

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Южно-Уральский государственный университет»
(Национальный исследовательский университет)
Факультет вычислительной математики и информатики
Кафедра экономико-математических методов и статистики

Реализация технического индикатора: индекс денежного потока

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ
по дисциплине «Современные компьютерные технологии»
ЮУрГУ-010400.68.2017.049.001 КР

Руководитель,
_____ А.К. Богушев
« » _____ 2016 г.

Автор проекта
студент группы ВМИ-113
_____ В.А. Безбородов
« » _____ 2016 г.

Проект защищен
с оценкой

« » _____ 2016 г.

Челябинск, 2016

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Южно-Уральский государственный университет»
(Национальный исследовательский университет)
Факультет вычислительной математики и информатики
Кафедра экономико-математических методов и статистики

УТВЕРЖДАЮ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Панюков А.В.
« » _____ 2016 г.

З А Д А Н И Е

на курсовую работу студента

Безбородова Вячеслава Александровича

Группа ВМИ-113

1. Тема работы: Реализация технического индикатора: индекс денежного потока.
2. Срок сдачи студентом законченной работы « » _____ 2016 г.
3. Исходные данные к работе
 - 3.1. Методическое пособие по техническим индикаторам рынка [3];
 - 3.2. Издательская система компьютерной верстки L^AT_EX.
4. Перечень вопросов, подлежащих разработке
 - 4.1. Изучение принципов работы с языком программирования Python;
 - 4.2. Разработка консольного приложения, считывающего данные котировок акций некоторой компании из файла и выводящего среднее значение для цен открытия, закрытия, максимума и минимума;

- 4.3. Добавление в приложение поддержки сети;
- 4.4. Расчет технического индикатора "индекс денежного потока".
5. Перечень графического материала
6. Календарный план

Наименование этапов дипломной работы	Срок выполнения этапов работы	Отметка о выполнении
1. Сбор материалов и литературы по теме курсовой работы	10.09.2015 г.	
2. Изучение принципов работы с системой L ^A T _E X	25.09.2015 г.	
3. Изучение базовых принципов программирования на Python	03.10.2015 г.	
4. Разработка консольного приложения	18.10.2015 г.	
5. Добавление поддержки сети	27.10.2015 г.	
6. Подготовка пояснительной записки курсовой работы	09.11.2015 г.	
Написание главы 1	13.11.2015 г.	
Написание главы 2	17.11.2015 г.	
Написание главы 3	22.11.2015 г.	
7. Оформление пояснительной записки	02.12.2015 г.	
8. Получение отзыва руководителя	05.12.2015 г.	
9. Проверка работы руководителем, исправление замечаний	11.12.2015 г.	
10. Подготовка графического материала и доклада	16.12.2015 г.	
11. Защита курсовой работы	08.06.2016 г.	

7. Дата выдачи задания « » _____ 2016 г.

Заведующий кафедрой _____/Панюков А.В./

Руководитель работы _____/А.К. Богушев/

Студент _____/В.А. Безбородов/

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Южно-Уральский государственный университет»
(Национальный исследовательский университет)
Факультет вычислительной математики и информатики
Кафедра экономико-математических методов и статистики

АННОТАЦИЯ

Безбородов, В.А. Реализация технического индикатора: индекс денежного потока / В.А. Безбородов – Челябинск: ЮУрГУ, Факультет вычислительной математики и информатики, 2016 – 28 с., 1 прил., библиогр. список – 3 названия.

В курсовой работе дается краткое введение в систему технических индикаторов рынка (ТИР), их видов, классификации и исторического развития. Дается подробное описание алгоритма построения одного из ТИР – индекса денежного потока для анализа текущих трендов. Приводится программная реализация индекса денежного потока с описанием и комментариями исходного кода.

В приложении приведен полный исходный код приложения для расчета индекса денежного потока.

ОГЛАВЛЕНИЕ

Введение	6
1 Технические индикаторы рынка	8
2 Индекс денежного потока	14
3 Реализация	16
Заключение	22
ПРИЛОЖЕНИЕ А . Исходный код приложения	24
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	28

Введение

Самые успешные трейдеры и инвесторы во всем мире используют технические индикаторы рынка. Эта малая часть наиболее проницательных участников рынка добивается результатов, значительно превосходящих среднерыночные, и с каждым годом увеличивает размер принадлежащих им активов. Самым надежным способом присоединиться к этой категории финансистов является использование наиболее трезвого и реалистического подхода к инвестированию, основанного на объективном анализе поведения рынка, зафиксированного в исторических данных о торгах.

