Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Кафедра «Компьютерные технологии»

П.С. Юдаев, К.Д. Суворов

Система эмуляции биржи

Программирование с явным выделением состояний

Проектная документация

Проект создан в рамках
«Движение за открытую проектную документацию»
http://is.ifmo.ru

Научный руководитель – доктор технических наук, профессор
А.А. Шалыто

Санкт-Петербург
2004

Содержание

«Правы оба: порядок сменит хаос»
Михаил Пухов. Черный шар.
Знание-сила. 1978. №8

Введение

Для алгоритмизации и программирования задач логического управления А.А. Шалыто была предложена SWITCH-технология [1], которая применительно к событийным программам была развита им совместно с Н.И. Туккелем [2]. Подробно ознакомиться с этой технологией и с конкретными примерами ее использования можно на сайтах http://is.ifmo.ru/ и http://www.softcraft.ru.

Указанная технология удобна для управления технологическими объектами. Это связано с тем, что при использовании автоматного подхода, в частности, удается повысить централизацию логики управления в программном коде. Другое достоинство этого подхода состоит в том, что код является изоморфным графу переходов, по которому он строился. Это позволяет для понимания логики работы программ не обращаться к текстам программ, а рассматривать лишь графы переходов.

В настоящей работе делается попытка использовать эту технологию для решения одной из экономических задач – для эмуляции биржи [3-5].

Для реализации выбран язык C#, позволяющий гибко строить объектно-ориентированные программы.

Ввиду развития технологий в области Internet была использована платформа ASP.NET. Поэтому исполняемый код лежит на WEB-сервере (http://work.youpas.ru/EconomicGame/)

1. Постановка задачи

В программе с некоторым приближением (весьма большим) эмулирована работа биржи. Целью работы является получение практических сведений о пользе выбранной (тестируемой) стратегии игры на бирже.

Правила биржи. В течение дня игроки, руководствуясь выбранной в начале игры стратегией, принимают решение о покупке/продаже акций. В программе реализовано четыре стратегии:

- если вчера продали - сегодня покупаем; если вчера ничего не было - сегодня продаем; если вчера покупали - сегодня продаем;
- если вчера акции падали - сегодня покупаем; если вчера акции росли - сегодня продаем;
- если вчера продали - сегодня покупаем; если вчера ничего не было - сегодня покупаем; если вчера покупали - сегодня продаем;
- если вчера акции падали - сегодня продаем; если вчера акции росли - сегодня покупаем.

Игрок может либо продать все свои акции, либо купить их на все свои деньги по вчерашнему курсу, либо ждать следующего хода. Когда все игроки приняли решения, происходит подсчет «спроса-предложения» и акции распределяются между игроками. Когда акции распределены, выполняется пересчет нового курса акции.

Пересчет курса акции. Функция пересчета базируется на «спросе-предложении». Курс акции так же зависит от предыстории.

Основное ограничение программы - всего один вид акций.

Проект назван «Хаос на бирже» после наблюдений за скачками индекса «РТС» (Российская торгово-сырьевая биржа).

2. Описание подхода

1. На основе анализа предметной области выделяется множество состояний и множество событий, оказывающих влияние на переходы между состояниями.

2. Формируется в общем случае система взаимосвязанных автоматов.

3. Для каждого автомата разрабатывается словесное описание, по крайней мере, в форме перечня решаемых задач.

4. Каждое событие описывается в словесной форме. При этом поясняются условия и причины его возникновения, необходимость обработки.

5. Для каждого автомата строится схема связей, а в случае системы автоматов – еще и схема их взаимодействия.

6. Для каждого автомата строится граф переходов.

7. Первоначально пишется шаблон с «заглушками», изоморфный графу переходов, а после проверки его работы заглушки заменяются функциями, реализующими выходные воздействия, например «подсчет курса акции».

3. Диаграмма классов

Рис. 1. Диаграмма классов

4. Класс “CGame” (автомат A0)

4.1. Нумерация и перечень событий

На основе анализа предметной области выделены следующие события:

e0	–	Программа стартовала
e1	–	Биржа стартовала
e2	–	Игрок определен
e3	–	Игрок сделал ход
e4	–	Завершен пересчет курса акции

Эти события являются для автомата входными. В системе так же используется внутреннее событие *e5*, порождаемое автоматом для вызова других автоматов *A1, A2*.

4.2. Нумерация и перечень входных переменных

На основе анализа предметной области выделены следующие события:

x0	–	Последний игрок
x1	–	Последний день
x2	–	Игрок использует первую стратегию
x3	–	Игрок использует вторую стратегию
x4	–	Игрок использует третью стратегию
x5	–	Игрок использует четвертую стратегию
x9	–	Курс акции равен нулю

4.3. Перечень состояний

4.3.1. Состояние «Старт программы» (*0*)

В это состояние автомат переходит сразу после запуска программы.

4.3.2. Состояние «Старт биржи» (*1*)

Вводятся следующие параметры (рис. 2):

- длительность игры (в днях);
- количество игроков с первой стратегией;
- количество игроков со второй стратегией;
- количество игроков с третьей стратегией;
- количество игроков с четвертой стратегией.

Приходя в это состояние, автомат производит чтение данных из входного файла (историю курса акции – используется сайт http://www.rbc.ru для получения истории). Так же “создаются” игроки, которым принудительно (рис.2) присваиваются стратегии в соответствии с перечисленными выше параметрами. Акции и ресурсы распределяются таким образом, что бы суммарное количество акций равнялось суммарному количеству ресурсов при данном курсе акции (последнем известном из истории).

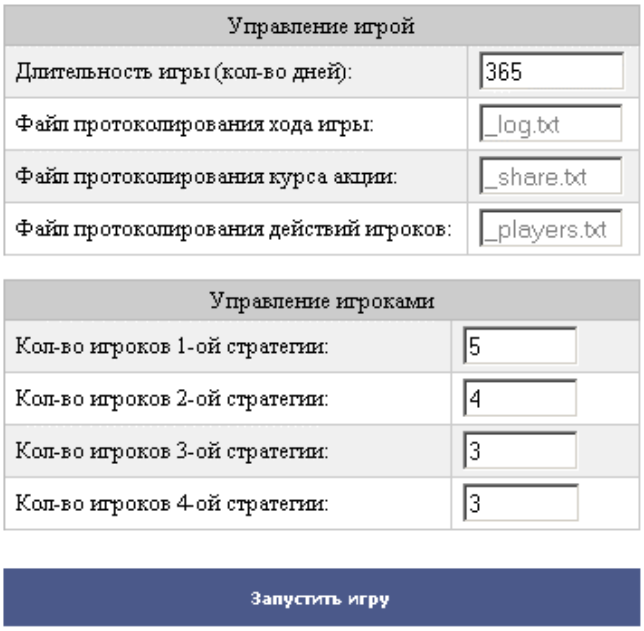

Рис. 2. Интерфейс программы (ввод параметров)

4.3.3. Состояние «Ход игрока» (*2*)

Когда автомат приходит в это состояние порождается действие *z2*. Далее в зависимости от внешних параметров (*x0, x2, x3, x4, x5*) автомат переходит в одно из состояний (*3, 4, 5, 6, 7*).

4.3.4. Состояние «Определение поведения игрока по первой стратегии» (*3*)

В этом состоянии действием *z3* определяется дальнейшее поведение игрока. После этого автомат переходит в состояние *2,* порождая событие *e4.*

4.3.5. Состояние «Определение поведения игрока по второй стратегии» (*4*)

В этом состоянии действием *z4* определяется дальнейшее поведение игрока. Затем автомат переходит в состояние *2,* порождая событие *e4.*

4.3.6. Состояние «Определение поведения игрока по третьей стратегии» (*5*)

В этом состоянии порождается событие e8 для автомата A1.

4.3.7. Состояние «Определение поведения игрока по четвертой стратегии» (*6*)

В этом состоянии порождается событие e8 для автомата A2.

4.3.8. Состояние «Пересчет курса акции» (*7*)

В этом состоянии действием *z5* определяется новый курс акции в соответствии со спросом/предложением. Далее в зависимости от внешних параметров (*x1, x9*) автомат переходит в одно из состояний (2*, 8)*.

