

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.

(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка деловой игры для рынка
электроэнергии, Часть I

Специальность Богословская Теология и программное обеспечение

Выполнил (а) Нурбаева А группа БТУ-13-1
(Фамилия и инициалы)

Научный руководитель Аманбаев А.А. К.Ф.-М.Н., доцент
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурзаев А.С., К.Ф.Н., доцент
(Фамилия и инициалы, ученая степень, звание)
А.С. « 23 » 05 2016 г.
(подпись)

по безопасности жизнедеятельности:

Мадалиев И.С., к.т.н., проф.
(Фамилия и инициалы, ученая степень, звание)
И.С. « 16 » 05 2016 г.
(подпись)

по применению вычислительной техники:

Аманбаев А.А. К.Ф.-М.Н., доцент
(Фамилия и инициалы, ученая степень, звание)
А.А. « _____ » _____ 2016 г.
(подпись)

Нормоконтролер: Аманбаев А.А. К.Ф.-М.Н., доцент
(Фамилия и инициалы, ученая степень, звание)

« _____ » _____ 2016 г.

Рецензент: Нуркадиев Айдан Сарсенбекович
(Фамилия и инициалы, ученая степень, звание)

А.С. « 01 » 06 2016 г.
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий и аэрокосмических
Специальность Вычислительная техника и программные обеспечения
Кафедра Компьютерной техники

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Абдусмагжан Аманжол
(фамилия, имя, отчество)

Тема проекта Разработка условий и др. для рынка
электроэнергии, Часть I.

утверждена приказом ректора № 149 от «23» октября 2015 г.

Срок сдачи законченной работы «__» 06 2016 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

1. Файлы проекта разработки условий и др. для рынка электроэнергетики.
2. Созданы все необходимые структуры программы.
3. Рассмотрена форма и разработка программы-конструктора.
4. Применены в видео стили.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Разработка условий и др. для рынка электроэнергетики. В исследовании приняты следующие показатели: мощность линий и срок на проектирование работы. Найден вариант программы для разработки с учетом данных задачи. Также изучены условия проектирования и цена. На разработку программы-конструктора сформированы условия программы. Вся информация о работе.

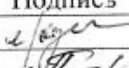


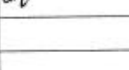

Перечень графического материала (с точным указанием обязательных чертежей)

1. Рисунок-1. Ввод талового объекта в гелевую диалоговую окно панели задвки.
- Рисунок 2 - Панель проекта задвки на экране экрана фотонного речека
- Рисунок 3 - Главное окно участка заводского речека - продажа
- Рисунок 4 - Главное окно участка заводского речека - потребителю.
- Рисунок 5 - Главное окно участка краткосрочного речека - продажа

Рекомендуемая основная литература

1. Гаврилов А.Ю. 100% самоучитель по созданию СИ-страницы / Ю. Гаврилов, В.А. Романовский. -
2. Грачев А. Создаем свой проект на Microsoft Visual Studio 8.0.
3. В.В. Пудовкин. Энциклопедия СИ Учебное пособие. - Москва: БИНОМ, 1994. 400с
4. Г. Шиллер. Самоучитель СИ.
5. Бруно Баба. Просто и ясно о СИ.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Б.И.Д	Магданов И.Ф.	11.15 - 16.15.16	
Эксперт. часть	Федосеева А.И.	24.03 - 25.05.16	
Рецензия	Нуркасов А.С.	01.06.2016	
Нормоконтроль	Григорьев А.А.	06.06.2016	
Руководитель	Александр А.А.	06.06.2016	

Аннотация

В выпускной работе была осуществлена разработка автоматизированной системы «Разработка деловой игры для рынка электроэнергии» управления позволяющего упростить работу персонала, путем автоматизации всех вычислительных и организационных процессов, максимально экономить как денежные, так и временные ресурсы.

Для обеспечения мобильности приложения, был произведен выбор технологии, которое позволяет программе функционировать на компьютерах, подключенных к сети Lan.

Были произведены расчеты затрат на разработку данного продукта, а также рассмотрены вопросы безопасности жизнедеятельности.

Аңдатпа

Дипломдық жобада қызметкерлердің жұмысын жеңілдету, уақыт және ақша ресурстарын сақтау үшін мүмкіндігінше барлық есептеу және ұйымдастырушылық процестерді автоматтандыру арқылы «Электрэнергия нарығы үшін іскерлік ойынды құру» жүйесі жүзеге асырылды.

Қолдану ұтқырлығын қамтамасыз ету үшін бағдарламаға Lan жүйесіне қосылған компьютерлерде жұмыс істеу мүмкіндігін беретін технологиялар қолданылды.

Бұл бағдарламаны дамыту үшін керек шығындары есептелген, сондай-ақ қауіпсіздік сұрақтар қарастырылды.

Abstract

The final work was carried out development of «The development of a business game for the electricity market» management system to simplify the work of staff by automating all the computational and organizational processes, as much as possible to save both money and time resources.

To ensure the mobility of the application was made choice of technologies, which will allow the program to operate on computers connected to the Internet.

It had been calculated the costs to develop the product, as well as the issues of life safety.

Содержание

Введение	7
1 Аналитическая часть	8
1.1 Характеристика игровой модели.	8
1.2 Исходные данные для имитационной модели.....	10
1.3 Порядок проведения игры.....	12
1.4 Долгосрочный рынок (сезонный).....	12
1.5 Краткосрочный рынок (на день вперед).....	14
1.6 Балансирующий рынок (на час в перед).....	16
2 Теоритическая часть.....	17
2.1 Языки C#.....	17
2.2 Особенности языка.....	17
2.3 Разработка среды Microsoft Visual Studio.....	23
2.4 Компоненты.....	23
3 Постановка задачи.....	30
3.1 Составление алгоритма.....	30
3.2 Разработка программы для серверной части	31
3.3 Создание интерфейса и структуры системного оператора	40
4 Техничко–экономическое обоснование.....	48
4.1 Краткая характеристика проекта.....	48
4.2 Расчет затрат на разработку информационных технологий.....	48
4.3 Расчет цены программного продукта.....	56
5 Безопасность жизнедеятельности.....	58
5.1 Анализ потенциально опасных и вредных производственных факторов.....	58

5.2	Требования к микроклимату, ионному составу и концентрации вредных.....	59
5.3	Расчет освещенности.....	61
5.4	Расчет микроклимата рабочей зоны.....	65
5.5	Расчет уровня шума.....	66
	Заключение.....	68
	Список литературы.....	70
	Приложение А.....	71
	Приложение Б.....	72
	Приложение В.....	72
	Приложение Г.....	73
	Приложение Д.....	83

Введение

Согласно действующему Закону Республики Казахстан «Об электроэнергетике», с 01.01.2008 года в имитационном режиме функционирует балансирующий рынок электрической энергии Республики Казахстан, на котором проводятся торги по купле-продаже электроэнергии, созданный с целью обеспечения условий для конкуренции в электроэнергетической отрасли. В мировой практике в торгах на подобных рынках продаётся от __ до __ процент э/э. Торги позволяют участникам сэкономить на закупках электроэнергии или получить большую прибыль от её продажи, чем в случае традиционных двухсторонних договоров между продавцами и покупателями. С января 2019 года единый рынок электроэнергии будет работать на территории стран ЕврАзЭС.

Так как участником торгов может стать любое предприятие, чья потребляемая мощность превышает пороговое значение для вхождения на рынок (на момент выпуска «Методических указаний» - 1 МВт потребляемой электроэнергии в час), то обучающиеся по специальности «Электроэнергетика» магистранты должны иметь представление о принципах работы рынка электроэнергии, как неперенного атрибута отдела энергоснабжения промышленного предприятия. Задача ознакомления с Рынком электроэнергии и мощности решается через проведение деловой игры путём организации торгов с помощью специализированной клиент-серверной программы «Рынок электроэнергии». Учащиеся знакомятся с механизмами функционирования различных бирж рынка, применяют полученные ранее профессиональные навыки для расчёта режимов работы заданной энергосистемы, строят графики нагрузок, определяют себестоимость производства, передачи и распределения электроэнергии, прогнозируют рыночную цену купли/продажи электроэнергии, принимают взвешенные ценовые решения и видят результаты, к которым приводят их действия на рынке.

Методическое пособие предназначено для проведения практического занятия по предмету «Проблемы организации рынка электроэнергии в Казахстане», тема «Рынок электроэнергии», для магистрантов специальности «Электроэнергетика». Продолжительность игрового процесса составляет 4 академических часа, подготовка к деловой игре икс до 8 часов, анализ результатов - 2 часа.

Аналитическая часть

1.1 Характеристика игровой модели.

Игровая модель имитирует реальные процессы, протекающие при работе рынка электроэнергии и мощности, и затрагивает все аспекты, определяющие результаты торгов.

Участниками игры являются студенты учебной группы или подгруппы в количестве 8 человек. Преподаватель исполняет роль наблюдателя и не может влиять на игровой процесс напрямую, а только давать рекомендации про ходу игрового процесса. Игра проводится с применением ЭВМ и требует предварительной подготовки исходных данных.

Роли участников игры:

- рыночный оператор (РО), наблюдает за ходом торгов, утверждает или отклоняет заключённые контракты, следит за соблюдением условий работы рынка и применяет санкции к нарушителям, формирует биржевую сводку по итогам работы рынка;

- системный оператор (СО), он же типе НЭС, контролирует физическую возможность выполнения контрактов по купле-продаже электроэнергии (э/э) в зависимости от физических параметров работы энергосистемы. В функции СО входит оказание услуг по передаче э/э через Национальные электрические сети;

- поставщики э/э (электростанции (ЭС), задача которых состоит в получении максимальной прибыли от продажи э/э на рынке. Поставщики подразделяются на три типа (в соответствии с существующими основными типами электростанций в Казахстане): гидроэлектростанции (в модели типе 2 ГЭС), теплоэлектроцентрали (три ТЭЦ), конденсационные электростанции (две КЭС);

- покупатели э/э, задача которых типе минимизировать затраты на покупку э/э путём участия в торгах на рынке. Покупатели разделяются на два типа: оптовые покупатели (ОП) (в количестве 6 участников рынка) типе часовой расход э/э выше 5 МВт, розничные покупатели (РП) (в количестве 56 участников рынка) типе часовой расход э/э ниже 5 МВт. Покупатели распределяются по отраслевым группам (промышленность, коммунально-бытовой сектор, сельское хозяйство, транспорт) и, в зависимости от объёма закупаемой э/э, относятся к вышеуказанным двум типам потребителей;

- региональные электросетевые компании (РЭК) (в количестве семи участников рынка, по числу электростанций) типе оказывают услуги по передаче и распределению электрической энергии по своим сетям и не влияют на результаты торгов. На долгосрочном и балансирующем рынках провайдеры РЭКов и НЭС выступают в качестве покупателей, приобретая

необходимы объёмы э/э для возмещения потерь э/э (постоянных и переменных (нагрузочных)) на прохождение по своим сетям;

- преподаватель, осуществляет наблюдение за работой рынка.

Число поставщиков и покупателей э/э может изменяться в зависимости от количества участников игры. Для случая типового компьютерного класса в игре участвуют 8 человек. Общее количество участников может быть расширено до 80. Все участники рынка, не представленные реально в виде игроков, участвуют в игре виртуально, через систему искусственного интеллекта, запускаемую рыночным оператором за несколько минут до окончания торговой сессии.

В ходе торгов покупатели и поставщики стремятся купить-продать электроэнергию по наиболее выгодной цене, заключая контракты в ходе работы трёх биржевых сессий. Каждой биржевой сессии соответствует один из трёх типов рынка: рынок долгосрочных контрактов, рынок краткосрочных контрактов, балансирующий рынок. В ходе торгов на каждом типе рынка производится уточнение и заполнение итоговых суточных графиков нагрузки. Долгосрочные контракты предполагают заключение соглашений между покупателями и продавцами о покупке э/э для покрытия базовой (постоянной) части графика нагрузок. Поставка э/э по таким контрактам производится в течение длительного времени (зимний или летний сезон) по постоянному графику. Краткосрочные контракты тире заключаются по результатам торгов методом аукциона на основе уточнённого графика нагрузок в полупиковой зоне графика нагрузок. Длительность контрактов тире до недели. Контракты на балансирующем рынке заключаются по результатам торгов методом встречных предложений. Данные контракты заключаются на срок от одного часа до суток и формируют пиковую часть графика нагрузок. По итогам трёх торговых сессий формируется итоговый график нагрузок и биржевая сводка на текущий час или сутки. Для исключения возможности спекуляций на балансирующем рынке каждый продавец обязан продавать определённую часть своего рабочего графика на определённом типе рынка и снимать её с торгов только при отсутствии на торгах покупателей или если предложения покупателей существенно ниже себестоимости производства энергии в соответствующий период графика нагрузок.

Участники делятся на команды тире предприятия, действующие на рынке потребительских товаров и услуг. Каждая компания получает определённую сумму денег на развитие. Средства компания может вкладывать в производство, в рекламу, в персонал и др.

Кроме того, каждый участник является еще и клиентом и получает средства на собственные нужды. Каждый участник продает свою рабочую силу своему предприятию.

Условия для игры имеют как предприятия, так и каждый участник.

- Раунд 1. Организация. Игроки создают компании, договариваются о заработной плате.

- Раунд 2. Производство. Предприятия создают товары и услуги.
 - Раунд 3. Продажа. Компании продают свои товары и услуги, а клиенты тире приобретают, тратят свои личные деньги.
 - Раунд 4. Расчет финансов. Каждая компания и каждый житель подсчитывает свои финансы. Компании выдают заработную плату сотрудникам (на расходы на след.этап).
 - Раунд 5. Аналитический. Команды проводят анализ и принимают решение о совершенствовании своего производства.
 - Раунд 6. Производство. Команды совершенствуют свои продукты, чтобы привлечь больше прибыли.
 - Раунд 7. Продажа.
 - Раунд 8. Подсчет финансов и определение победителей.
- Побеждает команда, компания которой заработала наибольшее количество денег.

1.2 Исходные данные для имитационной модели

Группе студентов даётся вариант задания, к которому из приложений А, Б и В берутся исходные данные и на ЭВМ выполняется расчёт всех параметров работы энергосистемы. Более подробно о подготовительных процессах и требуемых исходных данных изложено ниже для каждого участника рынка.