Индикаторы были разработаны рыночными профессионалами за несколько последних десятилетий путем тщательного ежедневного анализа поведения рыночных цен. Технические индикаторы предназначены для того, чтобы сделать сложный процесс принятия инвестиционных решений относительно простым, прозрачным и эффективным.

Целями работы являются:

- 1) изучить язык и принципы работы в системе компьютерной верстки \LaTeX ;
- 2) изучить базовые принципы программирования на Python;
- 3) разработать консольное приложение, считывающее данные котировок некоторой акции из файла или по сети и вычисляющее индекс денежного потока.

В соответствии с поставленными целями в работе решаются следующие **задачи**:

- проектирование интерфейса и архитектуры приложения;
- корректная обработка исключительных ситуаций;
- полная и информативная система логирования хода работы.

Работа состоит из введения, 3 глав, заключения, 1 приложения и списка литературы. Объем работы составляет 28 страниц. Список литературы содержит 3 наименования.

В первой главе рассматриваются технические индикаторы рынка, их историческое развитие, а также плюсы и минусы использования.

Во второй главе подробно описывается схема построения индекса денежного потока для анализа поведения трендов на исторических данных.

В третьей главе приводятся технические особенности реализации приложения, а также обсуждаются наиболее сложные случаи и уязвимые места.

В заключении перечислены основные результаты работы.

1 Технические индикаторы рынка

Изменение рыночных цен происходит в форме тренда. Характер тренда определяется многими переменными. На движение цен влияют перемены, происходящие на уровне соответствующих фундаментальных экономических факторов. Таким образом, в момент возникновения тренда причина его появления не всегда очевидна. Определить обстоятельства, способствовавшие рождению конкретного тренда, не всегда удастся даже по прошествии значительного временного периода. Технические индикаторы рынка – это инструмент идентификации тренда и изменений в нем; причины формирования тренда с помощью данного метода описаны быть не могут.

Тренды имеют обыкновение длиться. Время их жизни не может быть определено заранее. Однако с уверенностью можно сказать, что о скором окончании тренда свидетельствует нарушение баланса в системе спроса и предложения на соответствующий финансовый инструмент. Любой инвестор заинтересован в том, чтобы первым узнать о грядущих изменениях в трендах.

Ценные бумаги продаются и покупаются людьми, а человек по природе своей эмоционален. Факторами, наиболее важными для принятия инвестиционных решений и определяющими поведение участников рынка, помимо фундаментальных экономических обстоятельств, являются особенности массовой психологии. Эмоциональные характеристики инвесторов учтены при описании ленточных индикаторов и индикаторов настроения.

Тренды действуют на различных временных отрезках, их длина варьируется от нескольких лет до нескольких минут. Направления трендов также различны: тренды движутся либо вверх, либо вниз, либо в стороны. Тренды, таким образом, принято подразделять на:

- основные догосрочные, длящиеся годами;
- значительные средней протяженности, продолжительностью от нескольких недель до нескольких месяцев;
- малые, длящиеся несколько дней;
- моментальные, чье влияние значимо только для внутрисдневных трейдеров.

Установление наилучшего соотношения вознаграждение/риск в условиях трендов различных направлений и разных периодов длительности возможно в случае верного определения подходящих для конкретной ситуации параметров технического индикатора рынка.

Доводы в пользу технических индикаторов рынка (ТИР).

- 1) Принимая к использованию или отклоняя конкретные ТИР, инвестор руководствуется логическими рассуждениями, доводами здравого смысла, а также данными относительно практической действенности индикатора, основанными на результатах его эффективности в прошлом.
- 2) Изучение ТИР может проводиться на научной систематической основе.
- 3) Технические индикаторы являются удобным инструментом организации информации о текущем поведении рынка.
- 4) ТИР позволяют принимать торговые решения, обоснованные историческими прецедентами.
- 5) ТИР помогают сберечь драгоценное время. Мы не обязаны тратить десятилетия на наблюдения за рынком, однако имеем возможность выгодно использовать исторические данные. Эффективное тестирование и выбор индикаторов – наиболее легкий, быстрый и дешевый способ изучения исторических данных.
- 6) ТИР могут быть модифицированы для использования в тренде любого возможного направления – в боковом, растущем или падающем.
- 7) ТИР могут использоваться для выявления трендов в любом временном отрезке. Фрактальная природа рынка заставляет тренды разворачиваться одинаково в любом временном масштабе. Технические индикаторы, таким образом, равно пригодны для изучения: доминирующих основных трендов, длящихся годами; промежуточных движений, характерных для нескольких месяцев или недель, и тех моментальных флуктуаций, с которыми приходится иметь дело трейдеру, совершающему краткосрочные операции в течение дня.
- 8) ТИР пригодны для всех типов финансовых инструментов, торгуемых на открытом рынке: от акций, фьючерсов и товаров до обменных курсов валют и процентных ставок.

