ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования**

**Национальный исследовательский университет
«Высшая школа экономики»**

**Московский институт электроники и математики**

**Факультет Прикладной математики и кибернетики**

**Кафедра Кибернетики**

Дипломная работа

по специальности 230201.65 «Информационные системы и технологии»

**Разработка математического обеспечения системы поддержки принятия решений для проектирования программ лояльности клиентов**

Студент группы МС91 Буянов Д.И.

Руководитель доцент, к.т.н.

 Белов А.В.

Зав. кафедрой профессор, д.т.н.

 Афанасьев В.Н.

Москва 2014Аннотация

на дипломную работу по специальности 230201.65«Информационные системы и технологии» студента группы МС-91 Буянова Дмитрия Игоревича на тему:

Разработка математического обеспечения системы поддержки принятия решений для проектирования программ лояльности клиентов (руководитель: доцент, к.т.н. Белов А.В.). 2014.

**Структура работы**: введение, 3 главы, заключение, список использованной литературы, 1 приложение. Работа содержит 18 рисунков, 1 таблицу и изложена на 60 страницах.

**Ключевые слова**: маркетинг, CRM, программа лояльности, лояльность, СППР, розничная торговля, retail, ROI.

**Объекты исследования:**маркетинговые системы, работающие с программами лояльности.

**Предмет исследования**: методы расчёта эффективности программ лояльности.

**Цель работы:**проектирование системы поддержки принятия решений для проектирования программ лояльности.

**Результат**:разработано алгоритмическое и математическое обеспечение, а также информационно-логическая модель базы данных СППР.

**Новизна:**разработаны простые методы оценки эффективности программ лояльности и отслеживания трендов в численности лояльных клиентов, ориентированные на розничную торговлю с коротким циклом продаж.

**Перспективы внедрения:**результат работы будет использован компанией MaykorExpertekдля создания СППР в составе модуля управления лояльностью для единой системы учета товародвиженияTradeHouse.

Developing of Mathematical Support for Decision Support System for Clients Loyalty Programs Design

by

D.I. Buyanov

# Abstract

**Objects of research:**marketing systems that use loyalty programs.

**Subject of research:** methods of loyalty programs’ effectiveness calculating.

**Objective:** designing of a decision support system for clients loyalty programs design.

**Result:** algorithmic and mathematical supports, as well as information-logical modelof decision support system database have been developed.

**Novelty:** simple methods of loyalty programs’ effectiveness estimation and trend tracking in change of number of loyal customers, that focus on retail trade with short sales cycle, have been developed.

**Implementation prospects:** the results of this wok will be used by Maykor Expertek company for creating a decision support system for loyalty management module in Trade House integrated accounting system.

Supervisor: assistant professor, Ph.D. A.V. Belov.

2014

# Оглавление

[\_Toc390142597](#_Toc390142597)[Оглавление 2](#_Toc390142598)

[Введение 2](#_Toc390142599)

[1. Обзор текущей ситуации в сфере программ лояльности 2](#_Toc390142600)

[1.1. Задачи и проблемы, стоящие перед современными программами лояльности 2](#_Toc390142601)

[1.2. Задачи системы поддержки принятия решений для проектирования программ лояльности 2](#_Toc390142602)

[1.3. Обзор и анализ систем клиентской лояльности, представленных на рынке 2](#_Toc390142603)

[1.4. Позиционирование проектируемой системы 2](#_Toc390142604)

[2. Разработка математического обеспечения системы поддержки принятия решений для проектирования программ лояльности клиентов 2](#_Toc390142605)

[2.1. Общая структура системы поддержки принятия решений при реализации программ лояльности 2](#_Toc390142606)

[2.2. Сегментация клиентов предприятия розничной торговли 2](#_Toc390142607)

[2.3. Отслеживание трендов изменения численности клиентских групп 2](#_Toc390142608)

[2.3. Расчёт количественных характеристик программ лояльности 2](#_Toc390142609)

[Таблица коэффициентов 2](#_Toc390142610)

[Поиск ближайшего значения в таблице коэффициентов 2](#_Toc390142611)

[Расчёт коэффициентов 2](#_Toc390142612)

[2.4. Расчёт эффективности программы лояльности с заданной количественной характеристикой 2](#_Toc390142613)

[2.5. Информационно-логическая модель Базы Данных системы поддержки принятия решений 2](#_Toc390142614)

[2.6. Отчёты 2](#_Toc390142615)

[2.7. Электронное рабочее место маркетолога 2](#_Toc390142616)

[3. Компьютерный эксперимент 2](#_Toc390142617)

[Начальные условия 2](#_Toc390142618)

[Интерфейс программы 2](#_Toc390142619)

[Ожидаемые результаты эксперимента 2](#_Toc390142620)

[Полученные результаты эксперимента: 2](#_Toc390142621)

[Заключение 2](#_Toc390142622)

[Список использованной литературы 2](#_Toc390142623)

[Приложение 1 2](#_Toc390142624)

# Введение

В последние годы всё больше растёт понимание того, что главным источником прибыли торговых предприятий (в том числе и розничных торговых сетей) являются постоянные покупатели. Соответственно, по-другому стали расставляться акценты в сфере маркетинга.

На место разовых акций, направленных на стимулирование сбыта какого-то товара или группы товаров пришла парадигма максимально подробной работы с каждым отдельным покупателем: накопительные скидки, система бонусов, сегментация клиентов по группам и уникальные для этих групп скидки…

Если прибавить сюда то обстоятельство, что большинство компаний уже обзавелось хотя бы простейшими техническими средствами для осуществления различных примитивных акций стимулирования сбыта (например, скидок) – то становится очевидным, что на современном этапе предприятиям требуется система, позволяющая, во-первых, гибко и своевременно формировать и изменять эти акции, а во-вторых, предсказывать эффективность акций.

В центре такой системы будет лежать понятие «программы лояльности» – набора маркетинговых мероприятий (таких, как скидки разного вида, бонусы, подарки и т.п.), направленных на удержание и привлечение постоянных клиентов.

Важно ещё раз подчеркнуть, что главным для нас становится не сиюминутная прибыль от мероприятия, а долговременная прибыль от того, что клиент стал постоянным – т.е. стал «лояльным».

Таким образом, **цель работы** – разработка системы поддержки принятия решений для проектирования программ лояльности клиентов.

**Объектами исследования** являются существующие маркетинговые системы, работающие с программами лояльности

**Предметом исследования** являются модели и методы расчёта эффективности программ лояльности.

В работе поставлена **задача** разработки математического обеспечения системы поддержки принятия решений.

**Первая глава**: описываются основные задачи и проблемы, стоящие перед программами лояльности. Объясняются основные понятия, относящиеся к лояльности. Приводится анализ основных типов решений, предлагаемых на рынке.

**Вторая глава**: ставится задача разработки СППР для проектирования программ лояльности. Описываются основные формулы и алгоритмы СППР. Приводятся информационно-логическая модель базы данных СППР и эскизы интерфейса рабочего места маркетолога.

**Третья глава**: проводится компьютерный эксперимент для иллюстрации работы СППР.

**Заключение**: приводятся основные выводы работы и определяется её новизна и значимость.

**Используемые сокращения**: СППР – система поддержки принятия решений;

БД – база данных

# 1. Обзор текущей ситуации в сфере программ лояльности

## 1.1. Задачи и проблемы, стоящие перед современными программами лояльности

Как уже было сказано, маркетинговые акции в последние годы стали ориентироваться больше на привлечение и удержание постоянных покупателей, нежели на простое стимулирование сбыта товаров. Этот сдвиг акцентов обусловлен целым рядом причин, среди которых можно выделить следующие:

* Цена и качество товаров у разных компаний выравниваются в ходе конкуренции, и требуются новые способы удержать клиента.
* Расходы компании на привлечение новых клиентов в значительной степени превосходят расходы на удержание имеющихся. При этом расходы на удержание падают с течением времени, т.к. с какого-то момента имеющиеся клиенты начинают покупать больше товаров и из более высоких ценовых категорий.

Отсюда вытекает важность понятия «лояльности» клиента, т.е. его готовности постоянно покупать товары именно у вашей компании.

Обеспечивается лояльность с помощью так называемых «программ лояльности» - совокупности маркетинговых мероприятий (например, скидок или бонусов), действующих определённый срок для определённых групп клиентов при выполнении ими заданных условий.

Очевидным образом, перед программами лояльности ставятся следующие задачи (в порядке значимости):

* Удержание имеющихся клиентов.
* Привлечение новых клиентов.
* Повышение клиентского спроса.

Вполне логично, что всё больше и больше компаний приобретают соответствующие программные средства и начинают применять программы лояльности. Но, в связи с этим, одно только наличие программы лояльности перестаёт быть значительным конкурентным преимуществом, и перед компаниями встаёт вопрос об их эффективности.

Под «эффективностью» здесь следует понимать количество удерживаемых и привлечённых клиентов, а также повышение спроса от использования программы лояльности.