Участник группы «*Поставщики*» (минимум 2 участника, по типу электростанций) подготавливает следующий набор входных данных:

- по агрегатная установленная мощность ЭС;
- допустимые уровни мощности по каждому уровню напряжения;
- предельный тариф;
- отпускные тарифы с шин ЭС для каждого из типов рынка. Тарифы формируются на основе расчётных данных о цене э/э для трёх уровней нагрузки (пика, полу пика и провала) типового графика нагрузок заданного региона;
- размер необходимой суммы средств для штрафного денежного обеспечения (с него берутся средства для выплаты штрафов в случае невыполнения обязательств поставки э/э в надлежащем объёме и допущения небаланса). Величина денежного обеспечения соответствует 10, 15 и 20 процент от стоимости подаваемой участником рынка заявки в соответствии с типом рынка. При этом каждая заявка для каждого типа рынка должна обеспечиваться отдельно. После надлежащего выполнения продавцом своих обязательств перед покупателем сумма гарантийного обеспечения возвращается на счёт продавца. В случае нарушения обязательств или ненадлежащего их выполнения с гарантийного счёта удерживается сумма, обеспечивающая компенсацию всех издержек, понесённых другими участниками рынка. В перечень издержек входят: издержки на компенсацию потерь от небаланса мощности в результате

несоблюдения графика выдачи мощности, оплата услуг диспетчеризации и балансировки вызванного производителем небаланса. В случае форс-мажорных обстоятельств, препятствующих исполнению обязательств, продавец обязан предупредить о них оператора рынка и выставить на ближайших торгах свои обязательства по поставкам э/э.

Участник группы «РЭК» (минимум один участник):

- расчётная схема сети РЭК до фидеров на п/с потребителей;
- рассчитанные характерные режимы нагрузок потребителей в данной сети (расчёт производить в любой программе, позволяющей считать установившиеся режимы работы сетей (РТПЗ, RASTR, DAKAR и др.));

- заданный тариф на услуги по передаче э/э; (нужно ли представить действующую методику расчёта тарифов электросетевых организаций для расчёта тарифов РЭК-ми или достаточно ограничиться выдачей им тарифов? И кстати, может, для РЭК-в тоже предусмотреть несколько уровней тарифов, в соответствии с режимами работы сети?)

- результаты расчётов постоянных и переменных потерь электроэнергии для трёх уровней нагрузки (пика, полу пика и провала) типового графика нагрузок соответствующего участка сети заданного региона;

- цены на покупку потерь в торгах на долгосрочном и балансирующем рынках (преподавателем выдаются результаты предыдущих торгов для определения допустимых уровней цен для каждого типа рынка).

Участник группы «Потребители» (минимум два, для представления розничного и оптового секторов):

- типовой суточный график нагрузок для зимнего и летнего сезонов;

- планируемый уровень мощности, закупаемой на всех трёх рынках в соответствии с графиком нагрузок;

- тариф на вводе потребителя, учитывающий в себе тариф на шинах ЭС плюс тариф на услуги по передаче, диспетчеризации и балансировке э/э. Для обеспечения представления покупателя об уровне соответствующих тарифов ему выдаются данные о предыдущей торговой сессии по каждому типу рынка, с указанием доли ЭС, НЭС и РЭК в итоговом тарифе;

- размер необходимой суммы средств гарантийного обеспечения для покрытия издержек, понесённых другими участниками рынка из-за ненадлежащего выполнения покупателем своих обязательств. Величина денежного обеспечения соответствует, по типам рынка, 10, 15 и 20 процент от стоимости подаваемой участником рынка заявки. При этом каждая заявка для каждого типа рынка должна обеспечиваться отдельно. После надлежащего выполнения покупателем своих обязательств перед продавцом и передающей организацией сумма гарантийного обеспечения возвращается на счёт покупателя. В случае нарушения обязательств или ненадлежащего их выполнения с гарантийного счёта удерживается сумма, обеспечивающая компенсацию всех издержек, понесённых другими участниками рынка. В перечень издержек входят: издержки на

компенсацию потерь от небаланса мощности в результате несоблюдения покупателем графика отбора мощности, оплата услуг диспетчеризации и балансировки вызванного покупателем небаланса. При этом стоимость заявки покупателя не изменяется и оплачивается в полном размере. В случае форс-мажорных обстоятельств, препятствующих исполнению обязательств, покупатель обязан предупредить о них оператора рынка и продать на ближайших торгах свои обязательства по покупке э/э.

Системный оператор подготавливает:

- данные о характерном режиме работы своих сетей для трёх уровней графика нагрузок;
- действующий тариф на услуги по передаче э/э по сетям НЭС и её диспетчеризации;
- данные по постоянным и переменным потерям э/э в своих сетях для каждого уровня нагрузки;
- цены на покупку потерь э/э в торгах на долгосрочном и балансирующем рынках.

Данные для подготовки к проведению торгов включают в себя номер варианта исходных данных из приложения А, итоги «предыдущей» торговой сессии по каждому типу рынка с целью определения ценовых границ участниками торгов, схему расположения участников рынка на физической карте.

Подготовка к деловой игре включает в себя расчёт входных данных по выданному варианту в соответствии с приложением Г и расчёт режимов работы электрических сетей заданного региона в программе расчёта режимов. Выполненный расчёт необходимо сохранить для загрузки системным оператором и РЭК-м во время проведения деловой игры.

1.3 Порядок проведения игры

Каждому участнику игры выдаётся имя и пароль пользователя для запуска клиентской части деловой игры. После запуска каждый участник в заголовке окна видит назначенную ему роль и текущий тип рынка.

В игре произведена предварительная регистрация участников с назначением определённой игровой роли каждому имени входа. Преподаватель раздаёт имена входа в соответствии с назначенными ролями, участники входят в клиент программы под выданными им именами и паролями. В заголовке программы отображается информация о роли пользователя и текущем типе торгов (если они запущены рыночным оператором).

1.4 Долгосрочный рынок (сезонный)

Торги начинаются с проведения торговой сессии на долгосрочном рынке. В этих торгах участвуют объёмы э/э для покрытия постоянной части

графика нагрузок на весь отопительный или неотапительный сезоны. Торги проводятся в форме покупки поданных ЭС заявок и заключения по итогам торгов двухсторонних контрактов между участниками рынка тире покупателями и продавцами электроэнергии. В роли покупателей выступают так же и энергопередающие организации, закупающие постоянные потери, не зависящие от уровня нагрузок. По окончании торгов все прошедшие проверку на соответствие допустимым условиям работы сети и согласованные контракты утверждаются Оператором рынка, а затем оплачиваются покупателями.

Участник торгов-продавец подаёт заявку на участие в торгах через диалоговые окна подачи заявки (меню «Заявка») сразу на весь сезон. Дата начала и длительность поставок (182 или 183 дня) устанавливаются Оператором рынка для всех участников, привязываются к началу или окончанию отопительного сезона и указываются в итоговом договоре о поставках. В заявке даётся информация только об объёме поставляемой э/э, без указания её стоимости. Но, при этом, поставщик должен знать себестоимость производства э/э для периода провала графика нагрузок и использовать её в качестве опорной при оценке предложений покупателей.

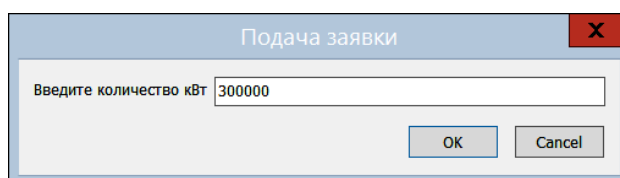


Рисунок 1 тире Ввод часового объёма э/э через диалоговое окно подачи заявки

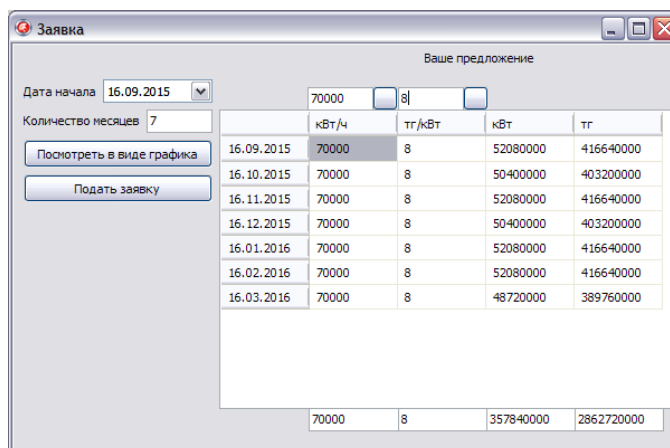


Рисунок 2 тире Панель подачи заявки поставщика долгосрочного рынка

Информация о заявке доступна через кнопку «Заявка» меню «Заявки».

После подачи заявки в левой части главного окна продавца отображается его текущая заявка. В правой части будут отображаться предложения покупателей по покупке э/э.

Рисунок 3 тире Главное окно участника долгосрочного рынка тире продавца

Ваши заявки на покупку электроэнергии				Заявки на продажу электроэнергии			
Покупатель	кВт/ч	Стоимость	Дата подачи	Продавец	кВт/ч	Стоимость	Дата подачи
▶ КОП1	5623200000	0608800000	23.07.2015 16:47:26	▶ ГЭС 1	766800000	9201600000	23.07.2015 16:47:26
				ТЭЦ1	1533600000	0735200000	23.07.2015 16:47:26
				КЭС1	7668000000	1344000000	23.07.2015 16:47:26

Рисунок 3 тире Главное окно участника долгосрочного рынка тире продавца

Рисунок 4 тире Главное окно участника долгосрочного рынка тире потребителя

Ваши заявки на покупку электроэнергии				Заявки на продажу электроэнергии			
Покупатель	кВт/ч	Стоимость	Дата подачи	Продавец	кВт/ч	Стоимость	Дата подачи
▶ КОП1	5623200000	0608800000	23.07.2015 16:47:26	▶ ГЭС 1	766800000	9201600000	23.07.2015 16:47:26
				ТЭЦ1	1533600000	0735200000	23.07.2015 16:47:26
				КЭС1	7668000000	1344000000	23.07.2015 16:47:26

Рисунок 4 тире Главное окно участника долгосрочного рынка тире потребителя

Выделив заинтересовавшее предложение и выбрав в меню «Заявка» пункт «Просмотреть заявку» (или дважды щёлкнув по заявке в главном окне), можно получить подробную информацию о заявке и купить её, нажав соответствующую кнопку и указав в появившихся диалоговых окнах покупаемый объём и предлагаемую цену. Если объём заявки меньше объёма, требуемого покупателю, то заявка считается полностью удовлетворённой и автоматически убирается из списка заявок, а покупатель может докупить оставшийся объём э/э из заявки другого продавца. Если требуемый покупателю объём меньше, чем объём заявки, то заявка автоматически публикуется повторно, с указанием объёма, оставшегося после принятия предложения покупателя. Когда все присутствующие участники торгов закончат процессы купли-продажи, оператор рынка запускает в работу виртуальных участников. После окончания первой сессии торгов все купленные заявки отправляются оператором рынка на утверждение системному оператору на предмет соответствия заявок допустимому режиму работы электрической сети. В случае наличия отклонённых заявок проводится дополнительная сессия торгов для отклонённых в первой сессии заявок. При этом заявки имеют преимущество по времени подачи, то есть, чем раньше заявка подана, тем больший приоритет она имеет при утверждении системным оператором в случае, если следующая заявка вносит недопустимый дисбаланс в работу сети.

1.5 Краткосрочный рынок (на день вперёд)

В торгах на день вперёд участвуют объёмы э/э в переменной части графика нагрузок, корректирующие и дополняющие объёмы, закупленные в торгах на долгосрочном рынке. Торги проводятся в форме аукциона.

Участник рынка тире продавец выставляет своё предложение на аукцион в виде оформленной заявки с указанием минимальной цены. Другой участник рынка тире покупатель, дважды щёлкнув по заявке, задаёт в появившемся окне объём требуемый объём и свою цену, большую цены продавца. Ставка покупателя становится ведущей, и новая цена отображается в качестве минимальной цены ставки. Любой другой покупатель может поднять ставку первого покупателя, назначив более высокую цену, в таком случае уже его ставка будет ведущей и будет иметь приоритет у продавца и у системного оператора. То есть, продавец может продать всю выставленную на торги э/э второму покупателю, если его ставка является ведущей. В таком случае первому покупателю может достаться объём э/э, оставшийся после удовлетворения заявки второго покупателя (предложение продавца автоматически выставляется на торги с объёмом, уменьшенным на объём удовлетворённой ставки, и ценой, первоначально выставленной продавцом), в случае, если его ставку не перебьёт ещё один покупатель. Так же первый покупатель может поднять ставку повторно, выше цены, предложенной вторым покупателем. Продавец может отключить повторное выставление своей заявки и выставить её на торги вручную, с иной ценой или не выставлять совсем.

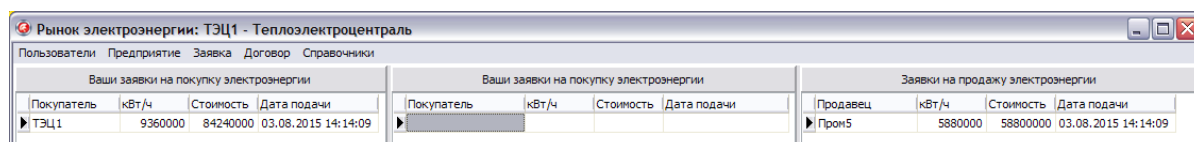


Рисунок 5 - Главное окно участника краткосрочного рынка - продавца

Выделив заинтересовавшее предложение и выбрав в меню «Заявка» пункт «Просмотреть заявку» (или дважды щёлкнув по заявке в главном окне), можно получить подробную информацию о заявке и купить её, нажав соответствующую кнопку. В этом случае покупателю заявки в диалоговом окне будет сделан запрос о покупаемом объёме, цене и графике получения мощности. Когда все присутствующие участники торгов закончат процессы купли-продажи, оператор рынка запускает в работу виртуальных участников рынка. После окончания торговой сессии все купленные заявки отправляются оператором рынка на утверждение системному оператору на предмет соответствия заявок допустимому режиму работы электрической сети. В случае наличия отклонённых заявок проводится дополнительная сессия торгов для отклонённых в первой сессии заявок. При этом заявки имеют преимущество по времени подачи, то есть, чем раньше заявка подана, тем больший приоритет она имеет при утверждении системным оператором в случае, если следующая заявка вносит недопустимый дисбаланс в работу сети.

Представители РЭК-в не участвуют в работе краткосрочного рынка, т. к. на этом этапе итоговый график нагрузок ещё не сформирован и нельзя рассчитать полные нагрузочные потери э/э.

1.6 Балансирующий рынок (на час вперёд)

Торги на балансирующем рынке предназначены для уточнения и окончательного формирования итогового суточного графика нагрузок.

Торги проводятся путём приёма заявок от продавцов и покупателей э/э, при этом покупателем/продавцом может выступать как потребитель, так и поставщик э/э. Например, на ЭС возникла необходимость срочного ремонта оборудования и проданные объёмы э/э не могут быть поставлены потребителю. В этом случае ЭС покупает на торгах объём э/э, который она не может поставить, у другого поставщика. Естественно, необходимость в такой сделке должна быть подтверждена документально, для исключения спекуляций. В учебных торгах такая ситуация может быть задана преподавателем. Точно так же и потребитель может выступить продавцом э/э, если у него возник излишек купленной ранее э/э, например, в результате непланируемой остановки какого-либо оборудования.

Торги на балансирующем рынке проводятся в два этапа тире на первом этапе формируется график нагрузок, на втором тире закупаются потери.