- 9) ТИР могут быть использованы для своевременного обнаружения изменения общей экономической тенденции. Финансовые рынки в первую очередь реагируют на перемены в экономике, а технические индикаторы помогают нам "держать руку на пульсе"рынка. Заметим, что большинство инвесторов, руководствующихся фундаментальными факторами и текущими новостями, как правило, с запозданием распознают новые тенденции.
- 10) ТИР позволяют принимать решения, не испытывая чувства неуверенности, волнения, стресса, не тратя сил на пустые догадки и сглаживание противоречий: все свои действия инвестор может заранее четко определить и подвергнуть проверке.
- 11) Правильное восприятие развивающихся рыночных трендов бывает затруднено свойственными инвесторам опасениями, неуверенностью в собственной правоте, увлечением ложными прогнозами, разного рода предубеждениями, надеждами и личными устремлениями. ТИР – источник объективных сигналов: руководствуясь ими, инвестор в состоянии действовать бесстрастно, с минимальными затратами эмоциональной и умственной энергии, повышая тем самым свои шансы на успех.
- 12) Выверенные и тщательно сформулированные правила использования ТИР служат источником торговых сигналов, позволяющих инвесторам уверенно совершать сделку. Уверенность зиждется на результатах тестирования, подтверждающих на историческом материале возможность и в будущем получать с помощью индикатора наилучшее соотношение вознаграждение/риск.
- 13) Будущее не является точным отображением прошлого. Тем не менее естественно предположить, что поведение рынка в будущем в основном повторит неметившиеся в прошлом модели. Следуя этому утверждению, инвестор может выбрать из возможных параметров технических индикаторов те, что наилучшим образом зарекомендовали себя в прошлые периоды.
- 14) По сравнению с прочими методами принятия решений метод использования ТИР отличается большей гибкостью и адаптивностью. Техни-

ческий аналитик без труда использует данные, не имеющие непосредственного отношения к рынкам, например, информацию о настроениях участников рынка, цикличность, межрыночные корреляции, а также фундаментальные макроэкономические данные, такие как уровень инфляции или объем денежной массы. Инструменты технического анализа в состоянии выявить тренды и изменения в них на материале данных любого рода.

- 15) Применение ТИР сопряжено с меньшими временными и умственными затратами, нежели использование прочих методов. В подавляющем большинстве инвестиционные стратегии бывают неполно описаны, чрезмерно усложнены и изобилуют многочисленными и малопонятными переменными. Многие сложные системы инвестор не в состоянии как освоить, так и подвергнуть проверке. ТИР предлагают, как правило, формулы, отличающиеся простотой, разумностью, точностью, кажущиеся очевидными и основанные на обозримом количестве переменных. Такого рода качества позволяют оперировать техническими индикаторами рынка четко и быстро, то есть определяют стиль работы, овладеть которым необходимо каждому желающему преуспеть на финансовом рынке.
- 16) Анализ, построенный на использовании ТИР, неизменно полезен для инвестора. Даже в том случае, когда результаты применения индикатора не оправдывают ожиданий, участник рынка оказывается в определенном выигрыше, поскольку отныне может отбросить неудачный индикатор и обратить внимание на другие. В то же время, столкнувшись с "лживым" индикатором, инвестор всегда может попробовать применить стратегию обратной трактовки сигналов, то есть покупать при сигнале к продаже и продавать при сигнале к покупке.
- 17) Традиционные ТИР можно легко модифицировать с использованием самых современных и сложных математических и статистических методов.
- 18) Сохранение капитала является первой задачей всякой разумной инвестиционной стратегии. Вероятность больших потерь существенно сни-

жается в случае систематического применения прошедших предварительное тестирование ТИР. Исследование исторических данных позволяет собрать наиболее точные сведения о степени возможного риска. Снижение риска, в свою очередь, увеличивает вероятность стабильного получения высоких доходов.