Очевидным путём её повышения является максимальное использование данных о клиентах, имеющихся в информационной системе компании.

Этот тезис подтверждает исследование, проведённое в 2014 годуRetailSystemResearch[14] (рис.1).Из всех отделов маркетинга в своей деятельности:

* 25% опираются преимущественно на данные
* 58% опираются примерно поровну на данные и на интуицию, опыт маркетологов.
* 17% опираются только на интуицию и опыт маркетологов

*Рис. 1. Опора на данные*

Причём, более 40% «отстающих» (по росту продаж, относительно других) компаний опираются в маркетинге только на интуицию, в то время как «опережающие» других компании делают это только в 11% случаев (рис. 2).

*Рис. 2. Как принимаются решения*

Более того, в том же исследовании указывается, что, по мнению компаний, именно маркетингу сейчас наиболее необходимо использование (анализ) данных.

Но и 51% «опережающих», которые, как мы видим, уже используют данные в маркетинге, считают, что первоочерёдной задачей для аналитики является увеличение эффективности маркетинга через «лучшее применение специальных предложений» и «лучше измеряемуюлояльность».

При этом главными препятствиями на этом пути «опережающие считают» то, что исторически маркетологи предпочитают использовать интуицию, а не данные (51%) и что данные ещё нужно как-то извлечь из работающей системы (40%).

Также стоит отметить, что для «отстающих» главным препятствием является дороговизна и сложность имеющихся систем.

Важными же путями развития имеющихся систем являются аналитика (63% компаний считает это «очень важным»), отчётность (57%), визуализация данных (52%) и оповещения (48%).

Конечно, все эти показатели относятся к международным компаниям. Но, т.к. в России технологии в целом (как их качество, так и их распространённость), к сожалению, не опережают мирового уровня, то все негативные выводы, следующие из исследования, справедливы (и даже более справедливы) и для нашей страны.

Таким образом, можно выделить следующий ряд проблем, стоящих перед современными программами лояльности:

* Недостаточное использование накопленных данных о клиентах
* Дороговизна и сложность
* Необходимость увеличения эффективности программ лояльности и, как следствие, необходимость оценки этой эффективности
* Неудовлетворительная система отчётов и визуализация данных
* Недостаток оповещений и отслеживания трендов

Ответ на поставленные выше проблемы может дать система поддержки принятия решений для проектирования программ лояльности.

## 1.2. Задачи системы поддержки принятия решений для проектирования программ лояльности

Исходя из перечисленных в предыдущем разделе проблем, стоящих перед программами лояльности, можно выделить несколько ключевых задач системы поддержки принятия решений:

* Оценка эффективности программ лояльности (как уже применяемых, так и проектируемых) на основе накопленных данных о клиентах.
* Отслеживание трендов в поведении клиентов (например, в их спросе или в количестве постоянных клиентов) и оповещение о существенных в них изменениях.
* Система отчётов и визуализации данных.
* Простота и дешевизна

Для решения первой задачи важно, какой объём данных имеется у компаний. Достаточно ли его для более-менее точного и широкого анализа?

Здесь можно обратиться к исследованиюRetailSystemResearch за 2009 год[11]. В нём приводятся данные, согласно которым только 3% всех компаний собирали данные о клиентах менее 1 года.

В целом, результаты исследования представлены на рисунке (рис.3):

*Рис.3. Сколько лет собираются данные.*

Отсюда можно сделать вывод, что объём данных, доступных для анализа, у подавляющего большинства компаний очень велик.

## 1.3. Обзор и анализ систем клиентской лояльности, представленных на рынке

Программное обеспечение, представляющее возможности работы с программами лояльности, обычно входит в системы управления взаимоотношения с клиентами (традиционно обозначаемыеCRM).

По подходу к проектированию и оценке программ лояльности, их можно разделить на несколько групп:

1. «Платформы-инструментарии». Самым ярким представителем является, конечно, Salesforce.com ExactTarget Marketing Cloud[16]. Условно, к этой же группе можно отнести системы типа Микротест – Розничная Торговля, хотя они входят в более низкую ценовую категорию, и потому функционал их значительно уже.

В этих системах упор сделан на создание всевозможных средств анализа, визуализации и т.д. данных, с помощью которых можно (в идеале) получить любую информацию в любом разрезе и в любом виде.

Т.е. вместо того, чтобы предлагать пользователю нужные отчёты – они ему предлагают инструментарий создания любых отчётов, - в том числе и нужных.

Преимущества подобных систем:

* можно получить любую информацию для анализа;
* аналитик может использовать оригинальные методы и алгоритмы аналитической обработки данных.

Недостатки:

* непонятно, какую информацию получать;
* нужно затрачивать дополнительное время и усилия, чтобы «вынуть» нужную информацию из системы;
* высокая стоимость.
1. Модули в крупных системах. К этому классу систем относятся SAP Loyalty Management[18] (где просто необходимо покупать всю ERP - систему) и др.

Эти системы являются лишь сравнительно небольшим дополнением к какой-то более крупной системе и в значительной степени полагаются на средства, лежащие вне её. Они предоставляют некоторые удобные интерфейсы и возможности для работы с программами лояльности, но, как правило, не содержат каких-либо особенных аналитических инструментов, настроенных под тему лояльности.

Преимущества:

* можно получить почти любую информацию, обеспечиваемую «базовой» ERP–системой;
* достоверность информации.

Недостатки:

* непонятно, какую информацию получать;
* нужно затрачивать ещё больше времени и усилий, чем в первом случае, т.к. крупная система не предполагает извлечения из неё информации, необходимой конкретно для программ лояльности;
* высокая стоимость.
1. «Лёгкие» системы. К этому классу относятсяNimble (как крайний случай), SageCRM и др.

Эти системы направлены на самую базовую работу с программами лояльности: учёт клиентов, применение программ, простую отчётность.

Преимущества:

* можно получить простую информацию;
* лёгкость использования;
* малая стоимость.

Недостатки:

* отсутствие многообразной информации о программах лояльности, и следовательно – и невозможность осуществления адекватной оценки эффективности программ;
* отсутствие возможности отслеживания трендов и формирования оповещений.

Полученные результаты анализа программных систем можно свести в таблицу:

*Таблица 1. Сравнение систем, работающих с лояльностью.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Ценовая категория\* | Сложность | Оценка эффективности программ лояльности | Оповещения | Визуализация данных |
| ExactTarget Marketing Cloud | Высшая | Высшая | Общими аналитическими средствами (своими) | ? \*\* | Общими средствами (своими) |
| SAP Loyalty Management | Высокая | Высокая\*\*\* | Общими аналитическими средствами SAPERP + CRM | Есть | Общими средствами SAP ERP + CRM |
| Sugar | Высокая | Средняя | Почти нет (оценка по общим отчётам), только бонусные программы | Нет | Есть |
| ORACLE Loyalty | Средняя | Средняя | Общими аналитическими средствами (своими + Business Intelligence Applications) | Есть | Есть |
| Sage CRM | Средняя | Средняя | Почти нет (оценка по общим отчётам) | Нет | Есть |
| Megaplan | Низкая | Низкая | Нет | Нет | Есть |
| Nimble | Низкая | Низкая | Нет | Есть | Есть |

\*- примерная относительная оценка на 2013 год [20]

\*\*- скорее всего есть, но явно не указано

\*\*\*- с учётом того, что большинство функционала лежит в SAP ERP + CRM

Стоит сделать ещё одно замечание: почти все вышеперечисленные системы рассчитаны на длинный цикл продаж, т.е. ведение клиента ещё от стадии «заявки на покупку», когда клиенту требуется время на принятие решения о покупке и т.п. Отсюда – и развитый функционал переписки с клиентами и т.д.

Но всё это становится лишним для предприятий розничной торговли, с коротким циклом продаж и с ограниченным общением между клиентом и продавцом. Соответственно, в этом случае необходим совершенно другой пользовательский интерфейс, несколько иное представление данных и другие аналитические методы и алгоритмы.

## 1.4. Позиционирование проектируемой системы

По результатам проведенного анализа, можно предложить позиционирование проектируемой системы в соответствие со следующими принципами:

* Система должна предоставлять готовые инструменты оценки эффективности программ лояльности, особенно в момент их проектирования.
* Система должна отслеживать тренды в данных, напрямую связанных с использованием программам лояльности, и предоставлять оповещения о значимых в них изменениях.
* Система должна предоставлять готовые варианты отчётов и способы визуализации результаты аналитической обработки данных, существенных для оценки программ лояльности.
* Система должна ориентироваться на розничную торговлю с коротким циклом продаж.
* Система должна быть дешёвой и простой.

Иными словами, система должна предоставлять не средства анализа всех данных вообще, а готовые средства анализа нужных для программ лояльности в розничной торговле данных, в нужный момент и нужном месте.