2 Теоритическая часть

2.1 Разработка среды Microsoft Visual Studio и языка C#

C# (произносится «си шарп») — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате ИКСМЛ.

Переняв многое от своих предшественников - языков C++, Pascal, Модула, Smalltalk и, в особенности, Java — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

2.2 Особенности языка

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, Сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Название языка. Название «Си шарп» (от англ. sharp — диэз) происходит от музыкальной нотации, где знак диэз означает повышение

соответствующего ноте звука на полутон, что аналогично названию языка C++, где «++» обозначает инкремент переменной. Название также является игрой с цепочкой C стрелка C++ стрелка C++++(C#), так как символ «#» можно составить из 4х знаков «+».



Рисунок тире 6 Нота с диезом.

Вследствие технических ограничений на отображение (стандартные шрифты, браузеры и т. д.) и того обстоятельства, что знак диез # не представлен на стандартной клавиатуре, знак номера # был выбран для представления знака диез при записи имени языка программирования. Это соглашение отражено в Спецификации Языка C# ECMA-334. Тем не менее, на практике (например, при размещении рекламы и коробочном дизайне), Майкрософт использует предназначенный музыкальный знак.

Названия языков программирования не принято переводить, поэтому язык следует называть по-английски «Си шарп».

Стандартизация. C# стандартизирован в ECMA (ECMA-334) и ISO (ISO/IEC 23270).

Известно как минимум о трёх независимых реализациях C#, базирующихся на этой спецификации и находящихся в настоящее время на различных стадиях разработки:

- Mono, начата компанией IKSimian, продолжена её покупателем и преемником Novell, а затем ИКСamarin.
- dotGNU и Portable.NET, разрабатываемые Free Software Foundation.

На протяжении разработки языка C# было выпущено несколько его версий:

Таблица 2.1 - версий языка C#

Версия	Спецификация языка	Дата	.NET Framework	Visual
--------	--------------------	------	----------------	--------

	ECMA	ISO/IEC	Microsoft			Studio
C# 1.0	Декабрь 2002	Апрель 2003	Январь 2002	Январь 2002	.NET Framework 1.0	Visual Studio .NET (2002)
C# 1.2			Октябрь 2003	Апрель 2003	.NET Framework 1.1	Visual Studio .NET 2003

Продолжение таблицы 2.1

C# 2.0	Июнь 2006	Сентябрь 2006	Сентябрь 2005	Ноябрь 2005	.NET Framework 2.0	Visual Studio 2005
C# 3.0	Отсутствует		Август 2007	Ноябрь 2007	.NET Framework 3.5	Visual Studio 2008
C# 4.0			Апрель 2010	Апрель 2010	.NET Framework 4	Visual Studio 2010
C# 5.0			Июнь 2013	Август 2012	.NET Framework 4.5	Visual Studio 2012
C# 6.0			Июль 2015	Июль 2015	.NET Framework 4.6	Visual Studio 2015

Версия 1.0. Проект C# был начат в декабре 1998 и получил кодовое название COOL (C-style Object Oriented Language). Версия 1.0 была анонсирована вместе с платформой .NET в июне 2000 года, тогда же появилась и первая общедоступная бета-версия; C# 1.0 окончательно вышел вместе с Microsoft Visual Studio .NET в феврале 2002 года.

Первая версия C# напоминала по своим возможностям Java 1.4, несколько их расширяя: так, в C# имелись *свойства* (выглядящие в коде как поля объекта, но на деле вызывающие при обращении к ним методы класса), индексомеры (подобные свойствам, но принимающие параметр как индекс массива), события, делегаты, циклы foreach, структуры, передаваемые по значению, автоматическое преобразование встроенных

типов в объекты при необходимости (*boxing*), атрибуты, встроенные средства взаимодействия с неуправляемым кодом (DLL, COM) и прочее.

Кроме того, в C# решено было перенести некоторые возможности C++, отсутствовавшие в Java: беззнаковые типы, перегрузку операторов (с некоторыми ограничениями, в отличие от C++), передача параметров в метод по ссылке, методы с переменным числом параметров, оператор `goto` (с ограничениями). Также в C# оставили ограниченную возможность работы с указателями — в местах кода, специально обозначенных словом `unsafe` и при указании специальной опции компилятору.

Версия 2.0. Проект спецификации C# 2.0 впервые был опубликован Microsoft в октябре 2003 года; в 2004 году выходили бета-версии (проект с кодовым названием Whidbey), C# 2.0 окончательно вышел 7 ноября 2005 года вместе с Visual Studio 2005 и .NET 2.0.

Новые возможности в версии 2.0

- Частичные типы (разделение реализации класса более чем на один файл).
- Обобщённые, или параметризованные типы (*generics*). В отличие от шаблонов C++, они поддерживают некоторые дополнительные возможности и работают на уровне виртуальной машины. Вместе с тем, параметрами обобщённого типа не могут быть выражения, они не могут быть полностью или частично специализированы, не поддерживают шаблонных параметров по умолчанию, от шаблонного параметра нельзя наследоваться, и т. д. скобка прямая13скобка прямая
- Новая форма итератора, позволяющая создавать сопрограммы с помощью ключевого слова `yield`, подобно Python и Ruby.
- Анонимные методы, обеспечивающие функциональность замыкания.
- Оператор `??`: `return obj1 ?? obj2`; означает (в нотации C# 1.0) `return obj1 != null ? obj1 : obj2`;
- Обнуляемые ('nullable') типы-значения (обозначаемые вопросительным знаком, например, `int? i = null`);, представляющие собой те же самые типы-значения, способные принимать также значение `null`. Такие типы позволяют улучшить взаимодействие с базами данных через язык SQL.
- Возможность создавать хранимые процедуры, триггеры и даже типы данных на .Net языках (в том числе и на C#).
- Поддержка 64-разрядных вычислений, что кроме всего прочего, позволяет увеличить адресное пространство и использовать 64-разрядные примитивные типы данных.

Версия 3.0. В июне 2004 года Андерс Хейлсберг впервые рассказал на сайте Microsoft о планируемых расширениях языка в C#3.0. скобка прямая14скобка прямая В сентябре 2005 года вышли проект спецификации C# 3.0 и бета-версия C# 3.0, устанавливаемая в виде дополнения к существующим Visual Studio 2005 и .NET 2.0. Окончательно эта версия языка вошла в Visual Studio 2008 и .NET 3.5.

Деревья выражений: лямбда-выражения теперь могут представляться в виде структуры данных, доступной для обхода во время выполнения, тем самым позволяя транслировать строго типизированные C#-выражения в другие домены (например, выражения SQL).

Вывод типов локальной переменной: `var икс = "hello";` вместо `string икс = "hello";`;

Безымянные типы: `var икс = new { Name = "James" };`;

Методы-расширения — добавление метода в существующий класс с помощью ключевого слова `this` при первом параметре статической функции.

```
public static class StringExtensions
{
    public static int ToInt32(this string val)
    {
        return Int32.Parse(val);
    }
}
// ...
string s = "10";
int икс = s.ToInt32();
```

Автоматические свойства: компилятор сгенерирует закрытое (`private`) поле и соответствующие аксессор и мутатор для кода вида `public string Name { get; private set; }`

C# 3.0 совместим с C# 2.0 по генерируемому MSIL-коду; улучшения в языке — чисто синтаксические и реализуются на этапе компиляции. Например, многие из интегрированных запросов LINQ можно осуществить, используя безымянные делегаты в сочетании с предикатными методами над контейнерами наподобие `List.FindAll` и `List.RemoveAll`.

Версия 4.0. Превью C# 4.0 было представлено в конце 2008 года, вместе с СТР-версией Visual Studio 2010.

Visual Basic 10.0 и C# 4.0 были выпущены в апреле 2010 года, одновременно с выпуском Visual Studio 2010.

Новые возможности в версии 4.0. Возможность использования позднего связывания, для использования:

- с языками с динамической типизацией (Python, Ruby)
- с COM-объектами
- отражения (`reflection`)
- объектов с изменяемой структурой (DOM). Появляется ключевое слово `dynamic`.
- Именованные и опциональные параметры
- Новые возможности COM interop
- Ковариантность и контравариантность

- Контракты в коде (Code Contracts)

Примеры:

```
dynamic calc = GetCalculator();  
int sum = calc.Add(10, 20); // Динамический вызов  
public void SomeMethod(int x, int y = 5, int z = 7); // Опциональные  
параметры
```

Пример «Hello, World!»

Ниже представлен код классической программы «Hello world» на C# для консольного приложения:

```
using System;  
  
namespace Еикsample  
{  
    class Program  
    {  
        static void Main()  
        {  
            Console.WriteLine("Hello World!"); // Вывод заданного текста в  
консоль  
            Console.ReadKey(); // Ожидание нажатия клавиши  
пользователем  
        }  
    }  
}
```

и код этой же программы для приложения Windows Forms:

```
// assembly: System.dll  
// assembly: System.Drawing.dll  
// assembly: System.Windows.Forms.dll  
using System;  
using System.Drawing;  
using System.Windows.Forms;  
  
namespace WindowsForms  
{  
    public class Program  
    {  
        скобка прямаяSTAThreadскобка прямая  
        public static void Main()  
        {  
            new DemoForm().ShowDialog();  
        }  
    }  
  
    public class DemoForm : Form
```

```

    {
        Label label = new Label();
    public DemoForm()
        {
            label.Текст = "Hello World!";
            this.Controls.Add(label);
            this.StartPosition = FormStartPosition.CenterScreen;
            this.BackColor = Color.White;
            this.FormBorderStyle = FormBorderStyle.Fixed3D;
        }
    }
}

```

2.3 Разработка среды Microsoft Visual Studio

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, ИКСбоикс, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

2.4 Компоненты

Visual Studio включает один или несколько компонентов из следующих:

- Visual Basic .NET, а до его появления — Visual Basic
- Visual C++
- Visual C#
- Visual F# (включён начиная с Visual Studio 2010)

Многие варианты поставки также включают:

Microsoft SQL Server либо Microsoft SQL Server Еикспресс

В прошлом в состав Visual Studio также входили продукты:

- Visual InterDev
- Visual J++
- Visual J#
- Visual ФоиксPro
- Visual Source Safe — файл-серверная система управления версиями

Таблица типе 2.2 Версии Visual Studio

Официальное название	Кодовое название	Внутренняя версия	Версии .NET Framework	Дата выхода
Visual Studio	N/A	4.0	N/A	Апрель 1995
Visual Studio97	<i>Boston</i>	5.0	N/A	Февраль 1997
Visual Studio 6.0	<i>Aspen</i>	6.0	N/A	Июнь 1998
Visual Studio .NET (2002)	<i>Rainier</i>	7.0	1.0	2002-02-13
Visual Studio .NET 2003	<i>Everett</i>	7.1	1.1	2003-04-24
Visual Studio 2005	<i>Whidbey</i>	8.0	2.0, 3.0	2005-11-07
Visual Studio 2008	<i>Orcas</i>	9.0	2.0, 3.0, 3.5	2007-11-19
Visual Studio 2010	<i>Dev10/Rosario</i>	10.0	2.0 — 4.0	2010-04-12
Visual Studio 2012	<i>Dev11</i>	11.0	2.0 — 4.5.2	2012-08-15
Visual Studio	<i>Dev12</i>	12.0	2.0 — 4.5.2	2013-10-17

2013				
Visual Studio 2015	<i>Dev14</i>	14.0	2.0 — 4.6	2015-07-20

До выхода Visual Studio Version 4.0 среды разработки Visual Basic 3, Visual C++, Visual ФойксPro и Source Safe поставлялись в качестве самостоятельных пакетов.

Visual Studio 97. Visual Studio 97 — первая выпущенная версия Visual Studio, в которой впервые были собраны вместе различные средства разработки ПО. Она была выпущена в двух версиях — Professional и Enterprise, и включала в себя Visual Basic 5.0, Visual C++ 5.0, Visual J++ 1.1, Visual ФойксPro 5.0 и впервые появившуюся среду разработки ASP — Visual InterDev. Visual Studio 97 была первой попыткой Microsoft создать единую среду для разработки на разных языках программирования: Visual C++, Visual J++, Visual InterDev и MSDN использовали одну среду, называемую Developer Studio. Visual Basic и Visual ФойксPro использовали отдельные среды для разработки.

Visual Studio 6.0 (1998). Visual Studio 6.0 — последняя версия Visual Studio, работающая на платформе Windows 9икс (выпущена в июне 1998 года). По-прежнему популярна среди программистов, использующих Visual Basic. Данная версия являлась основной средой разработки приложений под Windows от Microsoft до появления платформы .NET.

Visual Studio .NET (2002). Visual Studio .NET (кодовое имя *Rainier*; внутренняя версия 7.0) — выпущена в феврале 2002 года (включает .NET Framework 1.0). Service Pack 1 для Visual Studio .NET (2002) выпущен в марте 2002 года.

Visual Studio .NET 2003. Visual Studio .NET 2003 (кодовое имя *Everett*; внутренняя версия 7.1) — выпущена в апреле 2003 года (включает .NET Framework 1.1). В апреле 2005 года Microsoft объявила о специальном выпуске среды, получившей название Microsoft Visual Studio .NET 2003 Professional Special Edition. Спецвыпуск представлял собой обычное издание Visual Studio .NET 2003 Professional Edition с добавлением в комплект серверного ПО и других инструментов (в частности, операционной системы Windows Server 2003 Standard Edition и SQL Server 2000 Developer Edition). Для поощрения перехода на новую среду разработки корпорацией была объявлена специальная цена обновления, действовавшая при переходе на Visual Studio .NET 2003 Professional Special Edition с большого числа средств разработки как Microsoft, так и основных его конкурентов ^{скобка прямая} ~~скобка прямая~~.

Service Pack 1 для Visual Studio .NET 2003 выпущен 13 сентября 2006 года.

Visual Studio 2005. Visual Studio 2005 (кодовое имя *Whidbey*; внутренняя версия 8.0) — выпущена в конце октября 2005 года (включает .NET Framework 2.0). Последняя официально работающая на Windows 2000. В начале ноября 2005 года также вышла серия продуктов в редакции Еикспресс: Visual C++ 2005 Еикспресс, Visual Basic 2005 Еикспресс, Visual C# 2005 Еикспресс и др с икобка прямая4скобка прямая 19 апреля 2006 года редакция Еикспресс стала бесплатной. Service Pack 1 для VS2005 скобка прямая5скобка прямая и всех Еикспресс-редакций скобка прямаябскобка прямая выпущен 14 декабря 2006 года. Дополнительный патч для SP1, решающий проблему совместимости с Windows Vista, выпущен 6 марта 2007 года.