- 19) Наилучшая из возможных торговых стратегий – использование проверенных на историческом материале ТИР, подкрепленные мерами по постоянному контролю риска с целью ограничения убытков в случае возникновения непредвиденных обстоятельств. Альтернативные методы (попытки составления точных прогнозов, реакция на последние события, подражание признанным "гуру"рынка) не в состоянии принести стабильных положительных результатов.
- 20) В наши дни доступ к ТИР имеет каждый инвестор. Благодаря техническим новшествам, исторические данные, необходимые для проведения независимых исследований, легко как получить, так и обработать.
- 21) ТИР становятся одним из самых популярных инструментов принятия решений в среде хорошо информированных участников рынка. Большинство наиболее успешных инвесторов и трейдеров работают сегодня с ТИР.
- 22) ТИР, которыми пользуются самые успешные трейдеры, доступны сегодня любому участнику рынка.

Одно из неоспоримых достоинств работы с ТИР – возможность провести их тестирование на историческом материале. Проверка эффективности стратегии на основе данных прошлых лет – метод, применяемый многими современными инвесторами. Результатом проверок становится набор торговых правил, позволяющих успешно использовать тренды, быстро ограничивать убытки и заранее готовиться к неблагоприятным периодам.

Метод тестирования ТИР на историческом материале обязан своим успехом одному из свойств рыночных поведенческих моделей: с течением времени они меняются несущественно. По традиции большинство инвесторов принимает решения исходя из собственной интуиции и "народной мудрости" представленной в многочисленных популярных руководствах. К сожа-

нию, субъективная оценка широко известных всему сообществу инвесторов фактов делает такого рода решения малоэффективными и не позволяющими получить даже средний результат. Процесс разработки стратегии, базирующейся на случайном подборе фактов, полученных из разнородных источников информации, оказывается непомерно трудоемким. Инвестор не в состоянии ни подвергнуть анализу, ни даже выявить погрешности своего метода, а следовательно, лишен возможности учиться на собственных ошибках. Подобная ситуация может устраивать участника рынка в периоды, когда экономика переживает подъем и прибыли велики, но в тяжелые времена – а они неизменно нас настигают – небрежно составленная, непроверенная стратегия становится серьезной помехой в работе.

Тестирование ТИР на историческом материале – практический метод, позволяющий в любых условиях добиться хорошего результата. Проверенная, объективная и тщательно сформулированная стратегия исключает субъективизм и шарлатанство, не оставляя места догадкам и сомнениям. Работать с ней – значит исполнять ряд четких инструкций, дающих возможность контролировать риск, получая максимально высокую прибыль. Кроме того, проверив свои разработки на историческом материале, относящемся ко многим рыночным циклам, инвестор может получить модель для успешной торговли на любых типах рынков.

2 Индекс денежного потока

Индекс денежного потока (MFI от англ. money flow index) — технический индикатор, призванный показать интенсивность, с которой деньги вкладываются в ценную бумагу и выводятся из неё, анализируя объёмы торгов и соотношения типичных цен периодов [1].

В качестве ключевого ценового показателя для индекса денежного потока используется типичная цена (англ. typical price), которая вычисляется по следующей формуле [3]:

$$\text{TypicalPrice}_t = \frac{\text{high}_t + \text{low}_t + \text{close}_t}{3},$$

где TypicalPrice_t — типичная цена, high_t — наибольшая цена, low_t — наименьшая цена, close_t — цена закрытия рассматриваемого периода t .

Денежный поток (англ. money flow) в каждом периоде вычисляется как произведение типичной цены на объём торгов в этом периоде:

$$\text{MoneyFlow}_t = \text{TypicalPrice}_t \cdot \text{volume}_t,$$

где MoneyFlow_t — денежный поток, TypicalPrice_t — типичная цена, volume_t — объём торгов.