# 2. Разработка математического обеспечения системы поддержки принятия решений для проектирования программ лояльности клиентов

## 2.1. Общая структура системы поддержки принятия решений при реализации программ лояльности

В соответствии с обозначенными задачами и позиционированием, система поддержки принятия решений структурно должна состоять из четырёхфункциональных компонент:

1. Блок «Сегментация клиентов». Отвечает на вопрос:на **какие группы клиентов**будут сфокусированы применяемые программы лояльности?
2. Блок «Отслеживание трендов в изменении численности и доходности разных групп клиентов». Отвечает на вопрос: **когда** применять программу лояльности?
3. Блок «Проектирование программы лояльности». Отвечает на вопрос: **какую** программу лояльности применить?
4. Блок «Формирование отчётов о результатах применения программы лояльности». Отвечает на вопрос: **какова эффективность** применённой программы лояльности?

Сразу же надо дать некоторые разъяснения касательно третьего блока.

Специфика программ лояльности состоит в том, что их формирование зависит от огромного множества факторов, таких как:

* Сезонность
* Наличие конкурентов на рынке
* Прогнозов по состоянию отрасли
* Ожидаемые изменения в численности групп клиентов
* Реклама товаров
* Общеэкономическая ситуация в стране: изменение доходов населения и т.п.
* Мода
* И другие...

Такое количество и разнообразие факторов, влияющих на принятие решения о формировании программы лояльности, приводит к тому, что данная сфера трудно поддаётся автоматизации и прогнозированию. И, как мы видели выше, большинство производителей программного обеспечения ограничиваются только созданием инструментальных платформ для формирования программ лояльности и возможностью ручного формирования отчётов, предусматриваемой информационной системой вообще.

В итоге остаются без ответа существенные, казалось бы, вопросы: **какой процент** скидки мы хотим дать? Какой **размер** бонуса сделать?

Ситуацию осложняет тот факт, что и сами маркетологи (как видно из приведённых выше исследований) представляют себе процесс выбора размера процента скидки, размера бонуса и т.д. лишь в общем виде, следуя в этом лишь личному неформализованному опыту и, возможно, ориентируясь на некоторые данные в информационной системе, выбираемые из общих соображений и того же личного опыта.

Возвращаясь к перечисленным выше факторам, очевидно, что одна их часть может быть введена и обработана в информационной системе предприятия, тогда как другая не может быть учтена в рамках корпоративной информационной системы.

Это обстоятельство приводит нас к следующему выводу: мы не можем дать точный прогноз эффективности данной программы лояльности.

Однако в подавляющем большинстве эксплуатируемых информационных систем всё же содержатся данные по ряду значимых факторов, а именно:

* Сезонность
* Покупательская активность клиентов и её изменения
* Социодемографические характеристики клиентов (возраст, пол)

Отсюда следует, что **задача** системы поддержки принятия решений для проектирования программ лояльности состоит в том, чтобы **проанализировать и наглядно представить** маркетологу все факторы, учтённые в системе, при этом **оставив финальное решение за человеком**, а также **дать примерный прогноз**, основанный на истории применения аналогичных программ лояльности.

## 2.2. Сегментация клиентов предприятия розничной торговли

Для сегментации клиентов был выбран подход, описанный в работе Полежаева И.Е.[6]

Исходными данными для сегментации клиентов являются:

* время последнего заказа (R),
* число заказов (F)
* количество потраченных клиентом средств (M) за всю историю взаимоотношений с клиентом.

Клиенты разбиваются на 3 группы: активные, переходные и спящие в соответствии со следующим алгоритмом:

1. Проходит разделение по показателю R.

Клиенты с Rбольшим, чем среднее значение по всей клиентской базе – активные.

Среди оставшихся клиентов спящими считаются те, у кого вероятность возвращения в группу активных клиентов становится постоянной.

Все остальные клиенты – переходные.

Затем разбиение уточняется с помощью критерия знаков.

1. Разделение по F.

Сначала выделяютсяпереходные и спящие клиенты, сделавшие за всю историю только лишь 1 заказ (т.к. обычно это случайные, зашедшие ради разнообразия клиенты).

Затем, при наличии большого количества переходных и спящих клиентов с 2 и более заказами, их делят на квантили или на несколько ещё более мелких частей, исходя из существенных различий (определяется графически или задаётся вручную) в вероятности снова стать активными.

В конце, полученные сегменты присоединяются к активным, переходным или спящим в соответствии с критерием знаков.

1. Разделение по M.

Происходит аналогично F.

В принципе, можно использовать и иной метод сегментации, если только границы между группами не изменяются при последующих группировках клиентской базы (т.к. иначе весь статистический анализ будет некорректен).

## 2.3. Отслеживание трендов изменения численности клиентских групп

Отслеживание трендов в изменении численности разных групп клиентов ведётся по четырём показателям:

1. Изменение количества клиентов в группах за фиксированный промежуток времени
2. Изменение скорости изменения количества клиентов в группах за одинаковые промежутки времени.
3. Изменение среднего объёма средств, затраченных клиентами в группах за фиксированный промежуток времени.
4. Изменение скорости изменения среднего объёма средств, затраченных клиентами в группах за фиксированный промежуток времени.

Пользователь задаёт предельную величину изменения, при превышении\снижении которой выводится уведомление, и величины используемых промежутков времени для отслеживания трендов и для вычисления среднего объёма средств.

1. Для первого показателя величина изменения считается по формуле:

(1)

Где - величина относительного изменения, %; – численность клиентов в группе в начале промежутка времени, чел.; – численность клиентов в группе в конце промежутка времени, чел.

Или, если величина изменения задана в абсолютных величинах:

 (2)

2. Для второго показателя:

 (3)

Где - величина относительного изменения, %; – численность клиентов в группе в начале текущего промежутка времени, чел.; – численность клиентов в группе в конце текущего промежутка времени, чел.; – численность клиентов в группе в начале предыдущего промежутка времени, чел.; - численность клиентов в группе в конце предыдущего промежутка времени, чел.

3. Для третьего показателя, аналогично:

 (4)

Где - величина относительного изменения, %; – средний объём средств, затраченных клиентами в группе на начальный момент промежутка времени, чел.; – средний объём средств, затраченных клиентами в группе на конечный момент промежутка времени, чел.

4. Для четвёртого показателя:

(5)

Где - величина относительного изменения, %; – средний объём средств, затраченных клиентами в группе на начальный момент текущего промежутка времени, чел.; – средний объём средств, затраченных клиентами в группе на конечный момент текущего промежутка времени, чел.; – средний объём средств, затраченных клиентами в группе на начальный момент предыдущего промежутка времени, чел.; - средний объём средств, затраченных клиентами в группе на конечный момент предыдущего промежутка времени, чел.

Метод, использующий расчетные формулы (1) – (5) являются простейшим и наименее затратным способом решения поставленной задачи. Предполагается, что в большинстве случаев такие грубые расчёты окажутся достаточными.

В случае, когда необходимы более тонкие оценки, можно воспользоваться методами аппроксимации. Они позволят учитывать не только текущий и прошедший период, но и данные за более длинные отрезки времени.

Для примера, разберём использование линейной аппроксимации методом наименьших квадратов для всех четырёх показателей.

1. Изменение количества клиентов

**Постановка задачи**. У нас имеется некоторый набор значений (в нашем случае это – количество клиентов в группе), измеренных в моменты времени

Необходимо аппроксимировать эти значения линейной функцией вида (6), где a иb – неизвестные коэффициенты, которые необходимо найти.

**Метод наименьших квадратов** состоит в нахождении минимума суммы квадратов отклонений yот аппроксимирующей функции , то есть минимума функции (7)

Для этого необходимо найти экстремумы :

Отсюда, формулы для нахождения коэффициентов:

 (8)

 (9)

Таким образом, мы получаем зависимость с известными коэффициентами. Множество Yздесь – это количество клиентов в группе, множество X–время (например, дни).

Из зависимости (6) мы можем вычислить значение для момента времени и сравнить его с реальным значением в том же моменте времени.

Если различие этих значений больше наперёд заданного значения p, указанного пользователем, (10), то выводится уведомление.

1. Изменение скорости изменения количества клиентов

Берутся множества значений и для разных множеств моментов времени и . По вышеописанным формулам для них строятся две аппроксимирующие функции и .

Чтобы найти скорость изменения количества клиентов, от обеих функций берутся производные: и

Сравниваются полученные значения и , и если их разность превышает наперёд заданное значениеp, указанное пользователем, (11), то выводится уведомление.

В оставшихся случаях рассуждения – аналогичны.

## 2.3. Расчёт количественных характеристик программ лояльности

Расчёт будем проводить на примере скидки. Другие же маркетинговые мероприятия могут быть легко приведены к тем же алгоритмам. Например, бонус можно перевести в % скидки товара, например: для товара ценой в 100 р. бонус в 10 р. тождественен скидке в 10%.

Изначально берётся сегментация клиентов на активных, переходных, спящих и неучтённых. Сегментацию на эти группы будем называть **основной**, т.к. она является самой значимой и только она является обязательной.