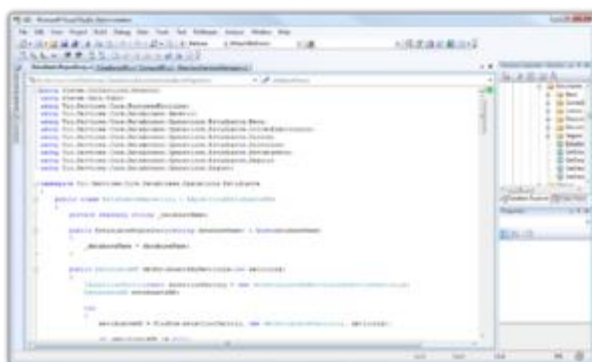


Рисунок типе 7 Внешний вид Visual Studio 2008 SP1 с программой на языке C# в Windows Vista

Visual Studio 2008. Visual Studio 2008 (кодовое имя *Orcas*; внутренняя версия 9.0) - выпущена 19 ноября 2007 года, одновременно с .NET Framework 3.5. Нацелена на создание приложений для ОС Windows Vista (но поддерживает и ИКСР), Microsoft Office 2007 и веб-приложений. Включает в себя LINQ, новые версии языков C# и Visual Basic. В студию не вошёл Visual J#. С 28 октября 2008 года впервые доступна версия на русском языке.

Visual Studio 2010. Visual Studio 2010 (кодовое имя *Hawaii*, для Ultimate — *Rosario*; внутренняя версия 10.0) — выпущена 12 апреля 2010 года вместе с .NET Framework 4.0. Visual Studio включает поддержку языков C# 4.0 и Visual Basic .NET 10.0, а также языка F#, отсутствовавшего в предыдущих версиях.

Visual Studio 2012. Visual Studio 2012 распространяется в тех же редакциях, что и 2010. Изменения коснулись Visual Studio 2012 Еикспресс — устанавливаются все языки программирования, а не один, как раньше (Visual Basic 2010 Еикспресс, Visual C# 2010 Еикспресс), а также теперь существует пять версий Visual Studio Еикспресс: Visual Studio Еикспресс 2012 для Web, Visual Studio Еикспресс 2012 для Windows 8, Visual Studio Еикспресс 2012 для Windows Desktop, Visual Studio Еикспресс 2012 для Windows Phone и Visual Studio Team Foundation Server Еикспресс 2012.

Все версии распространяются как отдельные приложения. Visual Studio Еикспресс 2012 для Windows 8 позволяет разрабатывать приложения для Windows Store с Modern-интерфейсом, а Visual Studio Еикспресс 2012 для Windows Desktop позволяет разрабатывать «классические» приложения для Рабочего стола. Что касается Visual Studio Team Foundation Server Еикспресс 2012, то эта версия поставляется с оболочкой Visual Studio 2012.

Разрабатывать приложения на С++ с помощью Visual Studio 2012 можно только под Windows 7 SP1 и Windows 8^{скобка прямая7скобка прямая}. Вышло исправление, позволяющее компилировать приложения для запуска под Windows ИКСР.

Visual Studio 2013. Финальный релиз Visual Studio 2013 стал доступен для загрузки 17 октября 2013 года вместе с .NET 4.5.1.

Visual Studio 2015. 12 ноября 2014 года было объявлено, что «Visual Studio 2015» принято в качестве окончательного варианта имени продукта^{скобка прямая8скобка прямая}.

Visual Studio 2015 предоставляется в трёх редакциях: бесплатной Community Edition, объединяющей все Еикспресс-версии, и платных Professional Edition для небольших проектов и Enterprise Edition для крупных проектов.

Первый СТР был выпущен 2 июня 2014 года, затем 29 апреля 2015 года был выпущен Release Candidate. Финальный релиз Visual Studio 2015 стал доступен для скачивания 20 июля 2015 года скобка прямая9скобка прямая.

Visual Studio «15». Visual Studio «15» Preview была выпущена 30 марта 2016 года. Visual Studio «15» Preview 2 была выпущена 10 мая 2016 года скобка прямая10скобка прямая.

Редакции Visual Studio. Основная статья: Microsoft Visual Studio Еикспресс. Набор легковесных сред разработки, представляющих собой урезанную версию Visual Studio. Она включает в себя небольшой набор инструментов, в отличие от полных версий: отсутствует дизайнер классов и многие другие инструменты, а также поддержка плагинов и удалённых баз данных в дизайнера данных. Microsoft позиционирует эту линейку IDE для студентов и любителей. На настоящий момент существуют следующие Еикспресс-редакции:

- Visual Basic Еикспресс
- Visual С++ Еикспресс
- Visual С# Еикспресс
- Visual Web Developer Еикспресс

Вместе с Visual Studio 2012 были выпущены новые Еикспресс-версии продукта:

- Visual Studio Еикспресс 2012 for Web — для web-разработчиков
- Visual Studio Еикспресс 2012 for Windows 8 — для разработки программ с modern-интерфейсом (языки: С#, Visual Basic.Net, С++, JavaScript)

- Visual Studio Еикспress 2012 for Windows Desktop — для разработки обычных десктопных приложений (языки: С#, Visual Basic.Net, С++)

- Visual Studio Еикспress 2012 for Windows Phone — для разработчиков под платформы Windows Phone 7.5 и 8.0

- Visual Studio Team Foundation Server Еикспress 2012

Ключевыми особенностями этих еикспress-версий продуктов являются:

- ориентирование на цель разработки, а не на язык

- необходимость регулярно продлевать бесплатную регистрацию для индивидуальных разработчиков, если разработка на Еикспress-версии ведётся не с целью обучения

- поддержка компиляции 64-битного кода

- поддержка unit-тестов

Visual Studio LightSwitch. Microsoft Visual Studio LightSwitch — это среда разработки, нацеленная на создание отрасли приложений, построенных на существующих .NET-технологиях и платформах Microsoft. Создаваемые приложения состоят из трёх ярусов: пользовательский интерфейс на Silverlight; логика и доступ к данным на сервисах WCF RIA и Entity Framework; хранение данных с помощью Microsoft SQL Server Еикспress, Microsoft SQL Server или SQL Azure. LightSwitch также поддерживает другие источники данных, включая SharePoint. LightSwitch включает в себя графические дизайнеры сущностей и их отношений, запросов, а также интерфейса пользователя. Бизнес-логика может быть написана на Visual Basic или на Visual С#. LightSwitch может быть установлен как самостоятельное приложение или как дополнение к Visual Studio 2010 Professional и более высоким редакциям.

Visual Studio Standard. Данная редакция предоставляет IDE для всех поддерживаемых продуктов и поддерживает полную версию библиотеки MSDN. Поддерживается как редактирование ИКСМЛ и ИКСЛТ, так и средства для тестирования объектов. Однако отсутствует обозреватель серверов и интеграция с Microsoft SQL Server. Поддержка разработки под мобильные устройства изначально была включена в Visual Studio 2005 Standard, но в версии 2008 она доступна только в Professional-редакции. Начиная с версии 2010 более не существует.

Visual Studio Professional. Редакция включает все возможности Standard Edition, расширяя их дополнительными, такими, как интеграция с Microsoft SQL Server и поддержка удалённой отладки.

Visual Studio Tools for Office. Visual Studio Tools for Office включает SDK и расширение для Visual Studio, которое содержит утилиты для разработки под платформу Microsoft Office. Начиная с Visual Studio 2008 включено в версии Professional и выше.

Visual Studio Team System. Основная статья: Visual Studio Team System

Предоставляет набор инструментов для совместной разработки, подсчёта метрик и создания отчётов, в дополнение к возможностям Professional редакции. Имеются различные редакции VSTS, разделяемые по ролям, для которых продукт будет использоваться:

- Team Explorer (клиент для TFS)
- Architecture Edition
- Database Edition
- Development Edition
- Test Edition

Совмещённая функциональность всех четырёх редакций представлена в отдельном пакете Visual Studio Team Suite Edition. Функциональность Database Edition будет совмещена с Development Edition в будущей версии пакета — Visual Studio 2010.

Помимо клиентских приложений, Team System также включает в себя Team Foundation Server.

Компилятор как сервис. В сентябре 2011 года было объявлено, что в октябре Microsoft выпустит специальную версию компилятора, разрабатываемого в рамках проекта Roslyn. Целью данного проекта была разработка «компилятора в виде сервиса» с возможностью выдачи программистам всей генерируемой компилятором информации. Данный компилятор будет обладать поддержкой языков C# и Visual Basic с полной совместимостью с Visual Studio. Кроме того, заявлена возможность конвертации кода с одного языка на другой ^{скобка прямая1} _{скобка прямая}.

Visual Studio Code. Visual Studio Code — среда разработки, работающая под Linux, OS ИКС и Windows. Включает в себя отладчик и клиент для Git. Исходный код открыт. 14 апреля 2016 года была выпущена версия 1.0.

Дополнения. Visual Studio построена на архитектуре, поддерживающей возможность использования встраиваемых дополнений (англ. Add-Ins) — плагинов от сторонних разработчиков, что позволяет расширять возможности среды разработки.

Некоторые из наиболее популярных дополнений:

- ReSharper
- Review Assistant - плагин просмотра и редактирования кода
- Visual Assist ИКС
- AnkhSVN — свободная реализация клиента Subversion в Visual Studio (в настоящее время поддерживаются версии с 2005 по 2013).

3 Постановка задачи

3.1 Составление алгоритма

Алгоритм — это понятный и точный порядок действий, изображающий процесс реорганизации объекта из первого состояния в конечное.

Для реализации алгоритма в данном дипломном проекте были учтены все параметры игры. Для всех этапов игры рассмотрены разные алгоритмы.

В этой структуре алгоритма показаны все этапы игры. Например, в долгосрочном этапе покупке проходит в режиме двухстороннего договора. А так же для начало игры в структуре алгоритма внесены такие правила: x (больше или равно) 2 или x (меньше или равно) 8. В остальных двух этапах торги проводится в таком образом: на краткосрочном этапе тире аукционном, а на балансирующим этапе - по подаче заявок. Все эти примеры можно посмотреть на следующем рисунке.

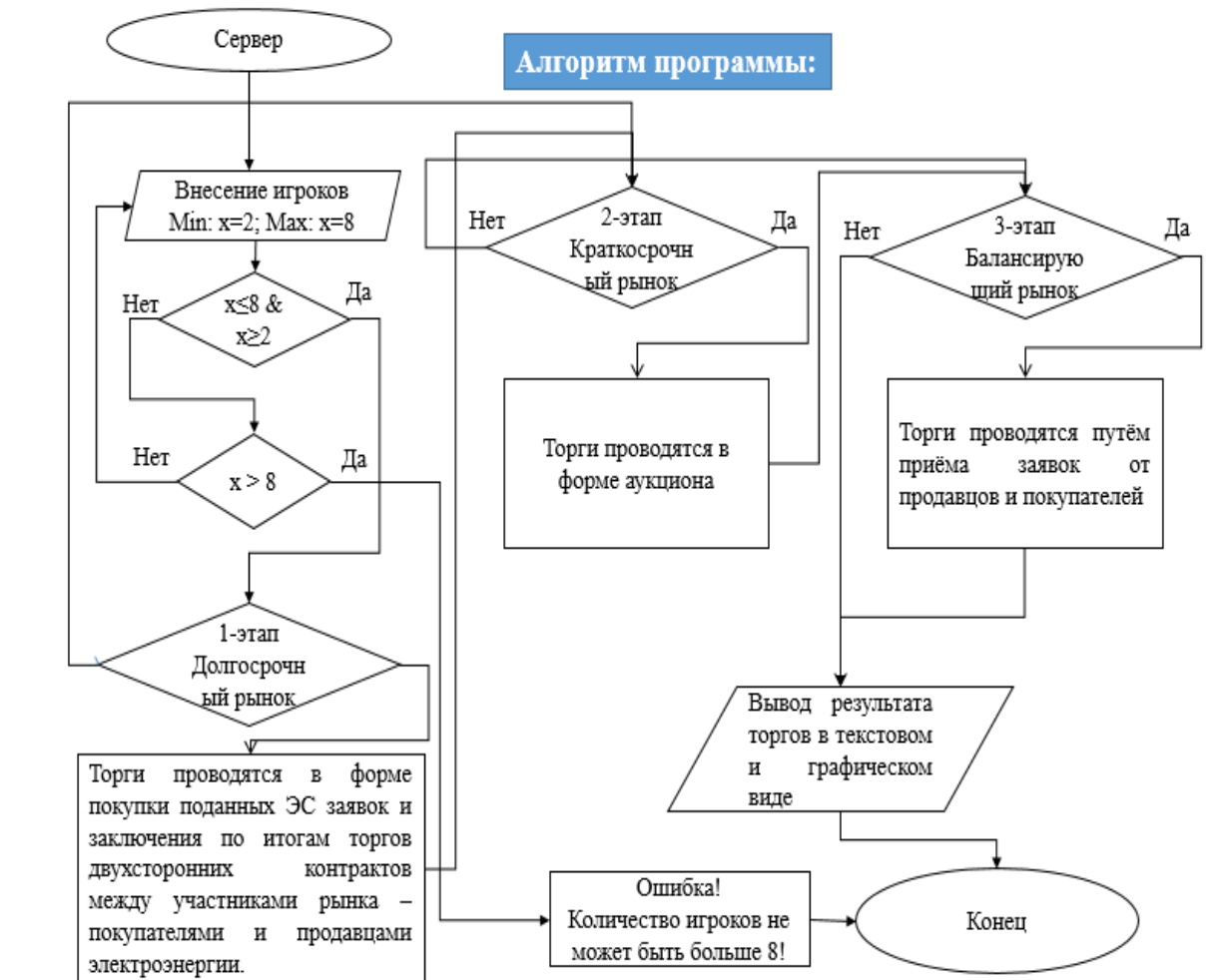


Рисунок тире 8. Алгоритм программы «Разработка деловой игры для рынка электроэнергии»

3.2 Разработка программы для серверной части

Для разработки программы была использовано язык C# в среде Microsoft Visual Studio.

Первом делом создано структура локальный сети для соединение клиентов с серверной части. Был поднять север с ограниченном количеством пользователей. Пользователей были разделены на две категорий: поставщики и потребители.

Для авторизаций пользователи были внесены данные с учетом имени и пароля. Эти данные идентифицируются в серверной части для доступа клиентов находящиеся в одной локальной сети.

Чтобы ступить в начальный этап игры задаются параметры электроэнергии и денег.

На данном рисунке приведены примеры сетевого соединения клиента с сервером.