4.3.9. Состояние «Завершение программы» (*8*)

В этом состоянии действием *z6* завершается работа автомата. Освобождается память, и программа завершается.

4.4. Нумерация и перечень выходных воздействий

В программе используются следующие выходные воздействия:

- z1 – Запустить биржу
- z2 – Выбрать игрока
- z3 – Определить поведение игрока по первой стратегии
- z4 – Определить поведение игрока по второй стратегии
- z5 – Пересчитать курс акции
- z6 – Выйти из программы

4.4.1. Действие «Установить параметры» (*z1*)

Введенные пользователем параметры из диалога (рис.2) записываются в соответствующие переменные программы.

4.4.2. Действие «Запустить биржу» (*z2*)

Создается список игроков. Им присваиваются стратегии, деньги и акции в соответствии с параметрами, установленными действием *z1*.

4.4.3. Действие «Определить поведение игрока при первой стратегии» (*z3*)

Запускается функция *str1*() из класса *CPlayer*.

4.4.4. Действие «Определить поведение игрока при второй стратегии» (*z4*)

Запускается функция *str2*() из класса *CPlayer*.

4.4.5. Действие «Пересчитать курс акции» (*z5*)

Запускается функция пересчета курса акции.

4.4.6. Действие «Выйти из программы» (*z6*)

Сохранение данных. Освобождение памяти. Завершение программа.

4.5. Схема связей автомата A0

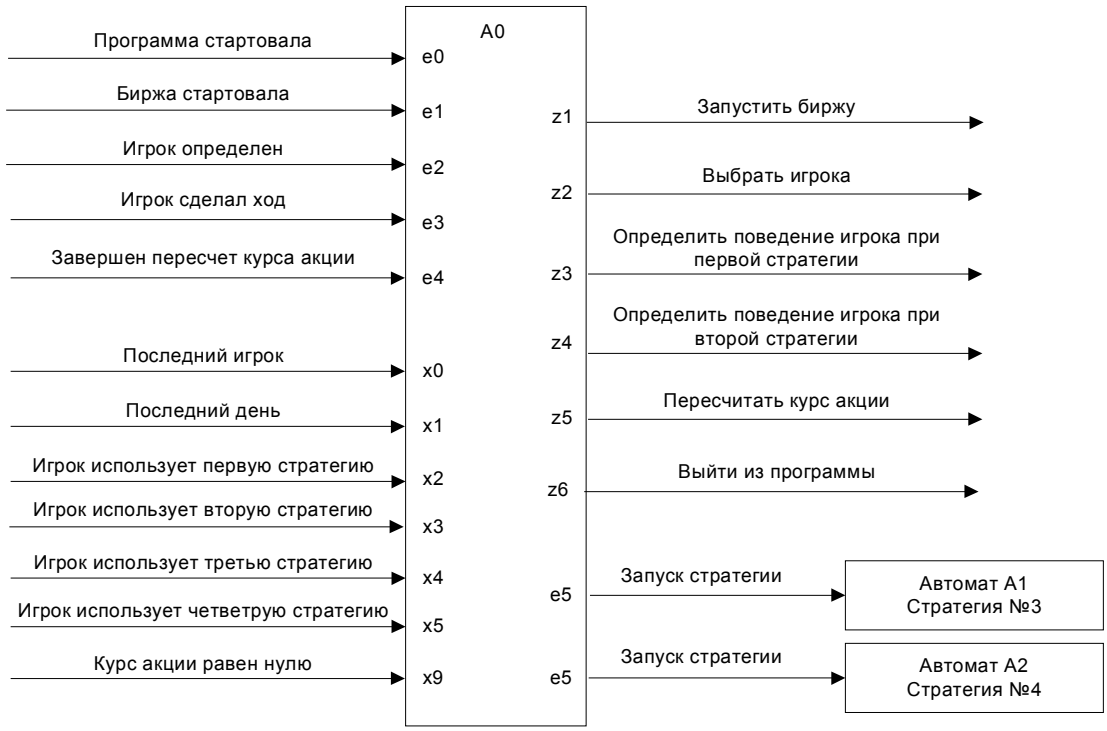

Рис. 3. Схема связей автомата «Управление игрой»

4.6. Граф переходов автомата A0

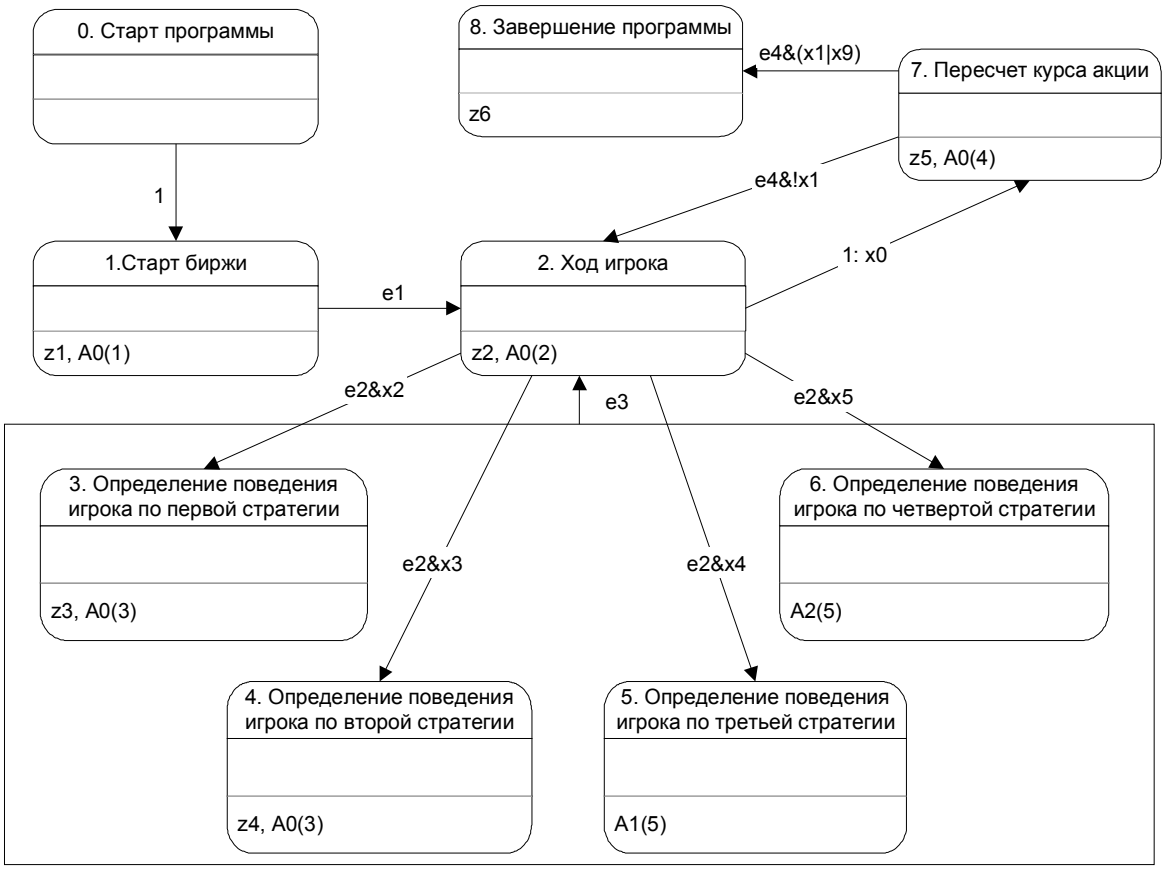

Рис. 4. Граф переходов автомата «Управление игрой»

5. Класс “CShare”

Класс отвечает за обработку курса акции. Он содержит функцию пересчета курса акции в соответствии со спросом/предложением игроков **public void NextIteration().**

Так же в классе хранится история курса акции. Начальные данные следующие: курс акции на начальный день – 11, а на предыдущий – 10.