Затем каждая из групп основной сегментации делится на подгруппы по желаемым признакам: возраст, пол и т.д. Такую сегментацию мы будем называть **неосновной**.

### Таблица коэффициентов

Для каждой группы статистически находится:

* Доля клиентов, переходящих в активную группу без применения новой программы лояльности. Обозначается r.Отсюда берутся ожидаемые издержки от программы.
* Доля клиентов, переходящих в активную группу из-за новой программы лояльности. Обозначается k.Отсюда берётся ожидаемая выручка от программы.

Доли клиентов – это коэффициенты, задающиеся набором параметров: {основная группа, неосновные группы, категория товара, сезон, количественная характеристика программы лояльности}. Таким образом, можно составить многомерную таблицу коэффициентов.

При этом возможно некое подобие иерархии коэффициентов, где в корне всегда находится деление на активных\переходных\спящих\неучтённых (рис.4):

*Рис. 4. Иерархия коэффициентов.*

При выборе ранее не использовавшихся характеристик программы лояльности, для расчётов используется ближайшая по таблице известная информация.

### Поиск ближайшего значения в таблице коэффициентов

Поиск ближайшего значения в таблице ведётся по параметрам, от которых зависят коэффициенты. Порядок определяется одним из используемых алгоритмов:

1. Последовательный поиск. Используется поумолчанию, т.к. он не требует никаких дополнительных данных.

Перебираются все возможные значения параметров в установленной последовательности:

1. Количественная характеристика программы лояльности
2. Неосновные группы
3. Сезон
4. Категория товара
5. Основная группа

Т.е., если у нас нет информации по коэффициенту для скидки с набором параметров А = {активные, до 25 лет, мужчины, спортивная обувь, летний сезон, 15%}, то мы ищем информацию по коэффициентам в следующей последовательности:

{активные, до 25 лет, мужчины, спортивная обувь, летний сезон, **14%**}

{активные, до 25 лет, мужчины, спортивная обувь, летний сезон, **16%**}

{активные, до 25 лет, мужчины, спортивная обувь, летний сезон, **13%**}

{активные, до 25 лет, мужчины, спортивная обувь, летний сезон, **17%**}

…

{активные, **после 25 лет**, мужчины, спортивная обувь, летний сезон, **14%**}

{активные, **после 25 лет**, мужчины, спортивная обувь, летний сезон, **16%**}

…

{активные, **до 25 лет**, **женщины**, спортивная обувь, летний сезон, **14%**}

{активные, **до 25 лет**, **женщины**, спортивная обувь, летний сезон, **16%**}

…

И т.д.

1. Поиск в соответствии с весами.

Каждому параметру придаётся свой вес d, в зависимости от его важности. Например, для отличияколичественной характеристики программы скидки на 1% d = 10; для отличия пола d = 3; для отличия возраста на одно деление d = 2.

Для каждого набора параметров, отличного от исходного, близость Dвычисляется как сумма этих весов. Ближайшим значением считается то, у которого D– минимально.

Тогда близость к А = {активные, до 25 лет, мужчины, спортивная обувь, летний сезон, 15%}:

{активные, до 25 лет, мужчины, спортивная обувь, летний сезон, **16%**}D = 10

{активные, **после 25 лет**, **женщины**, спортивная обувь, летний сезон, **16%**} D = 15

Веса определяются:

1. Экспертами-маркетологами.
2. Через среднее относительное изменение.

Фиксируем все параметры, кроме того, для которого ищется вес по формуле:

 (12)

Гдеd – вес; – значения коэффициентов; – соответствующие значения параметров; – число сочетаний из n по 2.

Затем изменяем фиксированные параметры и снова ищем d. Итоговое d для данного параметраравно их среднему арифметическому.

Естественно, что второй способ определения веса требует больших (и разнообразных) накопленных данных.

### Расчёт коэффициентов

Для любой группы клиентов (кроме активной) существует некоторое количество клиентов, постоянно переходящих в группу активных (возвращающихся). Помимо этого, существуют скачки в количестве возвращающихся клиентов, вызванные применением новой программы лояльности.

Общий случай представлен на рис. 5. Здесь в конце марта была применена некоторая программа лояльности.

*Рис.5. Количество клиентов, переходящих из группы «переходные» в группу «активные»*

Разобьём график на 3 интервала.

1. На этом интервале ещё не было применено никакой программы лояльности.

Коэффициент r для него высчитывается очевидным способом: (13), где – суммарное количество клиентов, перешедших из группы «переходные» в группу «активные» на интервале I; – суммарное количество клиентов в группе «переходные» на интервале I.

1. На этом интервале применена программа лояльности и ещё явно виден её эффект.

Коэффициент k для него высчитывается по формуле: (14), где r – коэффициент, высчитанный для интервала I; – суммарное количество клиентов, перешедших из группы «переходные» в группу «активные» на интервале II; – суммарное количество клиентов в группе «переходные» на интервале II

1. Интервал, на котором явный эффект программы лояльности закончился. Сначала необходимо определить границы этого интервала. Здесь есть 2 случая:
2. Программа лояльности, применённая на интервале II, завершилась. Тогда интервал III начинается в момент завершения программы лояльности.

В этом случае для расчёта r используется формула: (15), где r – коэффициент, высчитанный для интервала III; – суммарное количество клиентов, перешедших из группы «переходные» в группу «активные» на интервале III; – суммарное количество клиентов в группе «переходные» на интервале III.

1. Программа лояльности, применённая на интервале II, не завершается. Тогда интервал III можно выделить исходя из того, что количество клиентов, переходящих в группу «активные», вышло на «плато», т.е. изменение этого количества стало меньше заданного пользователем числа p (см. пункт 2.3, отслеживание трендов).

В общем случае эффект от программы лояльности проявляется с задержкой,т.е. после применения программы лояльности есть некоторый период времени (на рисунке – апрель), в котором уже выполняется условие, что изменение количества клиентов становится меньше p. Следовательно, чтобы данный критерий не сработал раньше времени, необходимо, чтобы периоду III обязательно предшествовал период II, в котором изменение количества клиентов было больше p.

В общем случае, это «плато» может оказаться выше, чем аналогичное «плато» до применения программы лояльности. Поэтому коэффициент kдля программы лояльности корректируется по формуле: (16).

А коэффициент rстановится равен: (17)

Кроме того, выделив интервалы IIи III, можно получить величину L–интервал времени, на котором виден эффект программы лояльности (в нашем случае – это интервал II).

## 2.4. Расчёт эффективности программы лояльности с заданной количественной характеристикой

1. Проверка ограничений

Если есть данные о наценке на реализуемый товар MU (%).

Скидка D (%) - должна быть строго меньше наценки:D<MU (18). Для группы товаров используется минимальная наценка на товар, входящий в группу.

При использовании бонусной системы, D – сколько % от стоимости товара составляет бонус.

1. Расчёт эффективности E

Для расчёта эффективности программы лояльности нам следует рассмотреть влияние программы на все имеющиеся группы клиентов. Возьмём простейший случай, т.е. основную сегментацию.

1. Переходные клиенты. Эта группа является самой важной, т.к. именно от неё ожидается наибольший отклик на программу лояльности.

Часть этой группы станет активной и без применения программы лояльности, так что здесь она принесёт убыток в размере (19), где:

* – доля тех переходных клиентов, кто вернётся без новой программы лояльности

* - объём средств, потраченных переходными клиентами за период до новой программы лояльности (берётся последний период его активности)

Оставшаяся часть этой группы не затратила бы больше средств, если бы её не привлекли программой, так что выручка равна (20), где:

* – доля переходных клиентов, привлечённых новой программой лояльности

Помимо этого, новая программа лояльности может простимулировать в этой группе спрос и принести дополнительную выручку (21), где:

* – доля увеличения количества потраченных переходными клиентами средств из-за новой программы лояльности

1. Спящие клиенты. Сильного отклика на программу лояльности в этой группе не ожидается.

Все формулы аналогичны переходным клиентам, но, скорее всего, с меньшими коэффициентами:

Убыток: (22), где:

* – доля тех спящих клиентов, кто вернётся без новой программы лояльности

* - объём средств, потраченных спящими клиентами за период до новой программы лояльности (берётся последний период его активности)

Выручка: (23) и (24), где

* – доля спящих клиентов, привлечённых новой программой лояльности

* – доля увеличения количества потраченных спящими клиентами средств из-за новой программы лояльности

1. Неучтённые клиенты. В отличие от других групп, в которых можно узнать точные объёмы потраченных средств у каждого клиента, здесь мы можем оперировать только общими и средними значениями.

Можно предположить, что люди, рассчитывающие на дальнейшее сотрудничество с данной компанией, присоединились бы к программе лояльности (т.е. переместились бы в другие группы).