Рисунок типе 9. Пример сетевого соединения

Интерфейс Серверной части и код в языке C#:

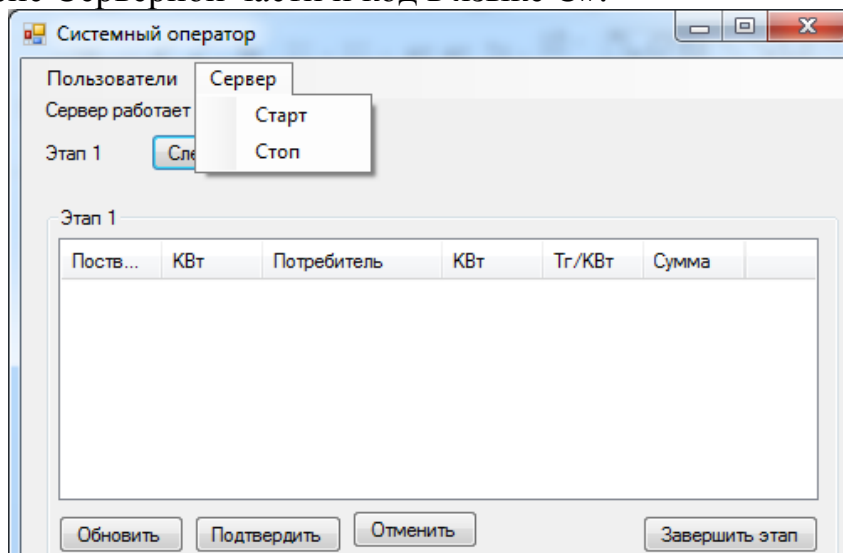


Рисунок типе 10. Интерфейс серверной части

```
using System;  
using System.Collections.Generic;  
using System.Text;  
using System.Net;  
using System.Net.Sockets;  
using System.IO;
```

```

using System.ИКСмл;
using System.ИКСмл.Serialization;

// асинхронный сервер
// http://msdn.microsoft.com/ru-ru/library/dd335942.aspx

namespace Game_Server
{
    class Server
    {
        private Socket _serverSocket;
        private int _port;
        private MyList<Userзнак большой users;
        private MyList<Elementзнак большой buy;
        private MyList<Elementзнак большой sell;
        private MyList<Dealзнак большой deals;
        public Server(int p, MyList<Userзнак большой ul, MyList<Elementзнак
большой bl, MyList<Elementзнак большой sl, MyList<Dealзнак большой dl)
        {
            _port = p;
            users = ul;
            buy = bl;
            sell = sl;
            deals = dl;
            //SetupServerSocket();
        }
        private void SetupServerSocket()
        {
            // Получаем информацию о локальном компьютере
            IPEndPoint localMachineInfo =
                Dns.GetHostEntry(Dns.GetHostName());
            IPEndPoint myEndpoint = new IPEndPoint(
                /*localMachineInfo.AddressMyListскобка прямая0скобка
прямая*/IPAddress.Any, _port);
            // Создаем сокет, привязываем его к адресу
            // и начинаем прослушивание
            _serverSocket = new Socket(
                myEndpoint.Address.AddressFamily,
                SocketType.Stream, ProtocolType.Tcp);
            _serverSocket.Bind(myEndpoint);
            _serverSocket.Listen((int)SocketOptionName.МаиксConnections);
        }

        private class ConnectionInfo

```

```

{
    public Socket Socket;
    public byte[] Buffer;
}

private MyList<ConnectionInfo> _connections =
    new MyList<ConnectionInfo>();

public void Start()
{
    SetupServerSocket();
    for (int i = 0; i < 10; i++)
        _serverSocket.BeginAccept(new
            AsyncCallback(AcceptCallback), _serverSocket);
}
// СТОП
public void Stop()
{
    foreach (ConnectionInfo conn in _connections)
        conn.Socket.Close();

    // _serverSocket.Shutdown(SocketShutdown.Both);
    _serverSocket.Close();
}
private void AcceptCallback(IAsyncResult result)
{
    ConnectionInfo connection = new ConnectionInfo();
    try
    {
        // Завершение операции Accept
        Socket s = (Socket)result.AsyncState;
        connection.Socket = s.EndAccept(result);
        connection.Buffer = new byte[512];
        lock (_connections) _connections.Add(connection);

        // Начало операции Receive и новой операции Accept
        connection.Socket.BeginReceive(connection.Buffer,
            0, connection.Buffer.Length, SocketFlags.None,
            new AsyncCallback(ReceiveCallback),
            connection);
        _serverSocket.BeginAccept(new AsyncCallback(
            AcceptCallback), result.AsyncState);
    }
}

```

```

catch (SocketException еиксс)
{
    CloseConnection(connection);
    //Console.WriteLine("Socket exception: " +
    //    еиксс.SocketErrorCode);
}
catch (Exception еиксс)
{
    CloseConnection(connection);
    //Console.WriteLine("Exception: " + еиксс);
}
}
// читаем входящее сообщение в connection.buffer
private void ReceiveCallback(IAsyncResult result)
{
    string response;
    string[] parts;
    string fl;
    ConnectionInfo connection =
        (ConnectionInfo)result.AsyncState;
    try
    {
        int bytesRead =
            connection.Socket.EndReceive(result);
        if (0 != bytesRead)
        {
            // --- разбор и отправка ответа --- //
            // принятые данные в строку
            response =
System.Text.Encoding.UTF8.GetString(connection.Buffer, 0, bytesRead);
            // разбираем
            parts = response.Split(new char[] { ':' },
StringSplitOptions.RemoveEmptyEntries);
            byte[] data;
            StringBuilder sb;
            switch (parts[0])
            {
                case "auth":
                    {
                        data =
System.Text.Encoding.UTF8.GetBytes(_Auth(parts));
                        connection.Socket.Send(data,
                            data.Length, SocketFlags.None);
                    }
            }
        }
    }
}

```

```

        break;

    case "get":
    {
        data =
System.Текст.Encoding.UTF8.GetBytes(_Get(parts));
        connection.Socket.Send(data,
            data.Length, SocketFlags.None);
    }
    break;

    case "set":
    {
        data =
System.Текст.Encoding.UTF8.GetBytes(_Set(partsскобка прямая1скобка
прямая));
        connection.Socket.Send(data,
            data.Length, SocketFlags.None);
        //////////////////////////////////
    }
    break;
    case "deal":
    {
        data =
System.Текст.Encoding.UTF8.GetBytes(_Deal(parts));
        connection.Socket.Send(data,
            data.Length, SocketFlags.None);

        //////////////////////////////////
    }
    break;

    default:
        break;
    }
    // начинаем новое ожидание
    connection.Socket.BeginReceive(
        connection.Buffer, 0,
        connection.Buffer.Length, SocketFlags.None,
        new AsyncCallback(ReceiveCallback),
        connection);
}
else CloseConnection(connection);
}

```

```

catch (SocketException еиксс)
{
    CloseConnection(connection);
    //Console.WriteLine("Socket exception: " +
    //    еиксс.SocketErrorCode);
}
catch (Exception еиксс)
{
    CloseConnection(connection);
    //Console.WriteLine("Exception: " + еиксс);
}
}
private void CloseConnection(ConnectionInfo ci)
{
    if (ci.Socket == null)
        return;

    ci.Socket.Close();
    lock (_connections) _connections.Remove(ci);
}
// --- реализация протокола --- //
private string _Auth(string скобка прямая скобка прямая message)
{
    string send = "-1";

    if (message.Length != 3)
        return send;
    else
    {
        for (int i = 0; i < users.Count; ++i)
        {
            if ((users.скобка прямая скобка прямая.Name == message.скобка
                прямая1 скобка прямая) && (users.скобка прямая скобка прямая.Password ==
                message.скобка прямая2 скобка прямая))
                return ((int)users.скобка прямая скобка
                прямая.UserRole).ToString();
        }
    }
    return send;
}
private string _Set(string message)
{
    string send = "-1";

```

```

        //Element e = new Element(DateTime.Parse(messageскобка
        прямая1скобка прямая), int.Parse(messageскобка прямая2скобка прямая),
        int.Parse(messageскобка прямая3скобка прямая), int.Parse(messageскобка
        прямая4скобка прямая), messageскобка прямая5скобка прямая);
        Element e;
        ИКCmlSerializer serializer =
new ИКCmlSerializer(typeof(Element));
        // Create a ТекстReader to read the file.
        ТекстReader reader = new StringReader(message);
        // Declare an object variable of the type to be deserialized.
        // Use the Deserialize method to restore the object's state.
        e = (Element)serializer.Deserialize(reader);
        if (e.buy)
            buy.Add(e);
        else
            sell.Add(e);
        //if (messageскобка прямаябскобка прямая == "b")
        //    buy.Add(e);
        //else
        //    sell.Add(e);

        //send = e.id.ToString();
        return send;
    }
private string _Deal(stringскобка прямаяскобка прямая message)
{
    string send = "1";
    if (message.Length != 3)
        return send;
    else
    {
        Deal deal = new Deal(int.Parse(messageскобка прямая1скобка
        прямая), int.Parse(messageскобка прямая2скобка прямая));
        deals.Add(deal);
    }
    return send;
}
private string _Get(stringскобка прямаяскобка прямая message)
{
    string send = "";

    if (message.Length != 2)
        return send;
    else

```

```

{
  if (messageскобка прямая1скобка прямая == "b")
  {
    foreach(Element e in buy)
    {
      send += e.start.ToString() + ":";
      send += e.months.ToString() + ":";
      send += e.kwt_per_h.ToString() + ":";
      send += e.tg_kwt.ToString() + ":";
      send += e.owner + ":";
      send += e.id.ToString() + ";";
    }
  }
  else
  {
    foreach (Element e in sell)
    {
      send += e.start.ToString() + ":";
      send += e.months.ToString() + ":";
      send += e.kwt_per_h.ToString() + ":";
      send += e.tg_kwt.ToString() + ":";
      send += e.owner + ":";
      send += e.id.ToString() + ";";
    }
  }
}
return send;
}
}
}

```

Интерфейс добавление игроков:

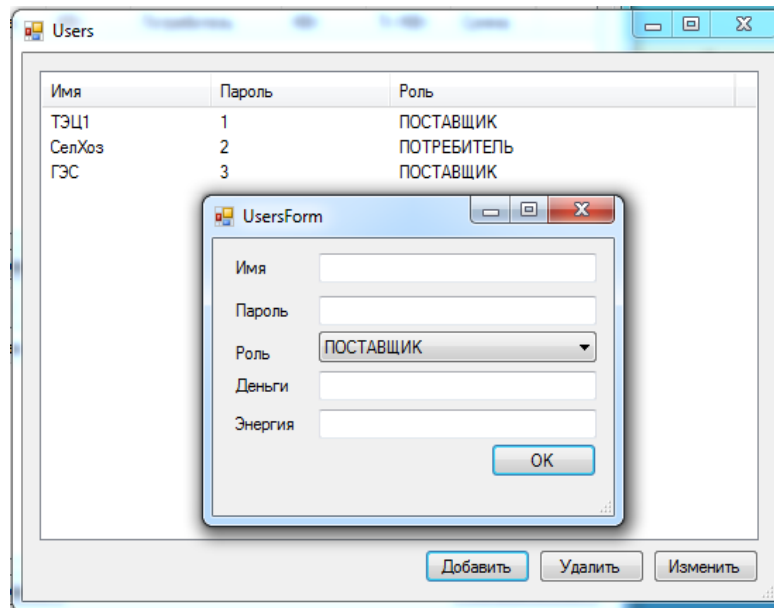


Рисунок тире 11. Добавление игроков

```

namespace Game_Server
{
    partial class UsersForm
    {
        /// <summary>знак большой
        /// Required designer variable.
        /// </summary>знак большой
        private System.ComponentModel.IContainer components = null;

        /// <summary>знак большой
        /// Clean up any resources being used.
        /// </summary>знак большой
        /// <param name="disposing">знак большойtrue if managed resources
should be disposed; otherwise, false.</param>знак большой
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>знак большой
        /// Required method for Designer support - do not modify

```

```

/// the contents of this method with the code editor.
/// </summary>знак большой
private void InitializeComponent()
{
    this.EdtBtn = new System.Windows.Forms.Button();
    this.DelBtn = new System.Windows.Forms.Button();
    this.AddBtn = new System.Windows.Forms.Button();
    this.listView1 = new System.Windows.Forms.ListView();
    this.Имя = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
    this.Пароль = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
    this.Роль = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
    this.SuspendLayout();
    //
    // EdtBtn
    //
    this.EdtBtn.Location = new System.Drawing.Point(447, 337);
    this.EdtBtn.Name = "EdtBtn";
    this.EdtBtn.Size = new System.Drawing.Size(75, 23);
    this.EdtBtn.TabIndex = 0;
    this.EdtBtn.Текст = "ИЗМЕНИТЬ";
    this.EdtBtn.UseVisualStyleBackColor = true;
    this.EdtBtn.Click += new
System.EventHandler(this.EdtBtn_Click);
    //
    // DelBtn
    //
    this.DelBtn.Location = new System.Drawing.Point(366, 337);
    this.DelBtn.Name = "DelBtn";
    this.DelBtn.Size = new System.Drawing.Size(75, 23);
    this.DelBtn.TabIndex = 1;
    this.DelBtn.Текст = "УДАЛИТЬ";
    this.DelBtn.UseVisualStyleBackColor = true;
    this.DelBtn.Click += new
System.EventHandler(this.DelBtn_Click);
    //
    // AddBtn
    //
    this.AddBtn.Location = new System.Drawing.Point(285, 337);
    this.AddBtn.Name = "AddBtn";
    this.AddBtn.Size = new System.Drawing.Size(75, 23);
    this.AddBtn.TabIndex = 2;

```

```

        this.AddBtn.Текст = "Добавить";
        this.AddBtn.UseVisualStyleBackColor = true;
        this.AddBtn.Click += new
System.EventHandler(this.AddBtn_Click);
        //
        // listView1
        //
        this.listView1.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyl
es.Top | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right));
        this.listView1.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[] {
        this.Имя,
        this.Пароль,
        this.Роль});
        this.listView1.FullRowSelect = true;
        this.listView1.Location = new System.Drawing.Point(12, 12);
        this.listView1.Name = "listView1";
        this.listView1.Size = new System.Drawing.Size(510, 319);
        this.listView1.TabIndex = 3;
        this.listView1.UseCompatibleStateImageBehavior = false;
        this.listView1.View = System.Windows.Forms.View.Details;
        //
        // Имя
        //
        this.Имя.Текст = "Имя";
        this.Имя.Width = 120;
        //
        // Пароль
        //
        this.Пароль.Текст = "Пароль";
        this.Пароль.Width = 127;
        //
        // Роль
        //
        this.Роль.Текст = "Роль";
        this.Роль.Width = 244;
        //
        // UsersForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

```

```

        this.AutoScaleMode
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(534, 372);
        this.Controls.Add(this.listView1);
        this.Controls.Add(this.AddBtn);
        this.Controls.Add(this.DelBtn);
        this.Controls.Add(this.EdtBtn);
        this.Name = "UsersForm";
        this.Текст = "Users";
        this.ResumeLayout(false);
    }
#endregion
private System.Windows.Forms.Button EdtBtn;
private System.Windows.Forms.Button DelBtn;
private System.Windows.Forms.Button AddBtn;
private System.Windows.Forms.ListView listView1;
private System.Windows.Forms.ColumnHeader Имя;
private System.Windows.Forms.ColumnHeader Пароль;
private System.Windows.Forms.ColumnHeader Роль;
    }
}

```

3.3 Создание интерфейса и структуры системного оператора

В структуре системного оператора будет отображаться участники, этапы игры, количества поданных заявок, цена и количества электроэнергии. Системный оператор может следить за ходом игры, подтвердить или расторгнуть заявки, полученные от участников. Продолжительность этапа и переход на следующие этапы зависит от системного оператора.