На основе денежного потока вычисляются положительный и отрицательный денежные потоки:

$$\text{PositiveMoneyFlow}_t = \text{MoneyFlow}_t, \text{ if } \text{TypicalPrice}_t > \text{TypicalPrice}_{t-1},$$

$$\text{NegativeMoneyFlow}_t = \text{MoneyFlow}_t, \text{ if } \text{TypicalPrice}_t < \text{TypicalPrice}_{t-1},$$

где $\text{PositiveMoneyFlow}_t$ и $\text{NegativeMoneyFlow}_t$ — положительный и отрицательный денежные потоки.

Денежное отношение (англ. money ratio) в приложении к индексу MFI равно отношению сумм положительных и отрицательных денежных потоков

за выбранный промежуток времени:

$$\text{MoneyRatio}_{t,n} = \frac{\sum_{i=0}^{n-1} \text{PositiveMoneyFlow}_{t-i}}{\sum_{i=0}^{n-1} \text{NegativeMoneyFlow}_{t-i}},$$

где $\text{MoneyRatio}_{t,n}$ – денежное отношение в периоде t , построенное по n предыдущим периодам.

Индекс денежного потока приводит денежное отношение к интервалу $[0; 100]$:

$$\begin{aligned} \text{MFI}_{t,n} &= 100 - \frac{100}{1 + \text{MoneyRatio}_{t,n}} = \\ &= 100 \cdot \frac{\sum_{i=0}^{n-1} \text{PositiveMoneyFlow}_{t-i}}{\sum_{i=0}^{n-1} \text{PositiveMoneyFlow}_{t-i} + \sum_{i=0}^{n-1} \text{NegativeMoneyFlow}_{t-i}}, \end{aligned}$$

где $\text{MFI}_{t,n}$ – значение индекса денежного потока в периоде t , построенное по n предыдущим периодам.

Индекс денежного потока является осциллятором в интервале $[0; 100]$. Нижние его значения указывают на перепроданность рынка, верхние – на перекупленность. Все торговые стратегии, применимые к осцилляторам, могут быть использованы и в отношении MFI. Например [1]:

- Купить, когда MFI опускается ниже 20;
- Продать, когда MFI превышает 80.

3 Реализация

Рассмотрим программную реализацию технического индикатора. В качестве языка программирования был выбран Python.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализации интерпретаторов для JVM (с возможностью компиляции), MSIL (с возможностью компиляции), LLVM и других. Проект PyPy предлагает реализацию Python на самом Python, что уменьшает затраты на изменения языка и постановку экспериментов над новыми возможностями.

Python – активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие официальные стандарты, их роль выполняет CPython.

Как и в большинстве случаев, управляющий поток начинает выполнение со стандартной конструкции с выводом строки приветствия.

```
90 || if __name__ == '__main__':
```



```
91 | print('Data processor 0.2')
```

Далее, подключаются необходимые для работы модули:

- **sys** – модуль предоставляет доступ к некоторым переменным и функциям, используемым интерпретатором;
- **argparse** – используется для написания дружественных интерфейсов командной строки;
- **logging** – модуль определяет функции и классы, реализующие расширяемую систему логирования для приложений и библиотек.

```
93 | import sys
94 | import argparse
95 | import logging
```

Производится настройка аргументов командной строки для конфигурирования параметров приложения.

```
97 |     argparser = argparse.ArgumentParser()
98 |     argparser.add_argument("--symbol", help="stock symbol")
99 |     argparser.add_argument("--file", help="file to process by program")
100 |     argparser.add_argument("--out", help="output file")
101 |     argparser.add_argument("--logfile", help="log file")
102 |     argparser.add_argument("--log", help="log level")
103 |     argparser.add_argument("--year", help="year")
104 |     argparser.add_argument("--mfi", help="money flow count")
105 |     argparser.add_argument("--level", help="mfi level")
106 |     args = argparser.parse_args()
```

Также настраивается система логирования для вывода сообщений в файл.

```
111 |         if args.log:
112 |             numeric_level = getattr(logging, args.log.upper(), None)
113 |
114 |             if not isinstance(numeric_level, int):
115 |                 raise ValueError('Invalid log level: %s' % args.log)
116 |
117 |             logging.basicConfig(filename=args.logfile, level=numeric_level)
```

При необходимости, вывод перенаправляется либо в файл, либо в стандартный поток вывода.

```
119 |         if args.out:
120 |             ostream = open(args.out, 'w')
121 |         else:
122 |             ostream = sys.stdout
```