Следовательно, по поведению неучтённые клиенты аналогичны либо спящим, либо переходным клиентам. Тем не менее, если средние показатели для них (например, за счёт случайности) держатся примерно на одном уровне, то возможно записать убытки как (25), где

* - средний объём средств, потраченных неучтённым клиентом за период до новой программы лояльности

* – количество неучтённых клиентов, считается по количеству чеков (заказов) за установленный период времени

А выручка, по аналогии с предыдущими группами, равна:

 (26) и (27), где:

* – доля присоединившихся к программе лояльности из-за новой программы лояльности неучтённых клиентов

* – доля увеличения количества потраченных неучтёнными клиентами средств из-за новой программы лояльности

1. Активные клиенты.

По определению, все клиенты в этой группе гарантированно совершают покупки, так что убытки здесь равны (28), где:

* - объём средств, потраченных активными клиентами за период до новой программы лояльности

Выручка же в группе активных клиентов может быть обеспечена только повышением спроса, т.е.: (29), где:

* – доля увеличения количества потраченных активными клиентами средств из-за новой программы лояльности

В итоге, все эти рассуждения (формулы с 19 до 29) можно свести в единую формулу эффективности программы лояльности:

 (30)

При использовании неосновной сегментации, выражения типа заменяются суммой подобных выражений по каждой из подгрупп основной группы, например: (31), где

* - объём средств, потраченных клиентами из подгруппы №1 переходной группы за период до скидки

* - объём средств, потраченных клиентами из подгруппы №2 переходной группы за период до скидки

* – доля клиентов из подгруппы №1 переходной группы, привлечённых скидкой

* – доля клиентов из подгруппы №2 переходной группы, привлечённых скидкой

И т.д.

Возможно рассматривать эффективность не с точки зрения выручки и убытков, а, например, с точки зрения количества клиентов, перешедших из одной группы в другую (что формально больше соответствует специфике лояльности). Однако:

1. Этот фактор итак уже учтён в формуле (30)
2. В этом случае невозможно будет учесть убытки или, по крайней мере, сопоставить их с приобретениями

Поэтому мы выражаем эффективность именно через выручку и убытки.

Эффективность E в данном случае является, по сути, - ожидаемой прибылью от применения программы лояльности (но без учёта убытков на реализацию программы, данных о которых нет в информационной системе; например, трат на рекламу, печать материалов и пр.).

Также с помощью данной формулы возможно рассчитать, какие из известных наборов значений в таблице коэффициентов дают максимум эффективности (максимальную выручку и максимальный переход клиентов в группу активных). Т.е. перебрать все известные комбинации коэффициентов, подставить их в формулу и сравнить полученные значения E.

При сравнении значений E важно, чтобы они были рассчитаны с использованием одинаковых периодов, на которых берётся объём средств, потраченных клиентами.

Эффект от программы лояльности будет виден на интервале времени =, где – длины временных интервалов, на которых был виден эффект предыдущих программ лояльности для данной категории товара; n–количество этих временных интервалов.


## 2.5. Информационно-логическая модель Базы Данных системы поддержки принятия решений

На основании приведённых выше задач СППР и выкладок по разработке её математического обеспечения можно построить следующую информационно-логическую модель БД (рис.6):

*Рис. 6. Информационно-логическая модель БД.*

Часть схемы, помеченная цифрой I – это БД информационной системы компании, содержащая в том или ином виде учёт клиентов и механизмы применения программ лояльности. Модель данных информационной системы приведена для того, чтобы показать состав и связи основных информационных объектов, являющихся источником данных для СППР.

Часть схемы, помеченная цифрой II –это модель БД СППР, которую мы проектируем.

Ниже приведеноописаниеосновных информационных объектов с примечаниями:

Iчасть.

ReceiptsData–таблица с данными, содержащимися в чеках:

* idReceipt (FK) – идентификатор чека
* idProduct (FK) – идентификатор товара
* ProdCount – количество товара в чеке
* ProdPrice – цена товара

ProductData – таблица с данными о товарах:

* idProduct – идентификатор товара
* ProdPrice – цена товара
* ProdName – название товара

ProductsInGroups – таблица-связь между товарами и группами товаров:

* idProductGroup (FK) – идентификатор группы товаров
* idProduct (FK) – идентификатор товара

SeasonTypes – таблица с сезонами:

* idSeason – идентификатор сезона
* SeasonName – название сезона

Receipts – таблица чеков:

* idReceipt – идентификатор чека
* idClient (FK) – идентификатор клиента
* RecDate – дата чека
* RecSum – сумма по чеку

Clients – таблица клиентов:

* idClient – идентификатор клиента
* FirstName - имя
* LastName - фамилия
* MiddleName - отчество
* DateOfBirth – дата рождения
* Gender - пол
* DateOfReg – дата регистрации

ProductGroups – таблица групп товаров:

* idProductGroup – идентификатор группы товаров
* ProductGroupName – название группы товаров

II часть.

ClientGroups – таблица групп клиентов (группы здесь задаются сразу в максимальных подробностях, например: «активные, о 25 лет, мужчины»):

* idClientGroup – идентификатор группы клиентов
* ClGroupName – имя группы клиентов

ClientsInGroups – таблица-связь клиентов и групп клиентов:

* idClientGroup (FK) – идентификатор группы клиентов
* idClient (FK) – идентификатор клиента
* DateOfJoin – дата внесения клиента в группу

LoyaltyGroups – таблица программ лояльности:

* idLoyaltyProgram – идентификатор программы лояльности
* idClientGroup (FK) – идентификатор группы клиентов (0 – все группы)
* idProdGroup (FK) – идентификатор группы товаров (0 – все товары)
* DateOfStart – дата начала программы лояльности
* LoyProType(FK) – тип программы лояльности
* LoyProParam – параметр (количественная характеристика, например %) программы лояльности
* LoyProEffTime – интервал времени, на котором виден эффект программы лояльности

Coefficients – таблица коэффициентов:

* LoyProType (FK) – тип программы лояльности
* LoyProParam – параметр (количественная характеристика, например %) программы лояльности
* idClientGroup (FK) – идентификатор группы клиентов
* idProdGroup (FK) – идентификатор группы товаров
* idSeason (FK) – идентификатор сезона
* Coefficient– коэффициент
* DateOfCal – дата расчёта

Effectiveness – таблица эффективности программ лояльности:

* LoyProType (FK) – тип программы лояльности
* LoyProParam – параметр (количественная характеристика, например %) программы лояльности
* idProdGroup (FK) – идентификатор группы товаров
* idSeason (FK) – идентификатор сезона
* Effect – эффективность
* DateOfCal – дата расчёта

LoyaltyProgramsTypes – таблица типов программ лояльности:

* LoyProType– тип программы лояльности
* LoyProName – название программы лояльности

## 2.6. Отчёты

Система должна позволять генерировать отчёты со следующей информацией:

* Графики численности клиентов в группах с отметками о моментах применения программ лояльности

*Рис.7. Пример графика численности клиентов*

* Гистограммы изменения численности клиентов в группах с отметками о моментах применения программ лояльности

*Рис. 8. Пример гистограммы изменения численности клиентов*

* Графики выручки по группам с отметками о моментах применения программ лояльности
* Гистограммы изменения выручки по группам с отметками о моментах применения программ лояльности
* Гистограммы изменения скорости изменения численности клиентов в группах с отметками о моментах применения программ лояльности

*Рис. 9. Пример гистограммы скорости изменения численности клиентов*

* Гистограммы изменения скорости изменения выручки по группам с отметками о моментах применения программ лояльности
* График распределения выручки по клиентам в одной группе.

*Рис. 10. Пример графика распределения выручки по клиентам*

* Таблица коэффициентов

Таблица формируется путём фиксирования всех параметров программы лояльности, кроме двух, например:

*Рис. 11. Пример таблицы коэффициентов*

Или путём фиксирования меньшего количества параметров и выведения строчек со всеми нефиксированными параметрами:

*Рис. 12. Пример таблицы коэффициентов (альтернативный).*

* Таблица эффективности E

Таблица формируется для определённой категории товара.

*Рис. 13. Пример таблицы эффективности*

При этом все группы и данные возможно рассчитывать по определённым группам клиентов и товара, по их комбинациям и в целом (например: по группе «активные»; по группе «активные» + группе «переходные»; по всем клиентам и т.п.).

## 2.7. Электронное рабочее место маркетолога

Интерфейс электронного рабочего места маркетолога должен содержать следующие элементы:

* Инструментарий для создания программы лояльности, в которой выбираются параметры программы лояльности (этот пункт выходит за пределы данной работы, но наличие его подразумевается)
* Набор коэффициентов для выбранной программы лояльности, с обозначением группы параметров, задающей этот набор
* Кнопка доступа к таблице коэффициентов
* Эффективность E выбранной программы лояльности
* Кнопка доступа к таблице эффективности E
* Текущее количество клиентов в основных группах
* Кнопка доступа к графику распределения выручки по клиентам в одной группе
* Возможность просмотреть количество клиентов в неосновных группах
* Кнопка доступа к отчётам

Коэффициенты должно быть можно изменять вручную, - при этом произойдёт перерасчёт E.