6. Класс “CPlayer”

6.1. Словесное описание

Класс реализует поведение игрока в зависимости от стратегии. Класс содержит две функции расчета поведения игрока и два автомата:

- **public void str1() – функция работает по следующему алгоритму:** `если вчера акции падали - сегодня продаем; если вчера акции росли - сегодня покупаем.`
- **public void str2() – функция работает по следующему алгоритму:** `если вчера акции падали - сегодня покупаем; если вчера акции росли - сегодня продаем;`
- **public void A1(int e) – автомат работает по следующему алгоритму:** `если вчера продали - сегодня покупаем; если вчера ничего не было - сегодня покупаем, если вчера покупали - сегодня продаем;`
- **public void A2(int e) – автомат работает по следующему алгоритму:** `если вчера продали - сегодня покупаем; если вчера ничего не было - сегодня продаем; если вчера покупали - сегодня продаем;`

6.2. Нумерация и перечень событий

На основе анализа предметной области выделено одно событие:

e5 – Запуск выбранной стратегии

Это событие является для автомата входным. В системе так же используется внутреннее событие *e3*, порождаемое автоматом для вызова автомата *A0*.

6.3. Нумерация и перечень входных переменных

На основе анализа предметной области выделены следующие события:

x6 – Вчера игрок продал акции

x7 – Вчера игрок пропустил ход

x8 – Вчера игрок купил акции

6.4. Схема связей автомата A1

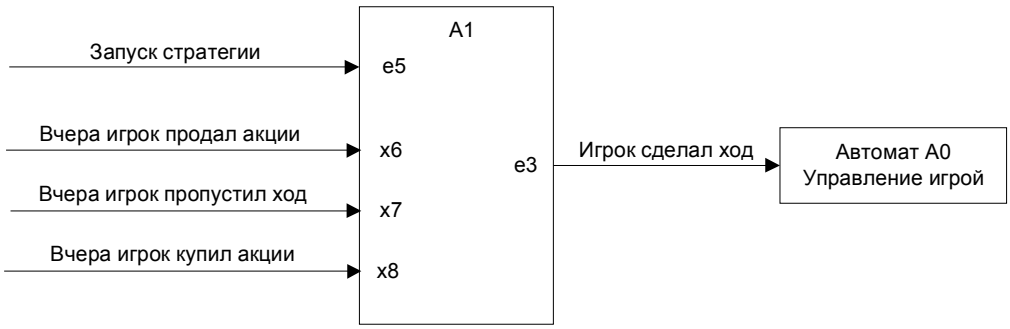

Рис. 5. Схема связей автомата A1

6.5. Граф переходов автомата A1

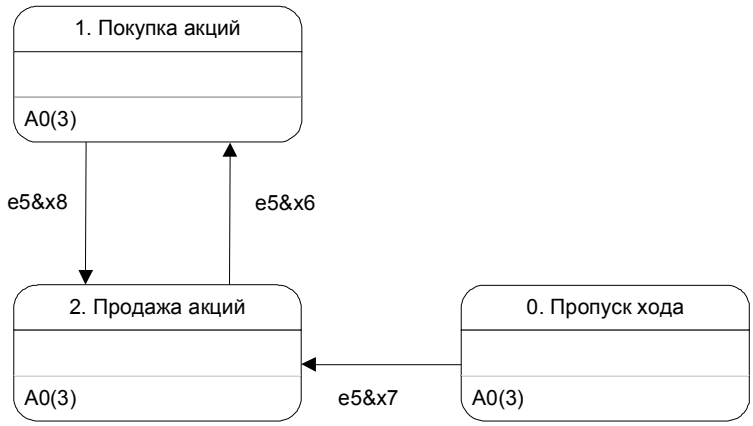

Рис. 6. Граф переходов автомата A1

6.6. Схема связей автомата A2

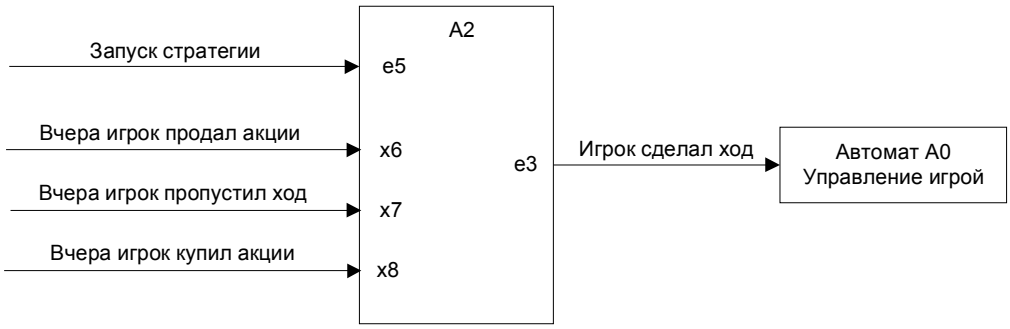

Рис. 7. Схема связей автомата A2

6.7. Граф переходов A2

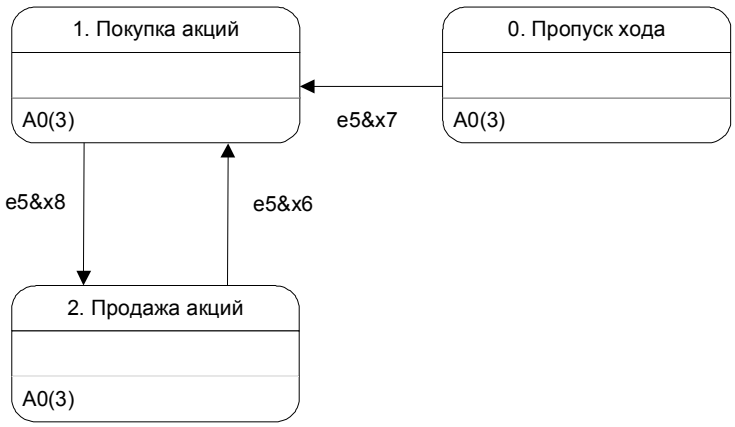

Рис. 8. Граф переходов автомата A2

7. Класс “CLog”

Класс отвечает за протоколирование хода программы. Он содержит несколько функций вывода:

- **LogPlayerTurn** – протоколирует ход игрока;
- **LogAddingPlayer** – протоколирует добавление игрока;
- **LogShareTurn** – протоколирует изменение курса акции;
- **LogGame** – протоколирует общий ход программы.

8. Протоколирование

Для тестирования и отладки программы обеспечен механизм протоколирования. Он позволяет так же продемонстрировать работу программы. Ниже (Приложение 1) приведены три протокола: общий (_log.txt), изменения курса акции (_share.txt) и поведения игроков (_players.txt). Первый протокол является полным, а два других – краткими.

В начале общего протокола выводятся исходные данные о игроках (номер игрока, номер стратегии, количество денег и количество акций). Затем для каждого дня работы биржи выводятся все изменения состояний автоматов и выполняемые при этом выходные воздействия. Так, например, игрок 5, который стартовал в нулевой день с начальными условиями (стратегия 1, деньги – 967, акции – 61) и по своей стратегии решил продать все свои акции, но их всех продать не удалось и у него осталось 23 акции.

- На следующий день он, в соответствии со своей стратегией решил купить акции на все свои деньги (1601), но у него не получилось и у него стало 112 акций и осталось 406 денег.

- На второй день игрок выставил все свои акции на продажу, и у него стало 1595 денег и 28 акций.

- На третий день игрок решил снова купить акции на все деньги, но это опять не получилось и у него стало 464 денег и 112 акций.

- На четвертый он решил продать свои акции, в результате у него стало 1594 денег и 33 акции.
- На пятый день игрок в соответствии со своей стратегией решил купить акции на все деньги. Это ему не удалось и у игрока стало 530 денег и 112 акций.
- На шестой день игрок решил продать свои акции, в результате у него стало 1585 денег и 39 акций.

Заключение

На первый взгляд, кажется, что не имеет смысла использовать автоматное программирование для решения столь простой по логике задачи. Однако, как показал опыт создания программы, в которой логика реализовывалась традиционным путем (на флагах), даже при простом поведении удавалось запутаться и сделать достаточно много ошибок. При использовании автоматов удалось избежать этих проблем. В «заглушках» выводились сообщения о номерах состояний и выходных воздействиях. Созданный код (Приложение 2) сразу заработал правильно. Для подтверждения работы (сертификации) программы по графу переходов были проверены все переходы и все выполняемые действия.