Происходит конфигурирование периода, за который необходимо получить информацию.

```
124 |         if args.year:
125 |             args.year = int(args.year)
126 |             from datetime import MAXYEAR
127 |             if args.year > MAXYEAR:
128 |                 args.year = MAXYEAR
129 |         else:
130 |             args.year = 1900
```

На этом этапе логика работы приложения делится. Либо данные читаются из файла, если в параметрах командной строки был задан путь до файла. Либо данные скачиваются из сети Интернет, если был задан адрес.

```
134 |         if args.file:
135 |             from os import path
136 |             uri = 'file://' + path.abspath(args.file)
137 |             dateFormat = '%Y-%m-%d'
138 |         elif args.symbol:
139 |             from datetime import date
140 |             from urllib2 import quote
141 |             start = date(args.year, 1, 1)
142 |             end = date(args.year, 12, 31)
143 |             uri = 'http://www.google.com/finance/historical?' +
144 |                 'q={0}&startdate={1}&enddate={2}&output=csv'
145 |             uri = uri.format(args.symbol.upper(),
146 |                             quote(start.strftime('%b %d, %Y')),
147 |                             quote(end.strftime('%b %d, %Y')))
148 |             dateFormat = '%d-%b-%y'
```

Происходит загрузка данных из источника, вывод результатов обработки в приемник и выдача стратегии MFI по требованию.

```
150 |         if uri:
151 |             data = load_data(uri, dateFormat)
152 |
153 |             from json import dump
154 |             dump(process_data(data, args.year), ostream)
155 |
156 |             if args.mfi:
157 |                 dump(get_mfi_strategy(data), ostream)
```

Все ошибки приложение обрабатывает с помощью механизма исключений с выводом причины в файл.

```
159 |         except IOError as e:
```

```

160 |         logging.error("Can't open source: {0}".format(e))
161 |     except:
162 |         logging.error("Bad thing happened")

```

Рассмотрим подробнее механизм загрузки данных. Функция принимает на вход адрес с данными и выводит информационное сообщение.

```

3 | def load_data(uri, dateFormat):
4 |     logging.info('loading data; uri: {0}'.format(uri))

```

Производится подключение необходимых для работы функций и классов.

- **urllib2** – модуль определяет функции и классы, которые предназначены для открытия веб страниц;
- **csv** – модуль реализует классы для чтения и записи табличных данных в формате CSV;
- **time** – модуль для работы с датой и временем.

```

6 |     from urllib2 import urlopen
7 |     from csv import DictReader
18 |    from time import strptime

```

Происходит открытие указанного адреса и чтение данных.

```

9 |     reader = DictReader(urlopen(uri).readlines())

```

Данные читаются построчно, формируя заданную структуру ответа, которая возвращается как результат работы функции.

```

20 |     for row in reader:
21 |         data.append({
22 |             'date': strptime(row['Date'], dateFormat),
23 |             'open': float(row['Open']),
24 |             'close': float(row['Close']),
25 |             'high': float(row['High']),
26 |             'low': float(row['Low']),
27 |             'volume': float(row['Volume'])
28 |         })
29 |
30 |     return data

```

Функция обработки данных принимает на вход сформированные данные и период обработки.

```

73 | def process_data(data, year):
74 |     logging.info('processing data; year: {0}'.format(year))

```

Происходит подсчет средних значений показателей за рассматриваемый период, и результат обработки возвращается.

```
75 |     res = {'open': 0, 'close': 0, 'high': 0, 'low': 0}
76 |
77 |     for row in data:
78 |         if row['date'].tm_year == year:
79 |             res['open'] += row['open']
80 |             res['close'] += row['close']
81 |             res['high'] += row['high']
82 |             res['low'] += row['low']
83 |
84 |     if len(data) != 0:
85 |         for (key, value) in res.items():
86 |             res[key] /= len(data)
87 |
88 |     return res
```

Функция получения MFI стратегии использует показание технического индикатора для формирования своего результата.

```
66 | def get_mfi_strategy(data):
67 |     mfi = get_mfi(data, len(data) - 1, len(data))
68 |     if mfi < 20:
69 |         return {'mfi': mfi, 'strategy': 'buy'}
70 |     elif mfi > 80:
71 |         return {'mfi': mfi, 'strategy': 'sell'}
```