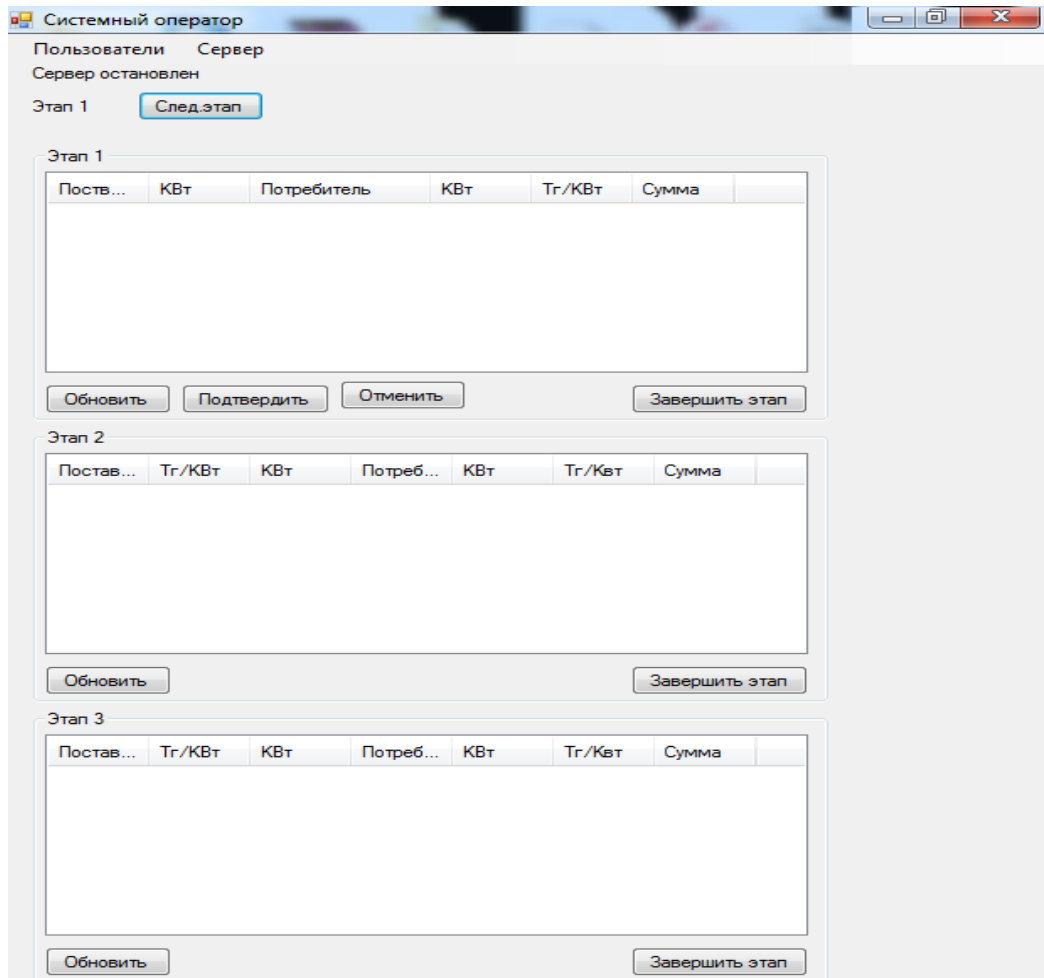


Рисунок типе 12. Интерфейс системного оператора

```

namespace Game_Server
{
    partial class ElementsForm
    {
        /// <summary>знак большой
        /// Required designer variable.
        /// </summary>знак большой
        private System.ComponentModel.IContainer components = null;

        /// <summary>знак большой
        /// Clean up any resources being used.
        /// </summary>знак большой
        /// <param name="disposing">знак большойtrue if managed resources
        should be disposed; otherwise, false.</param>знак большой
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
        }
    }
}

```

```

    }
    base.Dispose(disposing);
}
#region Windows Form Designer generated code
/// <summary>знак большой
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>знак большой
private void InitializeComponent()
{
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.splitContainer1 = new System.Windows.Forms.SplitContainer();
    this.listView1 = new System.Windows.Forms.ListView();
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.заявкиToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
    this.обновитьToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
    this.columnHeader1 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.columnHeader2 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.columnHeader3 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.columnHeader4 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.listView2 = new System.Windows.Forms.ListView();
    this.columnHeader5 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.columnHeader6 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
    this.columnHeader7 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader());
}

```

```

        this.columnHeader8                                     =
((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
        this.columnHeader9                                     =
((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
        this.columnHeader10                                   =
((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
        this.menuStrip1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).BeginInit();
        this.splitContainer1.Panel1.SuspendLayout();
        this.splitContainer1.Panel2.SuspendLayout();
        this.splitContainer1.SuspendLayout();
        this.SuspendLayout();
        //
        // menuStrip1
        //
        this.menuStrip1.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.заявкиToolStripMenuItem });
        this.menuStrip1.Location = new System.Drawing.Point(0, 0);
        this.menuStrip1.Name = "menuStrip1";
        this.menuStrip1.Size = new System.Drawing.Size(654, 24);
        this.menuStrip1.TabIndex = 0;
        this.menuStrip1.Text = "menuStrip1";
        //
        // splitContainer1
        //
        this.splitContainer1.Dock                                     =
System.Windows.Forms.DockStyle.Fill;
        this.splitContainer1.Location = new System.Drawing.Point(0, 24);
        this.splitContainer1.Name = "splitContainer1";
        //
        // splitContainer1.Panel1
        //
        this.splitContainer1.Panel1.Controls.Add(this.listView1);
        this.splitContainer1.Panel1.Controls.Add(this.label1);
        //
        // splitContainer1.Panel2
        //
        this.splitContainer1.Panel2.Controls.Add(this.listView2);
        this.splitContainer1.Panel2.Controls.Add(this.label2);

```

```

this.splitContainer1.Size = new System.Drawing.Size(654, 417);
this.splitContainer1.SplitterDistance = 322;
this.splitContainer1.TabIndex = 1;
//
// listView1
//
this.listView1.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right));
this.listView1.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[] {
this.columnHeader9,
this.columnHeader1,
this.columnHeader2,
this.columnHeader3,
this.columnHeader4});
this.listView1.Location = new System.Drawing.Point(12, 29);
this.listView1.Name = "listView1";
this.listView1.Size = new System.Drawing.Size(307, 376);
this.listView1.TabIndex = 1;
this.listView1.UseCompatibleStateImageBehavior = false;
this.listView1.View = System.Windows.Forms.View.Details;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(13, 13);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(50, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Покупка";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(12, 13);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(53, 13);
this.label2.TabIndex = 0;
this.label2.Text = "Продажа";
//
// заявкиToolStripMenuItem

```



```

//
this.заявкиToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItemскобка прямаяскобка прямая {
this.обновитьToolStripMenuItem});
this.заявкиToolStripMenuItem.Name =
"заявкиToolStripMenuItem";
this.заявкиToolStripMenuItem.Size = new
System.Drawing.Size(57, 20);
this.заявкиToolStripMenuItem.Текст = "Заявки";
//
// обновитьToolStripMenuItem
//
this.обновитьToolStripMenuItem.Name =
"обновитьToolStripMenuItem";
this.обновитьToolStripMenuItem.Size = new
System.Drawing.Size(152, 22);
this.обновитьToolStripMenuItem.Текст = "Обновить";
this.обновитьToolStripMenuItem.Click += new
System.EventHandler(this.обновитьToolStripMenuItem_Click);
//
// columnHeader1
//
this.columnHeader1.Текст = "Предприятие";
//
// columnHeader2
//
this.columnHeader2.Текст = "кВт/ч";
//
// columnHeader3
//
this.columnHeader3.Текст = "Стоимость";
//
// columnHeader4
//
this.columnHeader4.Текст = "Дата начала";
//
// listView2
//
this.listView2.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyl
es.Top | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right));

```

```

        this.listView2.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[] {
        this.columnHeader10,
        this.columnHeader5,
        this.columnHeader6,
        this.columnHeader7,
        this.columnHeader8});
        this.listView2.Location = new System.Drawing.Point(6, 29);
        this.listView2.Name = "listView2";
        this.listView2.Size = new System.Drawing.Size(310, 376);
        this.listView2.TabIndex = 2;
        this.listView2.UseCompatibleStateImageBehavior = false;
        this.listView2.View = System.Windows.Forms.View.Details;
//
// columnHeader5
//
this.columnHeader5.Text = "Предприятие";
//
// columnHeader6
//
this.columnHeader6.Text = "кВт/ч";
//
// columnHeader7
//
this.columnHeader7.Text = "Стоимость";
//
// columnHeader8
//
this.columnHeader8.Text = "Дата начала";
//
// columnHeader9
//
this.columnHeader9.DisplayIndex = 4;
this.columnHeader9.Text = "id";
//
// columnHeader10
//
this.columnHeader10.DisplayIndex = 4;
this.columnHeader10.Text = "id";
//
// ElementsForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

```

```

        this.AutoScaleMode
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(654, 441);
        this.Controls.Add(this.splitContainer1);
        this.Controls.Add(this.menuStrip1);
        this.Name = "ElementsForm";
        this.Текст = "ElementsForm";
        this.menuStrip1.ResumeLayout(false);
        this.menuStrip1.PerformLayout();
        this.splitContainer1.Panel1.ResumeLayout(false);
        this.splitContainer1.Panel1.PerformLayout();
        this.splitContainer1.Panel2.ResumeLayout(false);
        this.splitContainer1.Panel2.PerformLayout();

        ((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).EndInit();
        this.splitContainer1.ResumeLayout(false);
        this.ResumeLayout(false);
        this.PerformLayout();
    }
    #endregion

    private System.Windows.Forms.MenuStrip menuStrip1;
    private System.Windows.Forms.ToolStripMenuItem
заявкиToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem
обновитьToolStripMenuItem;
    private System.Windows.Forms.SplitContainer splitContainer1;
    private System.Windows.Forms.ListView listView1;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.ColumnHeader columnHeader1;
    private System.Windows.Forms.ColumnHeader columnHeader2;
    private System.Windows.Forms.ColumnHeader columnHeader3;
    private System.Windows.Forms.ColumnHeader columnHeader4;
    private System.Windows.Forms.ListView listView2;
    private System.Windows.Forms.ColumnHeader columnHeader5;
    private System.Windows.Forms.ColumnHeader columnHeader6;
    private System.Windows.Forms.ColumnHeader columnHeader7;
    private System.Windows.Forms.ColumnHeader columnHeader8;
    private System.Windows.Forms.ColumnHeader columnHeader9;
    private System.Windows.Forms.ColumnHeader columnHeader10;
}
}

```

4 Техничко–экономическое обоснование

4.1 Краткая характеристика проекта

Цель дипломного проекта «разработка деловой игры рынка электроэнергетики». С внедрением прикладного приложения повысится количество клиентов и спрос на предоставляемые работы. Наиболее важным моментом для разработчика, с экономической точки зрения, является процесс формирования цены. На разработку программного продукта средней сложности обычно требуются весьма незначительные средства.

В данном разделе приводится рассмотрение экономической составляющей реализации данного продукта, отражающее временные, трудовые и финансовые затраты на проект.

4.2 Расчет затрат на разработку информационных технологий

Под информационными технологиями понимаются экономические информационные системы (ЭИС), программные продукты (ПП), информационные базы данных и т.д.

Расчет полных затрат на разработку проектного решения в виде информационных технологий ($C_{\text{пi}}$) осуществляется по формуле

$$C_{\text{пi}} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + A + P_{\text{ми}} + П_{\text{зи}} + P_{\text{ни}}, \quad (4.1)$$

где, $Z_{\text{фот}}$ –тире общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$ –тире отчисления по социальному налогу, тенге;

M_i –тире затраты на материалы, тенге;

A –тире амортизация;

$P_{\text{ми}}$ –тире затраты, связанные с эксплуатацией техники, тенге;

$П_{\text{зи}}$ –тире прочие затраты, тенге;

$P_{\text{нi}}$ тире накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ($Z_{\text{ФОТ}}$) рассчитывается по формуле

$$Z_{\text{ФОТ}} = Z_{\text{oi}} + Z_{\text{di}}, \quad (4.2)$$

где, Z_{oi} тире основная заработная плата, тенге;

Z_{di} тире дополнительная заработная плата, тенге.

Затраты на оплату труда зависят от объема и трудоемкости разработки программного обеспечения.

Общий объем (V_0) программного продукта определяется исходя из количества и объема функции, реализуемых программой

$$V_0 = \sum_{j=1}^n V_i \quad (4.3)$$

где, V_i тире объем отдельной функции ПО;

n тире общее число функций.

Расчет уточненного объема ПО представлен в таблице 4.1.

$$V_0 \approx 10000$$

Таблица 4.1 - Перечень и объем функций программного модуля

НО мер	Наименование (содержание)	Объем функции (LOC)	
		по каталогу V_i	уточнен ный V_{yi}
	2	3	4
01	Генерация структуры базы данных	4300	1000
03	Формирование баз данных	2180	800
04	Обработка наборов и записей баз данных	2670	565
06	Обслуживание баз данных в интерактивном режиме	6950	571
07	Манипулирование данными	9550	3890
08	Организация поиска и поиск в базе данных	5480	1260
07	Графический вывод результатов	480	1292
	Итого:	31610	10000

Общая трудоемкость небольших проектов рассчитывается по формуле

$$T_0 = T_n \cdot K_c \cdot K_m \cdot K_n, \quad (4.4)$$

где, T_n – нормативная трудоемкость;

K_c – коэффициент, учитывающий сложность ПО;

K_m – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

K_n – коэффициент, учитывающий степень новизны ПО.

Посредством коэффициента сложности, показанный в таблице 4.2, учитываются затраты труда, связанные со сложностью ПП скобка прямая4скобка прямая.

В разрабатываемом проекте K_c , за счет наличия у программного модуля одновременно двух характеристик:

- режим работы в реальном времени;
- управление удаленными объектами;

Принимаем $K_c = 0,18$ из таблицы 4.2.

Таблица 4.2 - Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения K_c
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	0,12
4.1. 2 характеристики	0,18
4.2. 3 характеристики	0,26
4.3. Свыше 3тирех характеристик	

Коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_T). Степень использования в разрабатываемом ПО стандартных модулей определяется их удельным весом в общем объеме проектируемого продукта. В данном проекте степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО от 20 процент до 40 процент, следовательно $K_T = 0,8$.

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО (K_H) определяется на основе данных представленных в таблице 4.3 и составляет 1,0.

Таблица 4.3 - Поправочные коэффициенты, учитывающие новизну ПО(K_H).