В настоящее время проводится бета-тестирование программы для рассмотрения новых стратегий поведения. При этом автоматный подход должен позволить упростить модификацию программы.

Литература

1. *Шалыто А.А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
2. *Шалыто А.А., Туккель Н.И.* SWITCH-технология - автоматный подход к созданию программного обеспечения "реактивных" систем // Программирование. 2001. № 5.
3. *Первозванский А.А., Первозванская Т.Н.* Финансовый рынок: расчет и риск. М.: ИНФРА-М, 1994.
4. *Буренин А.Н.* Рынок производных финансовых инструментов. М.: Научно-техническое общество им. акад. С.И. Вавилова, 2002.
5. *Фельдман А.Б.* Основы рынка производных ценных бумаг. Учебно-практическое пособие. М.: ИНФРА-М, 1996.

Приложение 1. Протоколы

Протокол _log.txt:
Сообщение от автомата A0: Старт программы (состояние y0=0)
Сообщение от автомата A0: Старт биржи (состояние y0=1)
День 0: Добавляется игрок с номером 0, стратегией 0, деньгами 447,352524551152, акциями 31
День 0: Добавляется игрок с номером 1, стратегией 0, деньгами 129,876539385818, акциями 9
День 0: Добавляется игрок с номером 2, стратегией 0, деньгами 562,798337338546, акциями 39
День 0: Добавляется игрок с номером 3, стратегией 0, деньгами 937,997228897576, акциями 65
День 0: Добавляется игрок с номером 4, стратегией 1, деньгами 490,644704346425, акциями 34
День 0: Добавляется игрок с номером 5, стратегией 1, деньгами 966,858682094425, акциями 67
День 0: Добавляется игрок с номером 6, стратегией 1, деньгами 880,274322503879, акциями 61
День 0: Добавляется игрок с номером 7, стратегией 2, деньгами 0, акциями 0
День 0: Добавляется игрок с номером 8, стратегией 2, деньгами 764,828509716485, акциями 53
День 0: Добавляется игрок с номером 9, стратегией 3, деньгами 1096,73522148024, акциями 76
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 0: Игрок с номером 0 хочет купить акции (в наличии 447,352524551152 денег и 31 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 0: Игрок с номером 1 хочет купить акции (в наличии 129,876539385818 денег и 9 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 0: Игрок с номером 2 хочет купить акции (в наличии 562,798337338546 денег и 39 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 0: Игрок с номером 3 хочет купить акции (в наличии 937,997228897576 денег и 65 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 0: Игрок с номером 4 выставляет на продажу свои акции (в наличии 490,644704346425 денег и 34 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 0: Игрок с номером 5 выставляет на продажу свои акции (в наличии 966,858682094425 денег и 67 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 0: Игрок с номером 6 выставляет на продажу свои акции (в наличии 880,274322503879 денег и 61 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 0: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 0: Игрок с номером 8 выставляет на продажу свои акции (в наличии 764,828509716485 денег и 53 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 0: Завершен подсчет курса акции, спрос = 144 предложение = 215 новая цена = 13,4418686597056
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 1: Игрок с номером 0 выставляет на продажу свои акции (в наличии 0 денег и 62 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 1: Игрок с номером 1 выставляет на продажу свои акции (в наличии 0 денег и 18 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 1: Игрок с номером 2 выставляет на продажу свои акции (в наличии 0 денег и 78 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 1: Игрок с номером 3 выставляет на продажу свои акции (в наличии 43,2921797952728 денег и 127 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 1: Игрок с номером 4 хочет купить акции (в наличии 808,120689511758 денег и 12 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 1: Игрок с номером 5 хочет купить акции (в наличии 1601,81065242509 денег и 23 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 1: Игрок с номером 6 хочет купить акции (в наличии 1457,50338644085 денег и 21 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 1: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 1: Игрок с номером 8 хочет купить акции (в наличии 1269,90394066133 денег и 18 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)

День 1: Завершен подсчет курса акции, спрос = 381 предложение = 285 новая цена = 14,1625893804263
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 2: Игрок с номером 0 хочет купить акции (в наличии 833,395856901746 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 2: Игрок с номером 1 хочет купить акции (в наличии 241,953635874701 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 2: Игрок с номером 2 хочет купить акции (в наличии 1048,46575545704 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 2: Игрок с номером 3 хочет купить акции (в наличии 1736,96763091818 денег и 1 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 2: Игрок с номером 4 выставляет на продажу свои акции (в наличии 216,678468484712 денег и 56 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 2: Игрок с номером 5 выставляет на продажу свои акции (в наличии 405,484341711295 денег и 112 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 2: Игрок с номером 6 выставляет на продажу свои акции (в наличии 368,712025004697 денег и 102 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 2: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 2: Игрок с номером 8 выставляет на продажу свои акции (в наличии 328,973134481943 денег и 88 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 2: Завершен подсчет курса акции, спрос = 271 предложение = 358 новая цена = 13,4710154535583
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 3: Игрок с номером 0 выставляет на продажу свои акции (в наличии 11,9656728370206 денег и 58 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 3: Игрок с номером 1 выставляет на продажу свои акции (в наличии 1,18961640745334 денег и 17 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 3: Игрок с номером 2 выставляет на продажу свои акции (в наличии 0,434141305488993 денег и 74 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 3: Игрок с номером 3 выставляет на продажу свои акции (в наличии 37,4569052670199 денег и 121 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 3: Игрок с номером 4 хочет купить акции (в наличии 811,507222462617 денег и 14 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 3: Игрок с номером 5 хочет купить акции (в наличии 1595,1418496671 денег и 28 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 3: Игрок с номером 6 хочет купить акции (в наличии 1459,23140729752 денег и 25 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 3: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 3: Игрок с номером 8 хочет купить акции (в наличии 1263,70403359008 денег и 22 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 3: Завершен подсчет курса акции, спрос = 379 предложение = 270 новая цена = 14,3107689204304
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 4: Игрок с номером 0 хочет купить акции (в наличии 793,2845691434 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 4: Игрок с номером 1 хочет купить акции (в наличии 230,196879117944 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 4: Игрок с номером 2 хочет купить акции (в наличии 997,2892848688 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 4: Игрок с номером 3 хочет купить акции (в наличии 1653,97875969401 денег и 1 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 4: Игрок с номером 4 выставляет на продажу свои акции (в наличии 245,72457341317 денег и 56 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)

День 4: Игрок с номером 5 выставляет на продажу свои акции (в наличии 463,57655156821 денег и 112 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 4: Игрок с номером 6 выставляет на продажу свои акции (в наличии 421,963217373536 денег и 102 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 4: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 4: Игрок с номером 8 выставляет на продажу свои акции (в наличии 374,617013655234 денег и 88 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 4: Завершен подсчет курса акции, спрос = 255 предложение = 358 новая цена = 13,4706384147207
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 5: Игрок с номером 0 выставляет на продажу свои акции (в наличии 6,19227851972926 денег и 55 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 5: Игрок с номером 1 выставляет на продажу свои акции (в наличии 1,22457639105784 денег и 16 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 5: Игрок с номером 2 выставляет на продажу свои акции (в наличии 9,84622935910465 денег и 69 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 5: Игрок с номером 3 выставляет на продажу свои акции (в наличии 51,1726406058097 денег и 113 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 5: Игрок с номером 4 хочет купить акции (в наличии 803,844561309955 денег и 17 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 5: Игрок с номером 5 хочет купить акции (в наличии 1594,12729628221 денег и 33 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 5: Игрок с номером 6 хочет купить акции (в наличии 1452,33857964452 денег и 30 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 5: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 5: Игрок с номером 8 хочет купить акции (в наличии 1261,88468672192 денег и 26 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 5: Завершен подсчет курса акции, спрос = 377 предложение = 253 новая цена = 14,4547653988477
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 6: Игрок с номером 0 хочет купить акции (в наличии 747,07739132937 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 6: Игрок с номером 1 хочет купить акции (в наличии 216,75479102659 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 6: Игрок с номером 2 хочет купить акции (в наличии 939,320279974836 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 6: Игрок с номером 3 хочет купить акции (в наличии 1573,35478146925 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 6: Игрок с номером 4 выставляет на продажу свои акции (в наличии 265,019024721125 денег и 57 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 6: Игрок с номером 5 выставляет на продажу свои акции (в наличии 529,946861519271 денег и 112 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 6: Игрок с номером 6 выставляет на продажу свои акции (в наличии 482,452613784629 денег и 102 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 6: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 6: Игрок с номером 8 выставляет на продажу свои акции (в наличии 426,705105009231 денег и 88 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 6: Завершен подсчет курса акции, спрос = 237 предложение = 359 новая цена = 13,4312754659618
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 7: Игрок с номером 0 выставляет на продажу свои акции (в наличии 9,88435598813601 денег и 51 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)