*Рис. 14. Эскиз интерфейса рабочего места маркетолога.*

# 3. Компьютерный эксперимент

На языке программирования C# была написана программа, представляющая упрощённое рабочее место маркетолога для проведения компьютерного эксперимента. Исходный код программы представлен в приложении 1.

Целямиэкспериментаявляется:

* проверить правильность работы спроектированного математического обеспечения
* показать примерный ход работы системы поддержки принятия решений для проектирования программ лояльности клиентов и получаемые с помощью неё результаты.

Ряд алгоритмов (например, алгоритм сегментации) и интерфейсы были упрощены, т.к. не составляют принципиальную важность.

Начальные условия:

1. Одна группа товаров – обувь.
2. Один тип программ лояльности – скидка.
3. Клиенты сегментируются на: активных, переходных и спящих. Неучтённых нет.
4. Нет данных о количестве клиентов, возвращающихся без программы лояльности.
5. Расчёты производятся на промежутке времени в 1 месяц.
6. В базе данных есть информация о результатах скидок в 3% и 4%.

В базе данных представлена информация по 150 клиентам, совершающим покупки в среднем раз в неделю в течение 4 месяцев. Сумма покупок – случайная величина с равномерным распределением U(50, 150), т.е. находится в пределах от 50 до 150 условных денежных единиц.

Все клиенты стабильно совершают покупки раз в неделю, за исключением двух моментов:

1) Момент предыдущего применения скидки в 5%.

15 клиентов перестали делать покупки за 2 недели до этого, т.е. должны были попасть в группу спящих. После применения скидки – 5 клиентов стали активными.

Ещё 20 клиентов перестали делать покупки за 1 неделю до него, т.е. должны были войти в группу переходных. После применения скидки – 10 клиентов стали активными и ещё 10 спящими.

2) Момент проектирования новой скидки (текущий момент).

10 клиентов перестали делать покупки за 2 недели. То есть всего 30 клиентов должны попасть в группу спящих.

20 клиентов перестали делать покупки за 1 неделю. То есть 20 клиентов должны стать переходными.

Интерфейс программы:

*Рис.15. Интерфейс программы для компьютерного эксперимента*

Ожидаемые результаты эксперимента:

Применение программы лояльности со скидкой в 5%изначально не проанализировано. Из анализа её итогов можно получить два коэффициента: доли переходных и спящих клиентов, привлечённых этойпрограммой лояльности. Доля переходных = 0,5; доля спящих = 1/3 (или же 0,(3)). Эти доли должны быть добавлены в таблицу коэффициентов (после нажатия кнопки «Обновить таблицу коэффициентов»).

При проектировании новой программы лояльности мы выбираем группу товаров (обувь), тип программы лояльности (скидка) и вводим процент скидки (6).

Должны автоматически вывестись данные о количестве клиентов в группах на данный момент, а также должен произойти расчёт ожидаемой выручки и убытка по группам и в целом, эффективности программы лояльности.Для расчёта должны использоваться данные по программе лояльности со скидкой в 5%, т.к. она наиболее близка к желаемой программе с 6% скидки. Результаты расчёта также должны быть выведены на экран.

Должен произойти расчёт эффективностей при всех известных комбинациях коэффициентов. Его результаты должны быть внесены в таблицу эффективностей.

Полученные результаты эксперимента:

В таблицу коэффициентов добавлены строки с правильными долями переходных и спящих клиентов (рис. 16):

*Рис. 16. Таблица коэффициентов*

Расчёт эффективностибыл произведён с использованием данных о применении ближайшей известной программы лояльности, которой в данном случае является скидка на 5%. Информация об этомбыла выведена на экран (рис. 17).

*Рис. 17.Результаты моделирования*

Была рассчитана и таблица эффективностей (рис. 18).

*Рис. 18.Таблица эффективностей*

Верно посчитанные и выведенные на экран данные о группах клиентов, выручках, издержках и эффективности – являются **результатами компьютерного эксперимента**.

Также, реализация программы для компьютерного эксперимента, позволяет примерно оценить вычислительную сложность алгоритмов:

* Алгоритм сегментации клиентов требует перебора всех заказов клиентов за установленный период, который, скорее всего, окажется достаточно продолжительным (месяц и более).
* Алгоритм расчёта коэффициентов требует предварительной сегментации клиентов для конкретного периода и перебора всех клиентов в группах.

Тем не менее, расчёт коэффициентов требуется проводить только через некоторое время после применения новой программы лояльности. Поэтому в реальности он будет производиться в нерабочее время (например, по выходным или ночью) и не чаще раза в месяц.

* Алгоритм расчёта эффективности требует предварительной сегментации клиентов и требует единственного перебора всех заказов клиентов (для расчёта потраченного клиентами объёма средств за период), после чего расчёты ведутся уже с суммарными показателями и уже раннее просчитанными коэффициентами.

Данные об эффективности требуются только при проектировании новой программы лояльности. Поэтому предполагается, что достаточно будет обновлять таблицу эффективности раз в неделю, в нерабочее время, так как за неделю данные не будут существенно меняться.

# Заключение

Результатом проведённой работы является разработанное алгоритмическое и математическое обеспечение системы поддержки принятия решений для проектирования программ лояльности. Соответствующая СППР будет обладать следующими качествами, актуальность которых была показана в исследовании:

* Система предоставляет готовые инструменты оценки эффективности программ лояльности, как для прошедших, так и для проектируемых программ.
* Система может отслеживать тренды в данных, от которых зависит формирование программ лояльности, и оповещать о значимых в них изменениях.
* Система предоставляет варианты отчётов и визуализации данных, существенных для оценки программ лояльности.
* Система в первую очередьзаточена на розничную торговлю с коротким циклом продаж, хотя может использоваться и в других случаях.
* Система является достаточно простой и не требует больших ресурсов для реализации.

В основе системы лежит детерминированная математическая модель, которая может в дальнейшем стать основой для создания более совершенных, стохастических моделей.

Результат работы будет использован компанией MaykorExpertekдля создания СППР в составе модуля управления лояльностью для единой системы учета товародвиженияTradeHouse.

# Список использованной литературы

1. Абрамова Н.А. Как рассчитать экономическую эффективность скидки. // Профессиональное издательство. Журнал «Планово-экономический отдел» №3. 2011. Режимдоступа:http://www.profiz.ru/peo/3\_2011/kak\_rasshit\_effek\_skidki/ (датаобращения 18.05.2014)
2. Васин Ю.В., Лаврентьев Л.Г., Самсонов А.В. Эффективные программы лояльности. Как привлечь и удержать клиентов. 4-е издание. // Москва: «Альпина Бизнес Букс», 2007. – 152 с.
3. Дейан А., Троадек А., Троадек Л. Стимулирование сбыта. 9-е издание. // СПб.: «Нева», 2003. – 128 с.
4. Дымшиц М.Н. Потребительская лояльность. Механизмы повторной покупки. // Москва: «Веришна», 2007 – 196 с.
5. Макаров О., Широганова Т. Статья: «Не всё то золото, что блестит». // Журнал «Я — номер один». 2011. Режимдоступа:http://www.topclassgroup.ru/know/model/programmy-loyalnosti-razrushayuschie-loyalnost.php (датаобращения 18.05.2014)
6. Полежаев И.Е. Метод сегментации клиентских баз данных на основе жизненного цикла клиента // Исследовано в России. Электронный научный журнал. Том 9. 2006. Режим доступа: http://www.sci-journal.ru/articles/2006/200.pdf (дата обращения 18.05.2014)
7. Просветов И. Какова должна быть комбинация скидок и рекламы? // Forbes. 2009. Режимдоступа: http://m.forbes.ru/article.php?id=7715(датаобращения 18.05.2014)
8. Романов А.А., Паньков А.В. Маркетинговые коммуникации. // М.: «Эксмо», 2006. — 432 с.
9. Рыбченко С.А., Евстигнеева Т.В. Методы стимулирования сбыта. Учебное пособие. 1-е издание. // Ульяновск: УлГТУ, 2007. – 184 с.
10. Baird N., Rowen S. Retail Marketing 2013: Organizational Drift. Benchmark Report. // Retail Systems Research. 2013. Режим доступа: https://www.rsrresearch.com/2013/08/27/retail-marketing-2013-organizational-drift/ (дата обращения 18.05.2014)
11. Baird N., Rowen S. CRM and Loyalty 2009: Increasing Relevance to Drive Customer Demand // Retail Systems Research. 2009. Режим доступа: http://www.rsrresearch.com/2009/08/15/crm-and-loyalty-2009-increasing-relevance-to-drive-customer-demand/ (дата обращения 18.05.2014)
12. Baird N. Getting loyalty programs back to loyalty. Benchmark study. // Retail Systems Research. 2007. Режим доступа: http://www.rsrresearch.com/2007/08/01/getting-loyalty-programs-back-to-loyalty/ (дата обращения 18.05.2014)
13. Galyamov A.F., Tarkhov S.V. DECISION SUPPORT IN CUSTOMER RELATIONSHIP MANAGEMENT BASED ON INTEGRATED CUSTOMER SERVICE CENTRE. // Электронныйнаучныйжурнал«Modern problems of science and education» №4. 2013. Режимдоступа:http://www.science-education.ru/en/110-9978 (датаобращения 18.05.2014)
14. Kilcourse B., Rosenblum P. Retail Analytics Moves To The Frontline. BenchmarkReport. // Retail Systems Research. 2014. Режимдоступа: http://www.rsrresearch.com/2014/01/08/retail-analytics-moves-to-the-frontline-benchmark-2014/ (датаобращения 18.05.2014)
15. Мегаплан. Возможности. //Режим доступа: http://www.megaplan.ru/facilities/ (дата обращения 18.05.2014)
16. ExactTarget Marketing Cloud. Data & Analytics in Action. // Режимдоступа: http://www.exacttarget.com/products/data-analytics/ (датаобращения 18.05.2014)
17. ORACLE LOYALTY ANALYTICS // Oracle. 2009. Режимдоступа: http://www.oracle.com/us/products/middleware/bus-int/064330.pdf(датаобращения 18.05.2014)
18. SAPLoyaltyManagement. // Режимдоступа: http://www.sap.com/solution/lob/marketing/software/customer-loyalty-management/index.html (датаобращения 18.05.2014)
19. SmallBusinessLoyalty – Sugar Card. // Режимдоступа: https://sugarcard.co.nz/ (датаобращения 18.05.2014)
20. What Does CRM Actually Cost? // CrmSwitch. 2013. Режим доступа: http://www.crmswitch.com/crm-cost/crm-cost-list/(дата обращения 18.05.2014)