Категория новизны	Степень новизны	Использование		Значение K_H
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые	+	+	1,75
	ПО, не имеющие доступных	тире	+	1,6
		+		1,2
	аналогов	тире	тире	1,0

Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+ тире +	+ тире тире	1,0 0,9 0,8
В	ПО, являющиеся развитием определенного	тире	тире	0,7
	параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации			

Нормативная трудоемкость ПО (T_H) определяется на основе принятого в расчет V_y и категории сложности, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

В соответствии с этим, согласно укрупненным нормам времени на разработку ПО (T_H) в зависимости от уточненного объема ПО (V_0) и группы сложности: объем ПО (строки исходного кода, LOC) 10000, категория сложности ПО 2тирея тире $T_H = 273$, категория сложности ПО 40.

Следовательно T_0 будет равно:

$$T_0 = 273 \cdot 0.18 \cdot 0.8 \cdot 1 = 39,31 \text{ (чел./дн.)}$$

Численность исполнителей проекта ($Ч_p$) рассчитывается по формуле

$$Ч_p = \frac{T_0}{T_p \cdot \Phi_{эф}} \quad (4.5)$$

где, $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

T_0 – общая трудоемкость разработки проекта (чел./дн.);

T_p – срок разработки проекта (лет).

Срок разработки проекта (T_p) определяется по формуле

$$T_p = \frac{T_0}{\text{Ч}_p \cdot \Phi_{\text{эф}}} \quad (4.6)$$

где, Ч_p – типе плановое число разработчиков.

Эффективный фонд времени работы одного работника ($\Phi_{\text{эф}}$) рассчитывается по формуле

$$\Phi_{\text{эф}} = D_{\text{г}} \text{ типе } D_{\text{п}} \text{ типе } D_{\text{в}} \text{ типе } D_{\text{о}}, \quad (4.7)$$

где, $D_{\text{г}}$ – количество дней в году;

$D_{\text{п}}$ – количество праздничных дней в году;

$D_{\text{в}}$ – количество выходных дней в году;

$D_{\text{о}}$ – количество дней отпуска.

В соответствии с производственным календарем на 2016 год.:

$D_{\text{г}} = 366$; $D_{\text{п}} = 17$; $D_{\text{в}} \text{ типе } 104$; $D_{\text{о}} \text{ типе } 14$,

$\Phi_{\text{эф}} = 366 \text{ типе } 17 \text{ типе } 104 \text{ типе } 14 = 231 \text{ дней}$

Плановое число разработчиков $\text{Ч}_p = 1$, следовательно по формуле (4.6)

$T_p = 39,31 / (1 \cdot 231) = 0,17 \text{ года} = 60 \text{ дней}$.

Таким образом, согласно произведенным расчетам и в соответствии с формулой (4.5)

$\text{Ч} = 39,31 / (0,17 \cdot 231) = 1 \text{ чел}$.

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле

$$Z_{oi} = \sum T_{\text{чи}} \cdot T_{\text{ч}} \cdot \Phi_{\text{п}} \cdot K \quad (4.8)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$T_{\text{чи}}$ – часовая тарифная ставка i тирего исполнителя (тенге);

$\Phi_{\text{п}}$ – плановый фонд рабочего времени i тирего исполнителя (дней), 60 дня;

$T_{\text{ч}}$ – количество часов работы в день (час), 8 часов;

K – коэффициент премирования, составляет 1,2.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов типа исполнителей, участвующих в разработке ПО, с определением образия специальности, квалификации и должности в таблице 4.4.

Таблица 4.4 - Сведения по работникам, задействованным в проекте

Специалист типа Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист	1	120 000
Итого		120 000

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40 типа часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах (Φ_p) учитывая что в месяце 21 рабочий день

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p}, \quad (4.9)$$

где, $T_{\text{ч}}$ – часовая тарифная ставка (тенге);

$T_{\text{м}}$ – месячная тарифная ставка (тенге).

Таким образом

$$\Phi_p = 8 \cdot 21 = 168 \text{ часов}$$

$$T_{\text{д}} = \frac{120\,000}{21} = 5\,714,29 \text{ тенге/день}$$

$$T_{\text{м}} = \frac{120\,000}{21} \cdot 60 = 342\,857,14 \text{ тенге}$$

$$\text{ФОТ} = \left(\frac{120\,000}{21} \right) \cdot 60 \cdot 1,2 = 411\,428,57 \text{ тенге}$$

Социальный налог составляет 11 процент (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$З_{\text{сзи}} = (\text{ФОТ}_{\text{тире}} \text{ПО}) \cdot 11\% \quad (4.11)$$

где ПО тире пенсионные отчисления, которые составляют 10 процент от ФОТ и социальным налогом не облагаются

$$\text{ПО} = \text{ФОТ} \cdot 10\% \quad (4.12)$$

Таким образом,

$$\text{ПО} = 411\,428,57 \cdot 0,1 = 41\,142,857 \text{ тенге}$$

$$З_{\text{сзи}} = (411\,428,57 \text{ тире } 41\,142,857) \cdot 0,11 = 40\,731,428 \text{ тенге}$$

Затрат на материалы определяются по формуле

$$M_i = (З_{\text{осн}} \cdot N_{\text{мз}}) / 100\% \quad (4.13)$$

где $N_{\text{мз}}$ тире норма расхода материалов от основной заработной платы (3 тире 5 процент).

$$M_i = 411\,428,57 \cdot 0,05 = 20\,571,43 \text{ тенге}$$

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{C_{\text{обор}} \cdot N_A \cdot N}{366 \cdot 100} \quad (4.14)$$

где, N_a тире норма амортизации (25 процент);
 $C_{обор.}$ тире первоначальная стоимость оборудования;
 N тире фактический срок эксплуатации оборудования, 60 дней;
 Данные по стоимости оборудования представлены в таблице 4.5.

Таблица 4.5 тире Стоимость оборудования одного ПК с периферией

Наименование	Цена без НДС, тенге
Монитор	32 600
Материнская плата	15 000
Процессор	37 300
Видеокарта	27 500
HDD	10 560
DVDтиреRW	5 720
CPU Cooler	3 200
Оперативная память	8 900
Клавиатура	2 660
Мышь	2 530
Принтер	7 040
MS Visual Studio 2008	бесплатно
Итого:	153 010

$$C_{обор.} = 153\,010 \text{ тенге}$$

Тогда, согласно формуле (4.14)

$$A = \frac{153\,010 \cdot 25 \cdot 60}{366 \cdot 100} = 6\,270,9 \text{ тенге}$$

Расходы по статье «Прочие затраты» (Π_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научнотиретехнической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$\Pi_{zi} = 3_{oi} \cdot N_{пз} / 100, \quad (4.16)$$

где $N_{пз}$ – норматив прочих затрат в целом по организации (20 процент),
 Согласно формуле (4.16)

$$П_{зi} = 411\,428,57 \cdot 0,2 = 82\,285,714 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных(экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу ($H_{рн}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации

$$P_{ni} = Z_{oi} \cdot H_{рн} / 100\%, \quad (4.17)$$

где, P_{ni} - ире накладные расходы на конкретную ПО (тенге);
 $H_{рн}$ - ире норматив накладных расходов по организации (70 процент)
 Согласно формуле (4.17)

$$P_{ni} = 411\,428,57 \cdot 0,7 = 287\,999,99 \text{ тенге}$$

В соответствии с формулой (4.1) расчет полных затрат на разработку прикладного приложения составит

$$C_{ni} = 411\,428,57 + 40\,731,428 + 20\,571,43 + 6\,270,9 + 82\,285,714 + 287\,999,9 = 849\,288,041 \text{ тенге}$$

Сводные результаты расчета затрат на разработку ПО и их структура представлены в таблице 4.6 и на рисунке 4.1.

Таблица 4.6 - Затраты на разработку прикладного приложения для деловой игры рынок электроэнергетики

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{Фот}$	411 428,57	48,44
Социальн.налог	$Z_{сzi}$	40 731,428	4,79

Материалы	M_i	20 571,43	2,42
Амортизация	A	6 270,9	0,74
Прочие затраты	$П_{zi}$	82 285,714	9,7
Наклад. расходы	P_{ni}	287 999,99	33,9
Итого		849 288,041	100,0

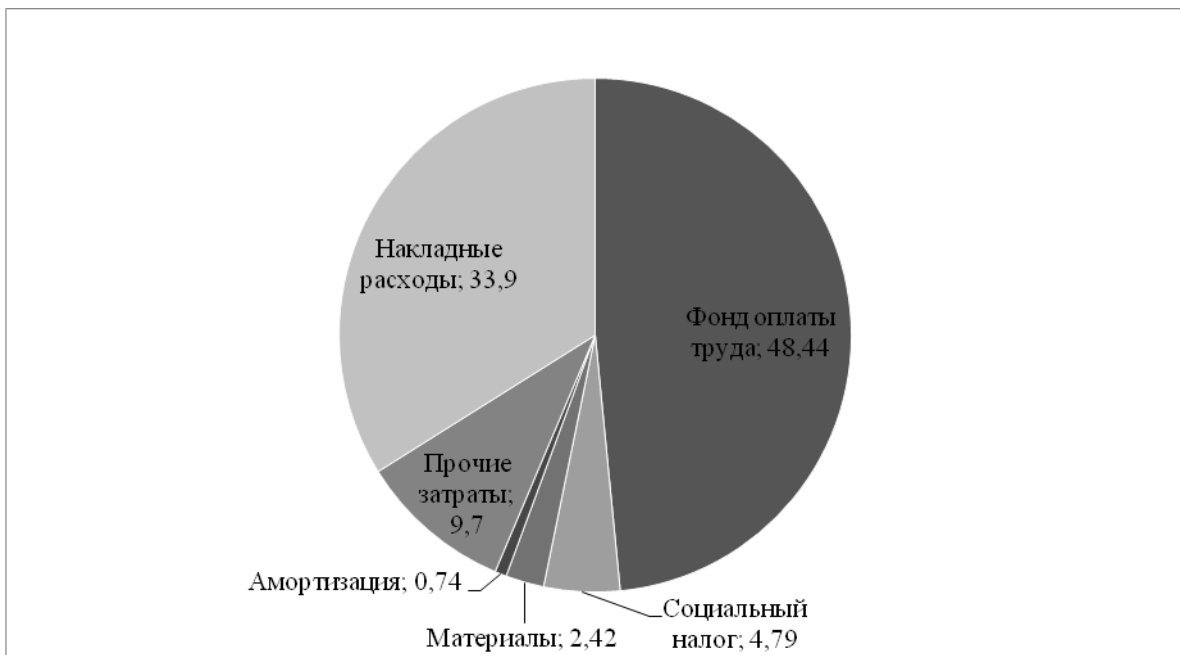


Рисунок 4.1 - Структура затрат на разработку прикладного приложения для деловой игры рынок электроэнергетики

4.3 Расчет цены программного продукта

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$C_{ПП} = Z_{РПР} + П_n + НДС \quad (4.18)$$

где $C_{ПП}$ - цена программного продукта, тенге;

$Z_{РПР}$ - затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_n$ - планируемая прибыль, тенге;

$НДС$ - налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле

$$\Pi_{\text{п}} = Z_{\text{РПР}} \cdot R_{\text{НПП}} \quad (4.19)$$

где $R_{\text{НПП}}$ – нормативная рентабельность ПП, определяемая организацией (20 процент).

НДС, начисленный на ПП, определяется следующим образом

$$\text{НДС} = (Z_{\text{РПР}} + \Pi_{\text{п}}) \cdot k_{\text{НДС}} \quad (4.20).$$

где $k_{\text{НДС}}$ – ставка налога на добавленную стоимость.

Подставляем все значения в формулы (4.18) и (4.20) и получаем:

$$\Pi_{\text{п}} = 849\,288,041 \cdot 0,2 = 169\,857,61 \text{ тенге}$$

Подставив данные в формуле (4.20) получаем

$$\text{НДС} = (849\,288,041 + 169\,857,61) \cdot 0,12 = 122\,297,48 \text{ тенге}$$

Подставив данные в формуле (4.18) получаем

$$Ц_{\text{ПП}} = 849\,288,041 + 169\,857,61 + 122\,297,48 = 1\,141\,433,131 \text{ тенге}$$

5 Безопасность жизнедеятельности

5.1 Анализ потенциально опасных и вредных производственных факторов

Факторы производственной среды могут оказывать вредное воздействие на организм работающего человека.

Пользователь ПЭВМ должен знать о вредном воздействии факторов и об эффективных способах защиты от них, что уменьшает вероятность получения различных профессиональных заболеваний, а также снижает количество сбоев и ошибок при работе с ПЭВМ.

Условия труда пользователя, работающего с персональным компьютером, определяются:

- особенностями организации рабочего места;
- условиями производственной среды (освещением, микроклиматом, шумом, электромагнитными и электростатическими полями, визуальными эргономическими параметрами дисплея и т. д.);
- характеристиками информационного взаимодействия человека и персональных электронно-вычислительных машин.

Работа сотрудников непосредственно связана с компьютером, а соответственно с вредным дополнительным воздействием целой группы факторов, что существенно снижает производительность их труда. К таким факторам можно отнести:

- неправильная освещенность;
- нарушение микроклимата;
- наличие напряжения.

Для существенного уменьшения боли и неприятных ощущений, возникающих у пользователей ПК, необходимы частые перерывы в работе и эргономические усовершенствования, в том числе оборудование рабочего места так, чтобы исключать неудобные позы и длительные напряжения.

Факторы, ухудшающие состояние здоровья пользователей компьютерной техники, можно разделить на следующие категории:

- физические,
- низкочастотные электрические и магнитные поля,
- статическое электричество,
- мягкое рентгеновское, лазерное и ультрафиолетовое излучение,
- повышенная температура,
- ионизация воздуха (повышенное содержание положительных

- аэроионов в воздухе и пониженное содержание отрицательное аэроионов),
- опасное напряжение в электросети,
 - акустический шум,
 - химические,
 - пыль,
 - вредные вещества, выделяемые при работе принтеров и копировальной техники,
 - психофизиологические,
 - напряжение зрения и внимания,
 - интеллектуальные и эмоциональные нагрузки,
 - длительные статические нагрузки и монотонность труда.

Таким образом, разработчик ПП и пользователь испытывают значительную нагрузку на глаза, на мозг, на части тела, что приводит к снижению его трудоспособности к концу рабочего дня.

Согласно ГОСТ 12.1.005-88 ССБТ «Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ», работа людей в помещении относится к работе лёгкой тяжести(1а), так как управление оборудованием осуществляется дистанционно с помощью компьютеров С целью создания нормальных условий для работников предприятий Связи установлены нормы производственного микроклимата. В помещениях при работе с ЭВМ должны соблюдаться следующие климатические условия в таблице 5.1.

Таблица 5.1 - Оптимальные нормы микроклимата помещений с ПК

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, процент	Скорость движения воздуха, м/с
Холодный	Легкая - 1а	22-24	40-60	0,1
	Легкая - 1б	21-23	40-60	0,1
Теплый	Легкая - 1а	23-25	40-60	0,1
	Легкая - 1б	22-24	40-60	0,2

Помещение, в котором находится рабочее место оператора, имеет следующие характеристики:

- Длина помещения: 6 м;
- Ширина помещения: 3,5 м;
- Высота помещения: 3 м;
- Площадь помещения составляет тире 21 м²;
- Число окон: 1, на северной стороне здания;
- Освещение: естественное и общее искусственное;

Помещение находится в здании на 1-м этаже, отдельно стоящего одноэтажного здания рассчитано на 2 рабочих места.