День 7: Игрок с номером 1 выставляет на продажу свои акции (в наличии 14,3880754427215 денег и 14 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 7: Игрок с номером 2 выставляет на продажу свои акции (в наличии 14,2152944485814 денег и 64 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по первой стратегии (состояние y0=3)
День 7: Игрок с номером 3 выставляет на продажу свои акции (в наличии 41,1496491913945 денег и 106 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 7: Игрок с номером 4 хочет купить акции (в наличии 799,845344478491 денег и 20 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 7: Игрок с номером 5 хочет купить акции (в наличии 1585,14473563516 денег и 39 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по второй стратегии (состояние y0=4)
День 7: Игрок с номером 6 хочет купить акции (в наличии 1450,92189550743 денег и 35 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 7: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Определение поведения игрока по третьей стратегии (состояние y0=5)
День 7: Игрок с номером 8 хочет купить акции (в наличии 1265,0814981424 денег и 30 акций)
Сообщение от автомата A0: Ход игрока (состояние y0=2)
Сообщение от автомата A0: Подсчет курса акций (состояние y0=7)
День 7: Завершен подсчет курса акции, спрос = 379 предложение = 235 новая цена = 14,6039139024439
Сообщение от автомата A0: Завершение программы (состояние y0=8)

В протоколе, приведенном ниже, фиксируются только спрос, предложение и курс акции для каждого дня.

Протокол _share.txt
День 0: Завершен подсчет курса акции, спрос = 144 предложение = 215 новая цена = 13,4418686597056
День 1: Завершен подсчет курса акции, спрос = 381 предложение = 285 новая цена = 14,1625893804263
День 2: Завершен подсчет курса акции, спрос = 271 предложение = 358 новая цена = 13,4710154535583
День 3: Завершен подсчет курса акции, спрос = 379 предложение = 270 новая цена = 14,3107689204304
День 4: Завершен подсчет курса акции, спрос = 255 предложение = 358 новая цена = 13,4706384147207
День 5: Завершен подсчет курса акции, спрос = 377 предложение = 253 новая цена = 14,4547653988477
День 6: Завершен подсчет курса акции, спрос = 237 предложение = 359 новая цена = 13,4312754659618
День 7: Завершен подсчет курса акции, спрос = 379 предложение = 235 новая цена = 14,6039139024439

В последнем протоколе записываются только действия, связанные с вводом каждого игрока на биржу и его работой.

Протокол _players.txt
День 0: Добавляется игрок с номером 0, стратегией 0, деньгами 447,352524551152, акциями 31
День 0: Добавляется игрок с номером 1, стратегией 0, деньгами 129,876539385818, акциями 9
День 0: Добавляется игрок с номером 2, стратегией 0, деньгами 562,798337338546, акциями 39
День 0: Добавляется игрок с номером 3, стратегией 0, деньгами 937,997228897576, акциями 65
День 0: Добавляется игрок с номером 4, стратегией 1, деньгами 490,644704346425, акциями 34
День 0: Добавляется игрок с номером 5, стратегией 1, деньгами 966,858682094425, акциями 67
День 0: Добавляется игрок с номером 6, стратегией 1, деньгами 880,274322503879, акциями 61
День 0: Добавляется игрок с номером 7, стратегией 2, деньгами 0, акциями 0
День 0: Добавляется игрок с номером 8, стратегией 2, деньгами 764,828509716485, акциями 53
День 0: Добавляется игрок с номером 9, стратегией 3, деньгами 1096,73522148024, акциями 76
День 0: Игрок с номером 0 хочет купить акции (в наличии 447,352524551152 денег и 31 акций)
День 0: Игрок с номером 1 хочет купить акции (в наличии 129,876539385818 денег и 9 акций)
День 0: Игрок с номером 2 хочет купить акции (в наличии 562,798337338546 денег и 39 акций)
День 0: Игрок с номером 3 хочет купить акции (в наличии 937,997228897576 денег и 65 акций)
День 0: Игрок с номером 4 выставляет на продажу свои акции (в наличии 490,644704346425 денег и 34 акций)
День 0: Игрок с номером 5 выставляет на продажу свои акции (в наличии 966,858682094425 денег и 67 акций)
День 0: Игрок с номером 6 выставляет на продажу свои акции (в наличии 880,274322503879 денег и 61 акций)
День 0: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
День 0: Игрок с номером 8 выставляет на продажу свои акции (в наличии 764,828509716485 денег и 53 акций)
День 1: Игрок с номером 0 выставляет на продажу свои акции (в наличии 0 денег и 62 акций)
День 1: Игрок с номером 1 выставляет на продажу свои акции (в наличии 0 денег и 18 акций)
День 1: Игрок с номером 2 выставляет на продажу свои акции (в наличии 0 денег и 78 акций)
День 1: Игрок с номером 3 выставляет на продажу свои акции (в наличии 43,2921797952728 денег и 127 акций)
День 1: Игрок с номером 4 хочет купить акции (в наличии 808,120689511758 денег и 12 акций)
День 1: Игрок с номером 5 хочет купить акции (в наличии 1601,81065242509 денег и 23 акций)
День 1: Игрок с номером 6 хочет купить акции (в наличии 1457,50338644085 денег и 21 акций)
День 1: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
День 1: Игрок с номером 8 хочет купить акции (в наличии 1269,90394066133 денег и 18 акций)
День 2: Игрок с номером 0 хочет купить акции (в наличии 833,395856901746 денег и 0 акций)
День 2: Игрок с номером 1 хочет купить акции (в наличии 241,953635874701 денег и 0 акций)