# Приложение 1

Исходный код программы для компьютерного эксперимента:

using System;

using System.Collections.Generic;

using System.ComponentModel;

usingSystem.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

namespace DSS

{

publicpartialclassForm1 : Form

 {

string conString = "User Id=root;password=lbvf;Persist Security Info=True;server=localhost;database=sppr";

public Form1()

 {

 InitializeComponent();

 }

privatevoid button1\_Click(object sender, EventArgs e)

{

//Узнаём, когда применялись программы лояльности

DataTable dt = newDataTable();

MySql.Data.MySqlClient.MySqlCommand mcommand = new MySql.Data.MySqlClient.MySqlCommand();

 mcommand.CommandText = "SELECT idClientGroup, DateOfStart, LoyProParam FROM sppr.loyaltygroups";

 mcommand.Connection = new MySql.Data.MySqlClient.MySqlConnection(conString);

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 dt.Load(mreader);

 }

finally

 {

 mcommand.Connection.Close();

 }

DateTime lpTime = Convert.ToDateTime(dt.Rows[0][1].ToString());

DateTime startTime = lpTime.AddDays(-30);

//Берём данные по клиентам (здесь - за месяц до и за месяц после)

//Считаем среднюю частоту покупки (получится 1 раз в неделю),

//проверяем клиентов, кто

//уже 1 неделю не покупает - они переходящие, кто 2 - те спящие

mcommand.CommandText = "SELECT COUNT(idReceipt) FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = startTime;

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 dt.BeginLoadData();

ArrayList al = newArrayList();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while(mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double count = 0;

for (int i = 0; i < al.Count; i++ )

 {

 count += Convert.ToDouble(al[i]);

 }

 count = count / al.Count;

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-4) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countClients = al.Count;

ArrayList al2 = newArrayList();

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-2) \* lpTime.Subtract(startTime).Days/count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al2.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al2.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countSleepPre = countClients - al2.Count;

//double countSleepPre = dt.Rows.Count;

ArrayList al3 = newArrayList();

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-1) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al3.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al3.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countTransPre = countClients- al3.Count - countSleepPre;

//Сравниваем с тем, что произошло в периоде после скидки

lpTime = lpTime.AddDays(31);

 startTime = startTime.AddDays(31);

ArrayList arSleep = newArrayList();

for (int i = 0; i < al.Count; i++)

 {

 arSleep.Add(al[i]);

 }

ArrayList arTrans = newArrayList();

for (int i = 0; i < al.Count; i++)

 {

 arTrans.Add(al[i]);

 }

for (int i = 0; i < al2.Count; i++)

 {

if(arSleep.Contains(al2[i]))

 arSleep.Remove(al2[i]);

 }

for (int i = 0; i < al3.Count; i++)

 {

if(arTrans.Contains(al3[i]))

 arTrans.Remove(al3[i]);

 }

for (int i = 0; i < arSleep.Count; i++)

 {

if (arTrans.Contains(arSleep[i]))

 arTrans.Remove(arSleep[i]);

 }

string com = "";

for (int i = 0; i < arSleep.Count; i++)

 {

 com += "'" + arSleep[i].ToString() + "'";

if (i != (arSleep.Count - 1))

 {

 com += ", ";

 }

 }

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) AND idClient IN ("+com.ToString()+") GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-2) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countSleepPost = al.Count;

 com = "";

for (int i = 0; i < arTrans.Count; i++)

 {

 com += "'" + arTrans[i].ToString() + "'";

if (i != (arTrans.Count - 1))

 {

 com += ", ";

 }

 }

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) AND idClient IN (" + com.ToString() + ") GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-1) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

//dt.Load(mreader);

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

doublecountTransPost =al.Count;

//Считаем и заносим в таблицу доли вернувшихся в группу активных

double procTrans = countTransPost/ countTransPre;

double procSleep = countSleepPost / countSleepPre;

 mcommand.Parameters.Clear();

 mcommand.CommandText = "INSERT INTO sppr.coefficients VALUES ('1','5','2','1','0','@c1','01.07.26'), ('1','5','3','1','0','@c2','01.07.26')";

 mcommand.Parameters.Add("@c1", MySql.Data.MySqlClient.MySqlDbType.Decimal);

 mcommand.Parameters.Add("@c2", MySql.Data.MySqlClient.MySqlDbType.Decimal);

 mcommand.Parameters["@c1"].Value = Convert.ToDecimal(procTrans);

 mcommand.Parameters["@c2"].Value = Convert.ToDecimal(procSleep);

try

 {

 mcommand.Connection.Open();

 mcommand.EndExecuteNonQuery(mcommand.BeginExecuteNonQuery());

 }

catch (Exception ex)

 {

 }

finally

 {

 mcommand.Connection.Close();

 }

 }

privatevoid label2\_Click(object sender, EventArgs e)

 {

 }

privatevoid comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

 {

 }

privatevoid procTbx\_TextChanged(object sender, EventArgs e)

 {

//Заполняемвсепоказатели

 MySql.Data.MySqlClient.MySqlCommand mcommand = new MySql.Data.MySqlClient.MySqlCommand();

 mcommand.Connection = new MySql.Data.MySqlClient.MySqlConnection(conString);

DateTime lpTime = Convert.ToDateTime("07.09.2001");

DateTime startTime = lpTime.AddDays(-30);

DataTable dt = newDataTable();

 mcommand.CommandText = "SELECT COUNT(idReceipt) FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = startTime;

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 dt.BeginLoadData();

ArrayList al = newArrayList();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double count = 0;

for (int i = 0; i < al.Count; i++)

 {

 count += Convert.ToDouble(al[i]);

 }

 count = count / al.Count;

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-4) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countClients = al.Count;

ArrayList al2 = newArrayList();

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-2) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al2.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al2.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countSleepPre = countClients - al2.Count;

ArrayList al3 = newArrayList();

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT idClient FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) GROUP BY idClient";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-1) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 dt.Clear();

 al3.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al3.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double countTransPre = countClients - al3.Count - countSleepPre;

ArrayList arSleep = newArrayList();

for (int i = 0; i < al.Count; i++)

 {

 arSleep.Add(al[i]);

 }

ArrayList arTrans = newArrayList();

for (int i = 0; i < al.Count; i++)

 {

 arTrans.Add(al[i]);

 }

for (int i = 0; i < al2.Count; i++)

 {

if (arSleep.Contains(al2[i]))

 arSleep.Remove(al2[i]);

 }

for (int i = 0; i < al3.Count; i++)

 {

if (arTrans.Contains(al3[i]))

 arTrans.Remove(al3[i]);

 }

for (int i = 0; i < arSleep.Count; i++)

 {

if (arTrans.Contains(arSleep[i]))

 arTrans.Remove(arSleep[i]);

 }

 actTbx.Text = (countClients - countTransPre - countSleepPre).ToString();

 traTbx.Text = countTransPre.ToString();

 sleTbx.Text = countSleepPre.ToString();

//Выбираемближайшийимеющийсякоэффициент

double col = Convert.ToDouble(procTbx.Text);

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT LoyProParam, idClientGroup, Coefficient FROM sppr.coefficients WHERE LoyProType = '1' AND idProdGroup = '1' AND idSeason = '0'";

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 dt.Load(mreader);

 }

finally

 {

 mcommand.Connection.Close();