Функция расчета индикатора нормирует значение денежного отношения.

```
63 | def get_mfi(data, t, n):
64 |     return 100 - 100 / (1 + get_money_ratio(data, t, n))
```

Денежное отношение – это отношение положительного денежного потока за период к отрицательному денежному потоку за тот же период.

```
60 | def get_money_ratio(data, t, n):
61 |     return get_positive_money_flow_total(data, t, n) /
62 |         get_negative_money_flow_total(data, t, n)
```

Функции получения полного положительного/отрицательного денежного потока используют методы получения денежного потока за период.

```
48 | def get_positive_money_flow_total(data, t, n):
49 |     total = 0
50 |     for i in range(n):
51 |         total += get_positive_money_flow(data, t, i)
52 |     return total
```

```

53 |
54 | def get_negative_money_flow_total(data, t, n):
55 |     total = 0
56 |     for i in range(n):
57 |         total += get_negative_money_flow(data, t, i)
58 |     return total

```

Методы получения знакового денежного потока оперируют понятием типичной цены.

```

38 | def get_positive_money_flow(data, t, i):
39 |     if get_typical_price(data, t) > get_typical_price(data, t - i):
40 |         return get_money_flow(data, t)
41 |     return 0
42 |
43 | def get_negative_money_flow(data, t, i):
44 |     if get_typical_price(data, t) < get_typical_price(data, t - i):
45 |         return get_money_flow(data, t)
46 |     return 0

```

Денежный поток – это произведение типичной цены на объем продаж.

```

35 | def get_money_flow(data, t):
36 |     return get_typical_price(data, t) * data[t]['volume']

```

Типичная цена, в свою очередь, рассчитывается как среднее среди максимальной, минимальной и ценой закрытия.

```

32 | def get_typical_price(data, t):
33 |     return (data[t]['high'] + data[t]['low'] + data[t]['close']) / 3.0

```

Таким образом, мы рассмотрели программные особенности реализации технического индикатора "денежный поток".

Заключение

В работе представлен вариант реализации расчета технического индикатора рынка "индекс денежного потока" на языке программирования Python.

В работе решены следующие **задачи**:

- изучены принципы работы с языком программирования Python;
- разработано консольное приложение, считывающего данные котировок акций некоторой компании из файла и выводящее среднее значение для цен открытия, закрытия, максимума и минимума;
- в приложение добавлена поддержка сети;
- рассчитан технический индикатор "индекс денежного потока".

По результатам работы можно сделать следующие выводы:

- язык программирования Python очень удобен для быстрой разработки небольших прототипов программ, проверки идей;
- индекс денежного потока способен подсказать инвестору, каким образом построить свою стратегию в текущих условиях рынка.

Приложение может быть использовано для расчета индекса денежного потока для любого исторического периода и любого набора данных.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРИЛОЖЕНИЯ

```
1  #!/usr/bin/python
2
3  def load_data(uri, dateFormat):
4      logging.info('loading data; uri: {0}'.format(uri))
5
6      from urllib2 import urlopen
7      from csv import DictReader
8
9      reader = DictReader(urlopen(uri).readlines())
10
11     encodedFieldNames = []
12     for fieldname in reader.fieldnames:
13         encodedFieldNames.append(fieldname.decode("utf-8-sig").encode("utf-8"))
14     reader.fieldnames = encodedFieldNames
15
16     data = []
17
18     from time import.strptime
19
20     for row in reader:
21         data.append({
22             'date':.strptime(row['Date'], dateFormat),
23             'open': float(row['Open']),
24             'close': float(row['Close']),
25             'high': float(row['High']),
26             'low': float(row['Low']),
27             'volume': float(row['Volume'])
28         })
29
30     return data
31
32 def get_typical_price(data, t):
33     return (data[t]['high'] + data[t]['low'] + data[t]['close']) / 3.0
34
35 def get_money_flow(data, t):
36     return get_typical_price(data, t) * data[t]['volume']
37
38 def get_positive_money_flow(data, t, i):
39     if get_typical_price(data, t) > get_typical_price(data, t - i):
40         return get_money_flow(data, t)
41     return 0
42
```



```

43 def get_negative_money_flow(data, t, i):
44     if get_typical_price(data, t) < get_typical_price(data, t - i):
45         return get_money_flow(data, t)
46     return 0
47
48 def get_positive_money_flow_total(data, t, n):
49     total = 0
50     for i in range(n):
51         total += get_positive_money_flow(data, t, i)
52     return total
53
54 def get_negative_money_flow_total(data, t, n):
55     total = 0
56     for i in range(n):
57         total += get_negative_money_flow(data, t, i)
58     return total
59
60 def get_money_ratio(data, t, n):
61     return get_positive_money_flow_total(data, t, n) /
62         get_negative_money_flow_total(data, t, n)
63 def get_mfi(data, t, n):
64     return 100 - 100 / (1 + get_money_ratio(data, t, n))
65
66 def get_mfi_strategy(data):
67     mfi = get_mfi(data, len(data) - 1, len(data))
68     if mfi < 20:
69         return {'mfi': mfi, 'strategy': 'buy'}
70     elif mfi > 80:
71         return {'mfi': mfi, 'strategy': 'sell'}
72
73 def process_data(data, year):
74     logging.info('processing data; year: {0}'.format(year))
75     res = {'open': 0, 'close': 0, 'high': 0, 'low': 0}
76
77     for row in data:
78         if row['date'].tm_year == year:
79             res['open'] += row['open']
80             res['close'] += row['close']
81             res['high'] += row['high']
82             res['low'] += row['low']
83
84     if len(data) != 0:
85         for (key, value) in res.items():
86             res[key] /= len(data)
87
88     return res

