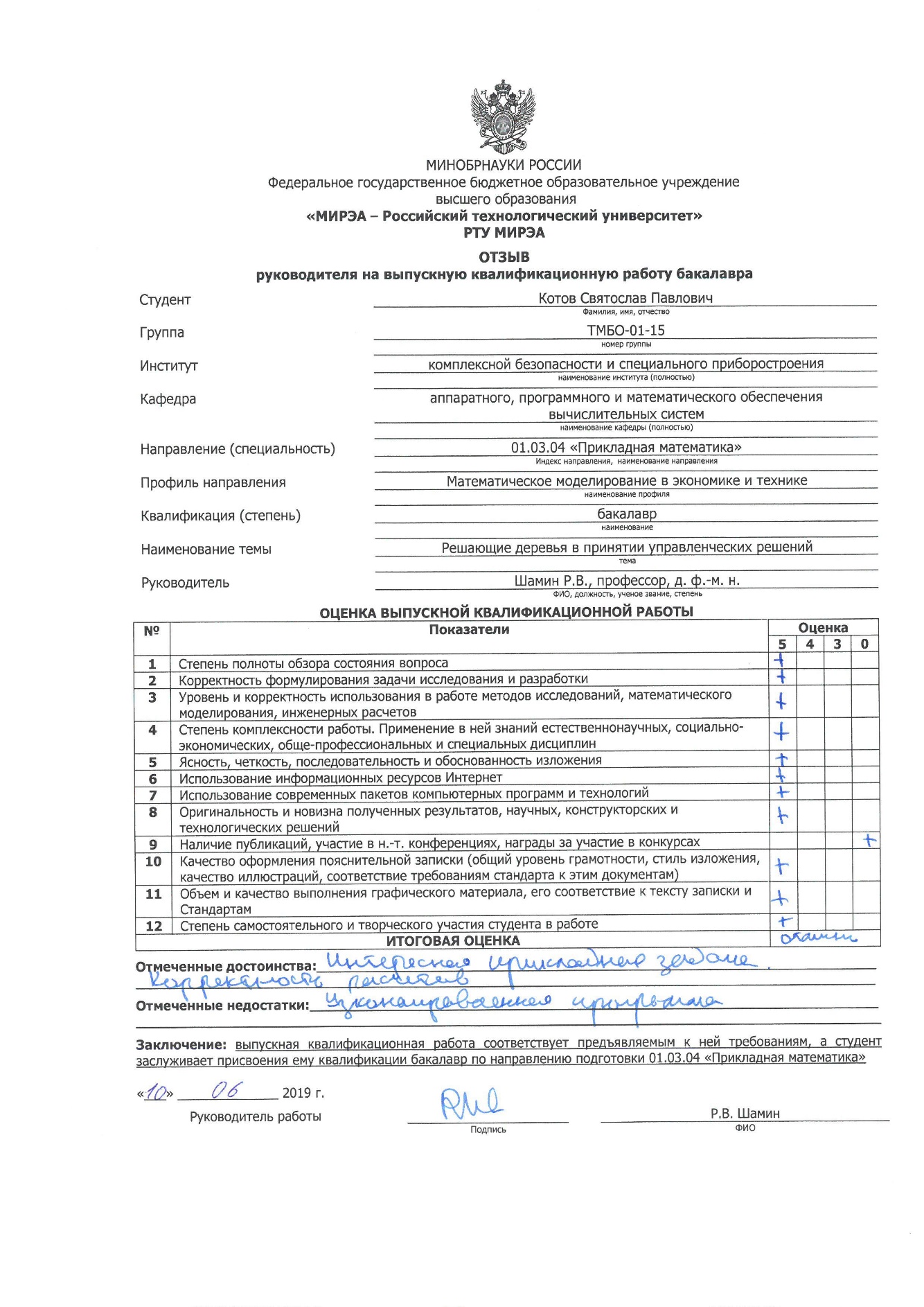
# **ПРИЛОЖЕНИЕ Г.**

**Заключение руководителя об оригинальности текста**



# **ПРИЛОЖЕНИЕ Д.**

**Отзыв научного руководителя**



1. Полную версию исходников, а также остальные файлы ВКР можно найти на

   GitHub по ссылке:

   <https://github.com/MrSlava2012z/VKR_010304_150261_Kotov_SP.git> [↑](#footnote-ref-1)