День 2: Игрок с номером 2 хочет купить акции (в наличии 1048,46575545704 денег и 0 акций)
День 2: Игрок с номером 3 хочет купить акции (в наличии 1736,96763091818 денег и 1 акций)
День 2: Игрок с номером 4 выставляет на продажу свои акции (в наличии 216,678468484712 денег и 56 акций)
День 2: Игрок с номером 5 выставляет на продажу свои акции (в наличии 405,484341711295 денег и 112 акций)
День 2: Игрок с номером 6 выставляет на продажу свои акции (в наличии 368,712025004697 денег и 102 акций)
День 2: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
День 2: Игрок с номером 8 выставляет на продажу свои акции (в наличии 328,973134481943 денег и 88 акций)
День 3: Игрок с номером 0 выставляет на продажу свои акции (в наличии 11,9656728370206 денег и 58 акций)
День 3: Игрок с номером 1 выставляет на продажу свои акции (в наличии 1,18961640745334 денег и 17 акций)
День 3: Игрок с номером 2 выставляет на продажу свои акции (в наличии 0,434141305488993 денег и 74 акций)
День 3: Игрок с номером 3 выставляет на продажу свои акции (в наличии 37,4569052670199 денег и 121 акций)
День 3: Игрок с номером 4 хочет купить акции (в наличии 811,507222462617 денег и 14 акций)
День 3: Игрок с номером 5 хочет купить акции (в наличии 1595,1418496671 денег и 28 акций)
День 3: Игрок с номером 6 хочет купить акции (в наличии 1459,23140729752 денег и 25 акций)
День 3: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
День 3: Игрок с номером 8 хочет купить акции (в наличии 1263,70403359008 денег и 22 акций)
День 4: Игрок с номером 0 хочет купить акции (в наличии 793,2845691434 денег и 0 акций)
День 4: Игрок с номером 1 хочет купить акции (в наличии 230,196879117944 денег и 0 акций)
День 4: Игрок с номером 2 хочет купить акции (в наличии 997,2892848688 денег и 0 акций)
День 4: Игрок с номером 3 хочет купить акции (в наличии 1653,97875969401 денег и 1 акций)
День 4: Игрок с номером 4 выставляет на продажу свои акции (в наличии 245,72457341317 денег и 56 акций)
День 4: Игрок с номером 5 выставляет на продажу свои акции (в наличии 463,57655156821 денег и 112 акций)
День 4: Игрок с номером 6 выставляет на продажу свои акции (в наличии 421,963217373536 денег и 102 акций)
День 4: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
День 4: Игрок с номером 8 выставляет на продажу свои акции (в наличии 374,617013655234 денег и 88 акций)
День 5: Игрок с номером 0 выставляет на продажу свои акции (в наличии 6,19227851972926 денег и 55 акций)
День 5: Игрок с номером 1 выставляет на продажу свои акции (в наличии 1,22457639105784 денег и 16 акций)
День 5: Игрок с номером 2 выставляет на продажу свои акции (в наличии 9,84622935910465 денег и 69 акций)
День 5: Игрок с номером 3 выставляет на продажу свои акции (в наличии 51,1726406058097 денег и 113 акций)
День 5: Игрок с номером 4 хочет купить акции (в наличии 803,844561309955 денег и 17 акций)
День 5: Игрок с номером 5 хочет купить акции (в наличии 1594,12729628221 денег и 33 акций)
День 5: Игрок с номером 6 хочет купить акции (в наличии 1452,33857964452 денег и 30 акций)
День 5: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
День 5: Игрок с номером 8 хочет купить акции (в наличии 1261,88468672192 денег и 26 акций)
День 6: Игрок с номером 0 хочет купить акции (в наличии 747,07739132937 денег и 0 акций)
День 6: Игрок с номером 1 хочет купить акции (в наличии 216,75479102659 денег и 0 акций)
День 6: Игрок с номером 2 хочет купить акции (в наличии 939,320279974836 денег и 0 акций)
День 6: Игрок с номером 3 хочет купить акции (в наличии 1573,35478146925 денег и 0 акций)
День 6: Игрок с номером 4 выставляет на продажу свои акции (в наличии 265,019024721125 денег и 57 акций)
День 6: Игрок с номером 5 выставляет на продажу свои акции (в наличии 529,946861519271 денег и 112 акций)
День 6: Игрок с номером 6 выставляет на продажу свои акции (в наличии 482,452613784629 денег и 102 акций)
День 6: Игрок с номером 7 выставляет на продажу свои акции (в наличии 0 денег и 0 акций)
День 6: Игрок с номером 8 выставляет на продажу свои акции (в наличии 426,705105009231 денег и 88 акций)
День 7: Игрок с номером 0 выставляет на продажу свои акции (в наличии 9,88435598813601 денег и 51 акций)
День 7: Игрок с номером 1 выставляет на продажу свои акции (в наличии 14,3880754427215 денег и 14 акций)
День 7: Игрок с номером 2 выставляет на продажу свои акции (в наличии 14,2152944485814 денег и 64 акций)
День 7: Игрок с номером 3 выставляет на продажу свои акции (в наличии 41,1496491913945 денег и 106 акций)
День 7: Игрок с номером 4 хочет купить акции (в наличии 799,845344478491 денег и 20 акций)
День 7: Игрок с номером 5 хочет купить акции (в наличии 1585,14473563516 денег и 39 акций)
День 7: Игрок с номером 6 хочет купить акции (в наличии 1450,92189550743 денег и 35 акций)
День 7: Игрок с номером 7 хочет купить акции (в наличии 0 денег и 0 акций)
День 7: Игрок с номером 8 хочет купить акции (в наличии 1265,0814981424 денег и 30 акций)

Приложение 2. Листинги программ

index.aspx.cs (класс CGame)

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace EconomicGame
{
    public class CGame : System.Web.UI.Page
    {
        protected System.Web.UI.HtmlControls.HtmlInputText Days;
        protected System.Web.UI.HtmlControls.HtmlInputText FileLog;
        protected System.Web.UI.HtmlControls.HtmlInputText FileShare;
        protected System.Web.UI.HtmlControls.HtmlInputText FilePlayers;
        protected System.Web.UI.HtmlControls.HtmlInputText N1Players;
        protected System.Web.UI.HtmlControls.HtmlInputText N2Players;
        protected System.Web.UI.HtmlControls.HtmlInputText N3Players;
        protected System.Web.UI.WebControls.Button StartGame;
        protected System.Web.UI.HtmlControls.HtmlInputText N4Players;

        public static CLog log;
        public static CShare share = new CShare();
        public static ArrayList players = new ArrayList();
        public static string HomeDir = "c:\\";
        public Random rand = new Random();

        public int y0 = 0;
        public static int iCurrentDay = 0;
        protected System.Web.UI.WebControls.LinkButton LinkButton1;
        public static int iCurrentPlayer = 0;

        private void Page_Load(object sender, System.EventArgs e)
        {
            // Put user code to initialize the page here
        }

        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            InitializeComponent();
            base.OnInit(e);
        }

        private void InitializeComponent()
        {
            this.StartGame.Click += new System.EventHandler(this.StartGame_Click);
            this.Load += new System.EventHandler(this.Page_Load);
        }
        #endregion

        private void StartGame_Click(object sender, System.EventArgs e)
        {
            A0(0);
            Response.Redirect("result.aspx");
        }

        public void A0(int e)
        {
            bool x0 = xx0(); // последний игрок
            bool x1 = xx1(); // последний день
            bool x2 = xx2(); // игрок использует первую стратегию
            bool x3 = xx3(); // игрок использует вторую стратегию
            bool x4 = xx4(); // игрок использует третью стратегию
            bool x5 = xx5(); // игрок использует четветую стратегию
            bool x9 = xx9(); // курс акции равен нулю
```

```
int y0_old = y0;
switch(y0)
{
    case 0:  // ввод параметров
            y0 = 1;
        break;

    case 1:  // старт биржи
        if (e==1)
        {
            y0 = 2;
        }
        break;

    case 2:  // ход игрока
        if (x0)
        {
            y0 = 7;
        }
        else if(e==2&x2)
        {
            y0 = 3;
        }
        else if(e==2&x3)
        {
            y0 = 4;
        }
        else if(e==2&x4)
        {
            y0 = 5;
        }
        else if(e==2&x5)
        {
            y0 = 6;
        }
        break;

    case 3: // определение поведения игрока по первой стратегии
        if (e==3)
        {
            y0 = 2;
        }
        break;
    case 4: // определение поаедения игрока по второй стратегии
        if (e==3)
        {
            y0 = 2;
        }
        break;
    case 5: // определение поаедения игрока по третьей стратегии
        if (e==3)
        {
            y0 = 2;
        }
        break;
    case 6: // определение поаедения игрока по четвертой стратегии
        if (e==3)
        {
            y0 = 2;
        }
        break;
    case 7: // пересчет курса акции
        if (e==4 & x1 | x9)
        {
            y0 = 8;
        }else if (e==4&!x1) {
            y0 = 2;
        }
        break;
    case 8: // завершение программы
        break;
}

int t = y0;
if (y0_old!=y0)
{
    switch(t)
    {
        case 0: // ввод параметров
```