```

```

89
90 if __name__ == '__main__':
91     print('Data processor 0.2')
92
93     import sys
94     import argparse
95     import logging
96
97     argparser = argparse.ArgumentParser()
98     argparser.add_argument("--symbol", help="stock symbol")
99     argparser.add_argument("--file", help="file to process by program")
100    argparser.add_argument("--out", help="output file")
101    argparser.add_argument("--logfile", help="log file")
102    argparser.add_argument("--log", help="log level")
103    argparser.add_argument("--year", help="year")
104    argparser.add_argument("--mfi", help="money flow count")
105    argparser.add_argument("--level", help="mfi level")
106    args = argparser.parse_args()
107
108    ostream = sys.stdout
109
110    try:
111        if args.log:
112            numeric_level = getattr(logging, args.log.upper(), None)
113
114            if not isinstance(numeric_level, int):
115                raise ValueError('Invalid log level: %s' % args.log)
116
117            logging.basicConfig(filename=args.logfile, level=numeric_level)
118
119        if args.out:
120            ostream = open(args.out, 'w')
121        else:
122            ostream = sys.stdout
123
124        if args.year:
125            args.year = int(args.year)
126            from datetime import MAXYEAR
127            if args.year > MAXYEAR:
128                args.year = MAXYEAR
129        else:
130            args.year = 1900
131
132        uri = None
133
134        if args.file:

```

```

135         from os import path
136         uri = 'file:/// ' + path.abspath(args.file)
137         dateFormat = '%Y-%m-%d'
138     elif args.symbol:
139         from datetime import date
140         from urllib2 import quote
141         start = date(args.year, 1, 1)
142         end = date(args.year, 12, 31)
143         uri = 'http://www.google.com/finance/historical?' +
144             'q={0}&startdate={1}&enddate={2}&output=csv'
145         uri = uri.format(args.symbol.upper(),
146             quote(start.strftime('%b %d, %Y')),
147             quote(end.strftime('%b %d, %Y')))
148         dateFormat = '%d-%b-%y'
149
150     if uri:
151         data = load_data(uri, dateFormat)
152
153         from json import dump
154         dump(process_data(data, args.year), ostream)
155
156         if args.mfi:
157             dump(get_mfi_strategy(data), ostream)
158
159     except IOError as e:
160         logging.error("Can't open source: {0}".format(e))
161     except:
162         logging.error("Bad thing happened")
163
164     ostream.close()

```

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акелис, С. Б. Денежных потоков индекс (Money Flow Index) // Технический анализ от А до Я. Полный набор инструментов торговли... от "Абсолютного индекса ширины" до "Японских свечей"/ Пер. с англ. М. Волкова, А. Лебедева. / Стивен Б. Акелис. — М.: Диаграмма, 1999. — 376 с.
2. Львовский, С. М. Набор и верстка в системе LATEX / С. М. Львовский. — 2003.
3. Роберт, К. Энциклопедия технических индикаторов рынка. / Колби Роберт. — М.: "Альпина Бизнес Букс 2006. — 837 с.