 }

double colSel = 0;

for (int i = 0; i < 100; i++)

 {

for (int j = 0; j < dt.Rows.Count; j++)

 {

if ((col+i) == (int)dt.Rows[j][0])

 {

 colSel = (col + i);

 }

 }

for (int j = 0; j < dt.Rows.Count; j++)

 {

if ((col-i) == (int)dt.Rows[j][0])

 {

 colSel = (col - i);

 }

 }

if (colSel != 0)

break;

 }

 textBox1.Text = colSel.ToString();

 resgrpTbx.Text = "Обувь";

 restypeTbx.Text = "Скидка";

double retTrans=0,retSleep=0;

for (int i = 0; i < dt.Rows.Count; i++ )

 {

if ((int)dt.Rows[i][0] == 5 && (int)dt.Rows[i][1] == 2)

 {

 textBox2.Text = dt.Rows[i][2].ToString();

try

 {

 retTrans = Convert.ToDouble(dt.Rows[i][2]);

 }

catch(Exception ex)

 {

 }

 }

if ((int)dt.Rows[i][0] == 5 && (int)dt.Rows[i][1] == 3)

 {

 textBox4.Text = dt.Rows[i][2].ToString();

 retSleep=Convert.ToDouble(dt.Rows[i][2]);

}

 }

//Получаем данные о тратах клиентов в каждой группе

//Причём для переходных и спящих берём последний месяц их активности

string com = "";

for (int i = 0; i < al3.Count; i++)

 {

 com += "'" + al3[i].ToString() + "'";

if (i != (al3.Count - 1))

 {

 com += ", ";

 }

 }

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT Sum(RecSum) FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) AND idClient IN (" + com.ToString() + ")";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = lpTime.AddDays((-2) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime;

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double activeSum = Convert.ToDouble(al[0]);

 com = "";

for (int i = 0; i < arTrans.Count; i++)

 {

 com += "'" + arTrans[i].ToString() + "'";

if (i != (arTrans.Count - 1))

 {

 com += ", ";

 }

 }

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT Sum(RecSum) FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) AND idClient IN (" + com.ToString() + ")";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = startTime.AddDays((-2) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime.AddDays((-2) \* lpTime.Subtract(startTime).Days / count);

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double transSum = Convert.ToDouble(al[0]);

 com = "";

for (int i = 0; i < arSleep.Count; i++)

 {

 com += "'" + arSleep[i].ToString() + "'";

if (i != (arSleep.Count - 1))

 {

 com += ", ";

 }

 }

 mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT Sum(RecSum) FROM sppr.receipts WHERE (RecDate>@st AND RecDate<@ft) AND idClient IN (" + com.ToString() + ")";

 mcommand.Parameters.Add("@st", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters.Add("@ft", MySql.Data.MySqlClient.MySqlDbType.DateTime);

 mcommand.Parameters["@st"].Value = startTime.AddDays((-3) \* lpTime.Subtract(startTime).Days / count);

 mcommand.Parameters["@ft"].Value = lpTime.AddDays((-3) \* lpTime.Subtract(startTime).Days / count);

 al.Clear();

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 {

while (mreader.Read())

 al.Add(mreader.GetInt32(0));

 }

 }

finally

 {

 mcommand.Connection.Close();

 }

double sleepSum = Convert.ToDouble(al[0]);

//Считаеможидаемуювыручкуиубыток

 textBox7.Text = (transSum \* retTrans \* (1-col/100)).ToString();

 textBox9.Text = (sleepSum \* retSleep \* (1 - col / 100)).ToString();

 textBox11.Text = (activeSum \* (col / 100)).ToString();

 textBox14.Text = (transSum \* retTrans \* (1 - col / 100) + sleepSum \* retSleep \* (1 - col / 100)).ToString();

 textBox15.Text = (activeSum \* (col / 100)).ToString();

 textBox16.Text = (transSum \* retTrans \* (1 - col / 100) + sleepSum \* retSleep \* (1 - col / 100) - activeSum \* (col / 100)).ToString();

//Добавляем рассчитанную эффективность в таблицу

 mcommand.Parameters.Clear();

mcommand.CommandText = "INSERT INTO sppr.effectiveness VALUES ('1','"+textBox1.Text+"','1','0','"+(transSum \* retTrans \* (1 - col / 100) + sleepSum \* retSleep \* (1 - col / 100) - activeSum \* (col / 100)).ToString()+"','01.09.07')";

try

 {

 mcommand.Connection.Open();

 mcommand.ExecuteNonQuery();

 }

catch (Exception ex)

 {

 }

finally

 {

 mcommand.Connection.Close();

}

//Сразу же считаем эффективности для другого наборов коэффициентов

mcommand.Parameters.Clear();

 mcommand.CommandText = "SELECT LoyProParam, idClientGroup, Coefficient FROM sppr.coefficients WHERE LoyProType = '1' AND idProdGroup = '1' AND idSeason = '0' ORDER BY LoyProParam ASC, idClientGroup ASC";

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 dt.Load(mreader);

 }

finally

 {

 mcommand.Connection.Close();

 }

double[,] coeffs = newdouble[dt.Rows.Count/2-1,3];

int g = 0;

for (int i = 0; i < dt.Rows.Count; i++)

 {

if ((int)dt.Rows[i][0] != 5 && (int)dt.Rows[i][1] == 2)

 {

try

 {

 coeffs[g, 0] = Convert.ToDouble(dt.Rows[i][0]);

 coeffs[g,1] = Convert.ToDouble(dt.Rows[i][2]);

 }

catch (Exception ex)

 {

 }

 }

if ((int)dt.Rows[i][0] != 5 && (int)dt.Rows[i][1] == 3)

 {

 coeffs[g, 2] = Convert.ToDouble(dt.Rows[i][2]);

 g++;

 }

 }

for (int i = 0; i < coeffs.GetLength(0); i++)

 {

 mcommand.Parameters.Clear();

 mcommand.CommandText = "INSERT INTO sppr.effectiveness VALUES ('1','"+coeffs[i,0].ToString()+"','1','0','" + (transSum \* coeffs[i, 1] \* (1 - col / 100) + sleepSum \* coeffs[i, 2] \* (1 - col / 100) - activeSum \* (col / 100)).ToString() + "','01.09.07')";

try

 {

 mcommand.Connection.Open();

 mcommand.EndExecuteNonQuery(mcommand.BeginExecuteNonQuery());

 }

finally

 {

 mcommand.Connection.Close();

 }

 }

 }

privatevoid button2\_Click(object sender, EventArgs e)

 {

Form2 f = newForm2();

 f.Show();

 }

privatevoid button3\_Click(object sender, EventArgs e)

 {

Form3 f = newForm3();

 f.Show();

 }

 }

}

namespace DSS

{

publicpartialclassForm2 : Form

 {

string conString = "User Id=root;password=lbvf;Persist Security Info=True;server=localhost;database=sppr";

public Form2()

 {

 InitializeComponent();

 button1\_Click(this, null);

 }

privatevoid button1\_Click(object sender, EventArgs e)

 {

 MySql.Data.MySqlClient.MySqlCommand mcommand = new MySql.Data.MySqlClient.MySqlCommand();

 mcommand.Connection = new MySql.Data.MySqlClient.MySqlConnection(conString);

DataTable dt = newDataTable();

 mcommand.CommandText = "SELECT LoyProParam as 'Parameter', sppr.coefficients.idClientGroup as 'Client Group ID', ClGroupName as 'Client Group Name', Coefficient FROM sppr.coefficients INNER JOIN sppr.clientgroups ON sppr.coefficients.idClientGroup = sppr.clientgroups.idClientGroup WHERE LoyProType = '1' AND idProdGroup = '1' AND idSeason = '0'";

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 dt.Load(mreader);

 }

finally

 {

 mcommand.Connection.Close();

 }

 dataGridView1.DataSource = dt;

 }

 }

}

namespace DSS

{

publicpartialclassForm3 : Form

 {

string conString = "User Id=root;password=lbvf;Persist Security Info=True;server=localhost;database=sppr";

public Form3()

 {

 InitializeComponent();

 button1\_Click(this, null);

 }

privatevoid button1\_Click(object sender, EventArgs e)

 {

 MySql.Data.MySqlClient.MySqlCommand mcommand = new MySql.Data.MySqlClient.MySqlCommand();

 mcommand.Connection = new MySql.Data.MySqlClient.MySqlConnection(conString);

DataTable dt = newDataTable();

 mcommand.CommandText = "SELECT LoyProParam as 'Parameter', Effect as 'Effectiveness', DateOfCalc as 'Calculation' FROM sppr.effectiveness";

try

 {

 mcommand.Connection.Open();

 MySql.Data.MySqlClient.MySqlDataReader mreader = mcommand.ExecuteReader();

if (mreader.HasRows)

 dt.Load(mreader);

 }

finally

 {

 mcommand.Connection.Close();

 }

 dataGridView1.DataSource = dt;

}

}

}