```
                    break;
                case 1:// старт биржи
                    z1();
                    A0(1);
                    break;
                case 2: // ход игрока
                    z2();
                    A0(2);
                    break;
                case 3: // определение поаедения игрока по первой стратегии
                    z3();
                    A0(3);
                    break;
                case 4: // определение поаедения игрока по второй стратегии
                    z4();
                    A0(3);
                    break;
                case 5: // определение поаедения игрока по третьей стратегии
                    A1(5);
                    break;
                case 6: // определение поаедения игрока по четвертой стратегии
                    A2(5);
                    break;
                case 7: // пересчет курса акции
                    z5();
                    A0(4);
                    break;
                case 8: // завершение программы
                    z6();
                    break;
            }
        }
    }

    public void z1()
    {
        HomeDir = Server.MapPath("output//");
        players.Clear();
        log    =    new    CLog(FileLog.Value.ToString(),    FileShare.Value.ToString(),
FilePlayers.Value.ToString());
        log.LogGame(0);
        log.LogGame(1);
        iCurrentDay = 0;
        for(int i=0; i<Convert.ToInt32(N1Players.Value.ToString()); i++)
        {
            AddPlayer(0);
        }
        for(int i=0; i<Convert.ToInt32(N2Players.Value.ToString()); i++)
        {
            AddPlayer(1);
        }
        for(int i=0; i<Convert.ToInt32(N3Players.Value.ToString()); i++)
        {
            AddPlayer(2);
        }
        for(int i=0; i<Convert.ToInt32(N4Players.Value.ToString()); i++)
        {
            AddPlayer(3);
        }
    }

    public void AddPlayer(int iStrategy)
    {
        CPlayer player    = new CPlayer();

        player.iPortfolio = rand.Next()%100;
        player.dMoney     = player.iPortfolio * share.dPrice;
        player.iStrategy  = iStrategy;
        player.iAction    = 0;
        player.iID        = players.Count;

        players.Add(player);
        log.LogAddingPlayer(player.iID.ToString(), player.dMoney.ToString(),
player.iPortfolio.ToString(), player.iStrategy.ToString());
    }

    public void z2()
    {
        log.LogGame(2);
```

```
            //iCurrentPlayer++;
        }

        public void z3()
        {
            CPlayer p = (CPlayer)players[iCurrentPlayer];
            if (p.iStrategy==0)
            {
                log.LogGame(3);
                p.str1();
                log.LogPlayerTurn(p.iID.ToString(), p.iAction.ToString(),
p.dMoney.ToString(), p.iPortfolio.ToString());
                iCurrentPlayer++;
            }
        }

        public void z4()
        {
            CPlayer p = (CPlayer)players[iCurrentPlayer];
            if (p.iStrategy==1)
            {
                log.LogGame(4);
                p.str2();
                log.LogPlayerTurn(p.iID.ToString(), p.iAction.ToString(),
p.dMoney.ToString(), p.iPortfolio.ToString());
                iCurrentPlayer++;
            }
        }

        public void z5()
        {
            log.LogGame(7);
            iCurrentPlayer = 0;
            share.NextIteration();
            iCurrentDay++;
        }

        public void z6()
        {
            log.LogGame(8);
            share.dPrevPrice = 10;
            share.dPrice = 11;
            players.Clear();
            iCurrentDay = 0;
            iCurrentPlayer = 0;
            y0 = 0;
        }

        public void A1(int e)
        {
            CPlayer p = (CPlayer)players[iCurrentPlayer];
            if (p.iStrategy==2)
            {
                log.LogGame(5);
                p.A1(e);
                if (p.y1==0){p.iAction = 0;}
                if (p.y1==1){p.iAction = 1;}
                if (p.y1==2){p.iAction = -1;}
                log.LogPlayerTurn(p.iID.ToString(), p.iAction.ToString(),
p.dMoney.ToString(), p.iPortfolio.ToString());
                iCurrentPlayer++;
            }
            A0(3);
        }

        public void A2(int e)
        {
            CPlayer p = (CPlayer)players[iCurrentPlayer];
            if (p.iStrategy==3)
            {
                log.LogGame(6);
                p.A2(e);
                if (p.y2==0){p.iAction = 0;}
                if (p.y2==1){p.iAction = 1;}
                if (p.y2==2){p.iAction = -1;}
                log.LogPlayerTurn(p.iID.ToString(), p.iAction.ToString(),
p.dMoney.ToString(), p.iPortfolio.ToString());
                iCurrentPlayer++;
            }
```

```
            A0(3);
        }

        public bool xx0()
        {
            return iCurrentPlayer==players.Count-1;
        }

        public bool xx1()
        {
            return iCurrentDay==Convert.ToInt32(Days.Value.ToString())+1;
        }

        public bool xx2()
        {
            if (y0>1)
            {
                CPlayer p = (CPlayer)players[iCurrentPlayer];
                return p.iStrategy==0;
            }
            else
            {
                return false;
            }
        }

        public bool xx3()
        {
            if (y0>1)
            {
                CPlayer p = (CPlayer)players[iCurrentPlayer];
                return p.iStrategy==1;
            }
            else
            {
                return false;
            }
        }

        public bool xx4()
        {
            if (y0>1)
            {
                CPlayer p = (CPlayer)players[iCurrentPlayer];
                return p.iStrategy==2;
            }
            else
            {
                return false;
            }
        }

        public bool xx5()
        {
            if (y0>1)
            {
                CPlayer p = (CPlayer)players[iCurrentPlayer];
                return p.iStrategy==3;
            }
            else
            {
                return false;
            }
        }

        public bool xx9()
        {
            return share.dPrice == 0;
        }

        private void LinkButton1_Click(object sender, System.EventArgs e)
        {
            A0(0);
            Response.Redirect("result.aspx");
        }
    }
}
```

CShare.cs (класс CShare)

```
using System;

namespace EconomicGame
{
    public class CShare
    {
        public CShare()
        {
            dPrice = 11;
            dPrevPrice = 10;

        }

        public double dPrice = 11;
        public double dPrevPrice = 10;

        public void NextIteration()
        {
            int bid = 0;
            int ask = 0;

            double dP=0;

            double alpha = 5; // коэф. для пересчета курса акции

            foreach (CPlayer p in CGame.players)
            {
                switch(p.iAction)
                {
                    case -1: bid += p.iPortfolio; break;
                    case  0: break;
                    case  1: ask += Convert.ToInt32(Math.Floor(p.dMoney /
CGame.share.dPrice)); break;
                }
            }

            int balance = 0; // измеряется в акциях

            foreach (CPlayer p in CGame.players)
            {
                switch(p.iAction)
                {
                    case -1:
                    { //seller
                        if (bid<ask)
                        {
                            balance -= p.iPortfolio;
                            p.dMoney += p.iPortfolio * CGame.share.dPrice;
                            p.iPortfolio = 0;
                        }
                        else
                        {
                            double udalos = p.iPortfolio;
                            udalos /= bid;
                            udalos *= ask;
                            balance -= Convert.ToInt32(Math.Floor(udalos));
                            p.dMoney += Math.Floor(udalos) * CGame.share.dPrice;
                            p.iPortfolio -= Convert.ToInt32(Math.Floor(udalos));
                        }
                        break;
                    }

                    case  0:
                    {
                        break;
                    }
                    case  1:
                    {  //buyer
                        if (bid>ask)
                        {
                            balance += Convert.ToInt32(Math.Floor(p.dMoney /
CGame.share.dPrice));
                            p.iPortfolio += Convert.ToInt32(Math.Floor(p.dMoney /
CGame.share.dPrice));
```

```
                        p.dMoney -= Convert.ToInt32(Math.Floor(p.dMoney /
CGame.share.dPrice))*CGame.share.dPrice;
                    }
                    else
                    {
                        int udalos =
Convert.ToInt32(Math.Floor((p.dMoney/(CGame.share.dPrice*ask))*bid));
                        balance += udalos;
                        p.dMoney -= udalos * CGame.share.dPrice;
                        p.iPortfolio += udalos;
                    }
                    break;
                }
            }
        }

        if (balance>0)
        {
            CPlayer looser= (CPlayer)CGame.players[0];
            foreach (CPlayer p in CGame.players)
            {
                if ((p.iAction==1)&&(p.iPortfolio > looser.iPortfolio))
                    looser = p;
            }
            looser.iPortfolio -= balance;
            looser.dMoney += balance*CGame.share.dPrice;
            if (looser.iPortfolio<0)
            {
                CGame.log.Write("Bad player (balance>0) = " +
looser.iPortfolio.ToString() + "  " + balance.ToString());
            }

        }
        else if (balance<0)
        {
            CPlayer looser= (CPlayer)CGame.players[0];
            foreach (CPlayer p in CGame.players)
            {
                if ((p.iAction==-1)&&(p.dMoney > looser.dMoney))
                    looser = p;
            }
            looser.iPortfolio += Math.Abs(balance);
            looser.dMoney -= Math.Abs(balance)*CGame.share.dPrice;
            if (looser.iPortfolio<0)
            {
                CGame.log.Write("Bad player (balance<0) = " +
looser.iPortfolio.ToString());
            }

        }

        if (bid + ask!=0)
        {
            dP = (ask - bid)*alpha / (bid + ask);
        }
        else
        {
            dP = 0;
        }

        CGame.share.dPrevPrice = CGame.share.dPrice;
        CGame.share.dPrice += dP;
        if (CGame.share.dPrice<=0)
        {
            CGame.share.dPrice = 0;
            CGame.log.Write("Курс акции меньше меньше или равен нуля. Завершение
работы.");
        }
        else
        {

    CGame.log.LogShareTurn(ask.ToString(),bid.ToString(),CGame.share.dPrice.ToString());
        }
    }
  }
}
```

CPlayer.cs (класс CPlayer)

```
using System;

namespace EconomicGame
{
    public class CPlayer
    {
        public CPlayer()
        {      }

        public void str1()
        {
            if (CGame.share.dPrice < CGame.share.dPrevPrice)
            {
                iAction = -1; // если вчера акции падали - сегодня продаем
            }
            else
            {
                if (dMoney<=0)
                {
                    iAction = -1;
                }
                else
                {
                    iAction = 1;
                }          // если вчера акции росли - сегодня покупаем
            }
        }

        public void str2()
        {
            if(CGame.share.dPrice < CGame.share.dPrevPrice)
            {
                if (dMoney<=0) {
                    iAction = -1;
                } else {
                    iAction = 1;
                }       // если вчера акции падали - сегодня покупаем
            } else {
                iAction = -1; // если вчера акции росли - сегодня продаем
            }
        }

        public void A1(int e)
        {
            bool x6 = xx6(); // вчера игрок продал акции
            bool x7 = xx7(); // вчера игрок пропустил ход
            bool x8 = xx8(); // вчера игрок купил акции

            int y1_old = y1;
            switch(y1)
            {
                case 0:
                    if (e==5&x7)
                    {
                        y1 = 2; // если вчера ничего не было - сегодня продаем
                    }
                    break;

                case 1:
                    if (e==5&x8)
                    {
                        y1 = 2; // если вчера продали - сегодня покупаем
                    }
                    break;

                case 2:
                    if (e==5&x6)
                    {
                        y1 = 1; // если вчера покупали - сегодня продаем
                    }
                    break;
            }

        }

        public void A2(int e)
```

```
        {
            bool x6 = xx6(); // вчера игрок продал акции
            bool x7 = xx7(); // вчера игрок пропустил ход
            bool x8 = xx8(); // вчера игрок купил акции

            int y2_old = y2;
            switch(y2)
            {
                case 0:
                    if (e==5&x7)
                    {
                        y2 = 1; // если вчера ничего не было - сегодня продаем
                    }
                    break;

                case 1:
                    if (e==5&x8)
                    {
                        y2 = 2; // если вчера продали - сегодня покупаем
                    }
                    break;

                case 2:
                    if (e==5&x6)
                    {
                        y2 = 1; // если вчера покупали - сегодня продаем
                    }
                    break;
            }
        }

        public bool xx6()
        {
            if(iAction==-1)
            {
                return true;
            }
            else
            {
                return false;
            }
        }

        public bool xx7()
        {
            if(iAction==0)
            {
                return true;
            }
            else
            {
                return false;
            }
        }

        public bool xx8()
        {
            if(iAction==1)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        public int iAction;
        public int y1;
        public int y2;
        public double dMoney;
        public int iPortfolio = 0;
        public int iStrategy = 0;
        public int iID = 0;
    }
}
```

CLog.cs (класс CLog)

```
using System;
using System.IO;
using System.Text;
using System.Web;

namespace EconomicGame
{
    public class CLog
    {
        public CLog(string filelog, string fileshare, string fileplayers)
        {
            FileLog = filelog;
            FileShare = fileshare;
            FilePlayers = fileplayers;

            FileStream ms = File.Create(EconomicGame.CGame.HomeDir + FileLog);
    ms.Close();
            ms = File.Create(EconomicGame.CGame.HomeDir + FileShare);
    ms.Close();
            ms = File.Create(EconomicGame.CGame.HomeDir + FilePlayers);
    ms.Close();
        }

        public string FileLog;
        public string FileShare;
        public string FilePlayers;

        public void LogPlayerTurn(string ID, string Action, string Money, string
Portfolio)
        {
            StreamWriter output = new StreamWriter(EconomicGame.CGame.HomeDir + FileLog,
true, System.Text.Encoding.Default);
            StreamWriter output1 = new StreamWriter(EconomicGame.CGame.HomeDir +
FilePlayers, true, System.Text.Encoding.Default);
            string line = "День " + CGame.iCurrentDay.ToString() + ": Игрок с номером ";
            line +=  ID ;
            switch (Action)
            {
                case "1" :    line +=  " хочет купить акции (в наличии "+Money+" денег и "
+ Portfolio +" акций)"; break;
                case "0" :    line +=  " пропускает ход"; break;
                case "-1":    line +=  " выставляет на продажу свои акции (в наличии
"+Money+" денег и " + Portfolio +" акций)"; break;
            }

            output.WriteLine(line); output.Flush();    output.Close();
            output1.WriteLine(line); output1.Flush(); output1.Close();
        }

        public void LogAddingPlayer(string ID, string Money, string Portfolio, string
Strategy)
        {
            StreamWriter output = new StreamWriter(EconomicGame.CGame.HomeDir + FileLog,
true, System.Text.Encoding.Default);
            StreamWriter output1 = new StreamWriter(EconomicGame.CGame.HomeDir +
FilePlayers, true, System.Text.Encoding.Default);
            string line = "День " + CGame.iCurrentDay.ToString() + ": Добавляется игрок с
номером ";
            line +=  ID ;
            line +=  ", стратегией " + Strategy;
            line +=  ", деньгами " + Money ;
            line +=  ", акциями " + Portfolio;

            output.WriteLine(line); output.Flush(); output.Close();
            output1.WriteLine(line); output1.Flush(); output1.Close();
        }

        public void LogShareTurn(string ask, string bid, string newprice)
        {
            StreamWriter output = new StreamWriter(EconomicGame.CGame.HomeDir + FileLog,
true, System.Text.Encoding.Default);
            StreamWriter output1 = new StreamWriter(EconomicGame.CGame.HomeDir +
FileShare, true, System.Text.Encoding.Default);

            string line = "День " + CGame.iCurrentDay.ToString() + ": Завершен подсчет
курса акции,";
```

```
            line +=  " спрос = " + ask;
            line +=  " предложение = " + bid ;
            line +=  " новая цена = " + newprice;

            output.WriteLine(line); output.Flush(); output.Close();
            output1.WriteLine(line); output1.Flush(); output1.Close();
        }

        public void LogGame(int State)
        {
            StreamWriter output = new StreamWriter(EconomicGame.CGame.HomeDir + FileLog,
true, System.Text.Encoding.Default);
            string line = "Сообщение от автомата A0: ";
            switch(State)
            {
                case 0: // Старт программы
                    line += "Старт программы (состояние y0=0)";
                    break;

                case 1: // Старт биржи
                    line += "Старт биржи (состояние y0=1)";
                    break;

                case 2: // Ход игрока
                    line += "Ход игрока (состояние y0=2)";
                    break;

                case 3: // Определение поведения игрока по первой стратегии
                    line += "Определение поведения игрока по первой стратегии (состояние
y0=3)";
                    break;

                case 4: // Определение поведения игрока по второй стратегии
                    line += "Определение поведения игрока по второй стратегии (состояние
y0=4)";
                    break;

                case 5: // Определение поведения игрока по третьей стратегии
                    line += "Определение поведения игрока по третьей стратегии (состояние
y0=5)";
                    break;

                case 6: // Определение поведения игрока по четвертой стратегии
                    line += "Определение поведения игрока по четвертой стратегии
(состояние y0=6)";
                    break;

                case 7: // Пересчет курса акции
                    line += "Подсчет курса акций (состояние y0=7)";
                    break;

                case 8: // Завершение программы
                    line += "Завершение программы (состояние y0=8)";
                    break;
            }
            output.WriteLine(line);
            output.Flush();
            output.Close();
        }

        public void Write(string text)
        {
            StreamWriter output = new StreamWriter(EconomicGame.CGame.HomeDir +
FilePlayers, true, System.Text.Encoding.Default);
            output.WriteLine(text);
            output.Flush();
            output.Close();
        }
    }
}
```