5.2 Требования к микроклимату, ионному составу и концентрации вредных химических веществ в воздухе помещений

На рабочих местах пользователей персональных компьютеров должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПин 2.2.4.548-96. Согласно этому документу для категории тяжести работ 1а температура воздуха должна быть в холодный период года не более 22-

24 °С, в теплый период года 20-25 °С. Относительная влажность должна составлять 40-60 процент, скорость движения воздуха - 0,1 м/с. Для поддержания оптимальных значений микроклимата используется система отопления и кондиционирования воздуха. Для повышения влажности воздуха в помещении следует применять увлажнители воздуха с дистиллированной или кипяченой питьевой водой.

Ионный состав воздуха должен содержать следующее количество отрицательных и положительных аэроионов; минимально необходимый уровень 600 и 400 ионов в 1 см³ воздуха; оптимальный уровень 3 000-5 000 и 1 500-3 000 ионов в 1 см³ воздуха; максимально допустимый - 50 000 ионов в 1 см³ воздуха. Для поддержания оптимального ионного состава воздуха, обеспыливания и обеззараживания воздуха в помещении рекомендуется применять аппараты завода «Диод» серии «Эллион».

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м².

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не должна превышать 40 кд/м². Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 типично 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПО36 с зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5,

ЛСО4, ЛПО34, ЛПО31 с люминисцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении тире линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 °С. Светильники местного освещения должны иметь непросвечивающийся отражатель с защитным углом не менее 40 °С.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

5.3 Расчет освещенности

Искусственное освещение применяется при отсутствии или недостаточности естественного освещения, осуществляется путем использования таких источников света как лампы накаливания, газоразрядные лампы, плоские и щелевые световоды.

Искусственное освещение делят по типу системы освещения:

- Местное - концентрируется световой поток непосредственно на рабочих местах;
- Общее, которое делится на равномерное и локализованное;
- Комбинированное тире совмещение общего и местного освещений.
- Искусственное освещение подразделяется также на:
 - Аварийное, которое применяется при внезапном отключении рабочего освещения (5 процент от общего освещения);
 - Рабочее тире освещение во всех помещениях и на территории, для создания условий нормальной работы;
 - Эвакуационное тире предусматривается в местах, опасных для прохода людей ((больше или равно)0.5 лк тире освещенность в зданиях, 0.2 Лк тире вне их).

Нормирование искусственного освещения производится в соответствии со СН и ПП-4-79, освещенность на рабочих местах нормируется в зависимости от условий выполнения зрительных работ, вида источника света и системы освещения.

Расчет освещения производится для комнаты площадью 21 м² , ширина которой 3.5 м, высота тире 3 м. Воспользуемся методом светового потока.

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = \frac{E \cdot K \cdot S \cdot Z}{n}, \quad (5.1)$$

где, F - рассчитываемый световой поток, Лм;

E - нормированная минимальная освещенность, Лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет E равно 300 Лк при газоразрядных лампах;

S - площадь освещаемого помещения (в нашем случае S равно 21 м²);

Z - отношение средней освещенности к минимальной (обычно принимается равным 1.1-1.2 , пусть Z равно 1.1);

K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение определяется по таблице коэффициентов запаса для различных помещений и в нашем случае K равно 1.5);

n - коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (P_с) и потолка (P_п)), значение коэффициентов P_с и P_п определим по таблице зависимостей коэффициентов отражения от характера поверхности: P_с равно 30 процент, P_п равно 50 процент. Значение n определим по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле:

$$I = \frac{S}{h \cdot (A+B)}, \quad (5.2)$$

где, S - площадь помещения, S равно 21 м²;

h - расчетная высота подвеса, h равно 3 м;

A - ширина помещения, A равно 3.5 м;

B - длина помещения, B равно 6 м.

Подставив значения получим:

$$I = \frac{21}{3 \cdot (6 + 3,5)} = 0,7$$

Зная индекс помещения I, P_с и P_п, по таблице находим n равно 0.38

Подставим все значения в формулу для определения светового потока F:

$$F = \frac{300 \cdot 1,5 \cdot 21 \cdot 1,1}{0,38} = 27355 \text{ Лм}$$

Для освещения выбираем лампы накаливания типа НГ-50, световой поток которых F равно 2800 Лм.

Рассчитаем необходимое количество ламп по формуле:

$$N = \frac{F}{F_{л}}, \quad (5.3)$$

где, N - определяемое число ламп;

F - световой поток, F равно 27355 Лм;

F_л- световой поток лампы, F_л равно 2800 Лм.

$$N = \frac{27355}{2800} = 9 \text{ ламп.}$$

Рассчитаем погрешность: δ процент равно (Φ_л - Φ) / Φ_л равно (27355 - 2800) / 35171 = 0,9 %

Требования к шуму и вибрации в помещениях. Уровни шума на рабочих местах пользователей персональных компьютеров не должны превышать значений, установленных СанПиН 2.2.4/2.1.8.562-96 и составляют не более 50 дБА. На рабочих местах в помещениях для размещения шумных агрегатов уровень шума не должен превышать 75 дБА, а уровень вибрации в помещениях допустимых значений по СН 2.2.4/2.1.8.566-96 категория 3, тип «в».

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

Требования к организации и оборудованию для рабочих мест с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина в тире соответственно 1 600 мм и 1 200 мм. *Рабочая поверхность* стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной в тире не менее 500 мм, глубиной на уровне колен в тире не менее 450 мм и на уровне вытянутых ног в тире не менее 650 мм.

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 °С вниз от горизонтали).

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно-поворотные рабочие стулья с сиденьем и спинкой, регулируемые по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов С и назад до 5 градусов С.;
- высоту опорной поверхности спинки 300 плюс-минус 20 мм, ширину в тире не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 плюс-минус 30 градусов С;
- регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;

- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 плюс-минус 30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим не электризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 °С. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

5.4 Расчет микроклимата рабочей зоны

Микроклимат в рабочей зоне определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также температурой окружающих поверхностей. Повышенная влажность затрудняет теплоотдачу организма путем испарений при высокой температуре воздуха и способствует перегреву, а при низкой температуре, наоборот, усиливает теплоотдачу, способствуя переохлаждению. Оптимальны такие параметры микроклимата, которые при длительном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма без напряжения реакций терморегуляции, что создает ощущение теплового комфорта и служит предпосылкой для высокой работоспособности. Поддержание оптимального микроклимата возможно только в том случае, если предприятие оснащено установкам кондиционирования микроклимата. В остальных случаях следует обеспечивать допустимые микроклиматические условия, т.е. такие, при которых хотя и могут возникать напряжения терморегуляции организма, но не выходят за пределы его физиологических приспособительных возможностей. При этом не возникают нарушения состояния здоровья, но может наблюдаться ухудшение самочувствия и понижение работоспособности.

Обеспечение воздушного комфорта в жилых, общественных и производственных помещениях зависит от систем аспирации, вентиляции и кондиционирования воздуха. Воздушное отопление, совмещенное с вентиляцией, создает в помещении вполне удовлетворительный климат и обеспечивает благоприятные условия воздушной среды. Система кондиционирования воздуха, помимо выполнения задач вентиляции и отопления, позволяет создать благоприятный микроклимат в летний жаркий период года, благодаря использованию в своем составе фреоновой

холодильной машины. Задача кондиционирования воздуха состоит в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, т.е. не замечал влияния этой среды.

Рассчитаем расход воздуха по формуле:

$$L = (Q_{\text{изб}} \cdot 860) / (C \cdot \Delta t \cdot \gamma), \quad (5.1)$$

$$Q_{\text{изб}} = Q_{\text{об}} + Q_{\text{ос}} + Q_{\text{рад}}, \quad (5.2)$$

где C равно $0,24$ ккал/(кг \cdot $^{\circ}\text{C}$) - теплоемкость воздуха;

γ равно $1,206$ кг/м³ - удельная масса приточного воздуха;

$Q_{\text{об}}$ - количество тепла от оборудования;

$Q_{\text{ос}}$ - количество тепла от осветительных устройств

$$Q_{\text{об}} = P_{\text{об}} \cdot \eta \cdot n_{\text{об}}, \quad (5.3)$$

$$Q_{\text{ос}} = \eta N_{\text{ос}} \cdot F, \quad (5.4)$$

где F равно $3,5 \cdot 6$ равно 21 - площадь помещения (м²),

$$Q_{\text{об}} = 21 \cdot 0,5 \cdot 4,75 = 0,049 \text{ кВт.}$$

$$Q_{\text{ос}} = 0,92 \cdot 40 \cdot 21 \cdot 4 = 3,09 \text{ кВт,}$$

Рассчитаем количество тепла от людей

$$Q_{\text{л}} = q_{\text{я}} \cdot n, \quad (5.5)$$

где $q_{\text{я}}$ - тепловыделение явное (Вт);

$Q_{\text{рад}}$ - количество тепла от солнечного излучения

$$Q_{\text{рад}} = m \cdot F \cdot (q_1 + q_2) \cdot B \cdot K_1 \cdot K_2, \quad (5.6)$$

где, m - количество окон;

F_0 - площадь окна;

B - коэффициент тепло пропускания;

K_1 - коэффициент затемнения остекления;

K_2 - коэффициент загрязнения стекла

$$Q_{\text{рад}} = 3 \cdot 3 \cdot (214 + 79 + 73 + 77) \cdot 0,15 \cdot 0,95 \cdot 1,28 = 0,73 \text{ кВт.}$$

$$L = (4,75 + 1,47 + 0,5 + 0,73) \cdot 860 / 0,24 \cdot 1,206 \cdot 8 = 2766,98 \text{ м}^3/\text{час.}$$

Итак за один восьмичасовой рабочий день расход воздуха, для помещения составит 2766,98 м³.

Выбранный кондиционер Fujitsu ASY12USCCW/AOY12USCC ,
Производительность вентилятора (внутр. / внешн.), м3/ч : 540/1700.

5.5 Расчет уровня шума

Одним из неблагоприятных факторов производственной среды в информационно тире вычислительного центра является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников

$$L_{\Sigma} = 10 \lg \left(\sum_{i=1}^n 10^{0.1 L_i} \right) , \quad (5.7)$$

где L_i тире уровень звукового давления i -го источника шума;

n тире количество источников шума.

Полученные результаты расчета сравнивается с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в таблице 5.2.

Таблица 5.2 - Уровни звукового давления различных источников

Источник шума	Уровень шума, дБ
Жесткий диск	42

Вентилятор	42
Монитор	19
Клавиатура	14
Принтер	40
Сканер	41

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу (5,7), получим

$$L_{\Sigma} = 10 \lg(10^{4,2} + 10^{4,2} + 10^{1,9} + 10^{1,4} + 10^4 + 10^{4,1}) = 49.5 \text{ дБ}$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если учесть, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, то эта цифра будет еще ниже. Кроме того при работе принтера непосредственное присутствие оператора необязательно, т.к. принтер снабжен механизмом автоподачи листов.

Заключение

В данном дипломном проекте проанализированы тенденции в развитии электроэнергетики. Сформулированы значимые факторы, обуславливающие становление и развитие региональных отечественных рынков электроэнергии и первоочередные задачи исследования такого рынка (оценка емкости рынка электроэнергии в целом и по основным сегментам, анализ тенденций потребления энергии, прогноз объема продаж и оплаты, определение коэффициентов эластичности спроса в зависимости от основных факторов). Анализировались последние изменения в законодательстве, постановлениях Правительства, документы РК. Выявлены существенные изменения, происходящие на федеральном уровне и проблемы, с которыми сталкиваются региональные энергетические- снабжающие компании. Особо эти проблемы были освещены с точки зрения стоящих задач повышения конкурентоспособности региональных энергокомпаний. Выявлены значимые факторы, обуславливающие становление и развитие региональных рынков электроэнергии и сформулированы первоочередные задачи исследования такого рынка (оценка емкости рынка электроэнергии в целом и по основным сегментам, анализ тенденций потребления энергии, прогноз объема продаж и оплаты, определение коэффициентов эластичности спроса в зависимости от основных факторов).

Для этого была разработана и апробирована совокупность методов решения указанных задач исследования региональных рынков электроэнергии. К ним относятся: регрессионные модели и имитационная балансовая модель для оценки эластичности спроса на электроэнергию; регрессионные модели прогнозирования спроса с выделением шести значимых факторов с приемами уменьшения погрешностей прогнозов; регрессионная модель зависимости тарифов от объема полезного отпуска; модель для построения функции средних и предельных издержек региональной энергосистемы; методика исследования сегмента рынка «Население» с выявлением 13 основных факторов, которые влияют на потребление электроэнергии и ее оплату.

Полученные результаты были далее встроены в систему маркетинговых исследований по прогнозированию спроса в Энерго-сбыте ОАО «Новосибирскэнерго». Коэффициенты эластичности также с успехом были применены при анализе отклика потребителей на рост тарифа. Они служат индикатором устойчивости потребителя из соответствующей отрасли к связанным с тарифами факторами (инфляцией, индексами отраслевых цен и др.).

Создана модель для построения функции средних и предельных издержек региональной энергосистемы. Она позволяет гибко конструировать различные виды тарифов, учитывая поведение спроса в различных диапазонах и удерживать величину тарифа на уровне,

обеспечивающем сохранение основных производственных фондов энергосистемы.

Создана методика исследования сегмента рынка «Население» с выявлением 13 основных факторов, которые влияют на потребление электроэнергии и ее оплату:

Оценка эластичности спроса на электроэнергию была проведена на следующих отраслях Новосибирской области: машиностроение, цветная металлургия, прочие отрасли, железнодорожный транспорт, транспорт (кроме железнодорожного транспорта), связь.

Создан метод определения позиции предприятия-потребителя тепло- и электроэнергии на региональном рынке, базирующийся на учете особенностей расчетов конкретного клиента и использовании среднеотраслевых коэффициентов в качестве индикативных величин.

Разработана методика определения структуры расчетов для потребителя электроэнергии на базе факторного анализа.

Поскольку все предлагаемые методики являются новыми, для их использования предложен комплекс учебных материалов и деловых игр, позволяющий быстро адаптировать их к практической деятельности. Автор обеспечил разработку и реализацию нескольких проектов исследований рынка электроэнергии Новосибирской области, что позволило более точно прогнозировать потребность в электроэнергии, повысить примерно на 30 процент уровень сбора денежных средств, более чем на 8 процент уменьшить дебиторскую задолженность. В итоге это привело к росту эффективности работы, расширило возможности сохранения и развития промышленного потенциала региона и способствовало улучшению социально-экономической обстановки. Ряд методов вошел в состав комплекса учебно-методических средств, которые перспективны с позиций подготовки и переподготовки специалистов в области экономики энергетики.

Список литературы

- 1 М. Эллис, Б. Строуструп. Справочное руководство по языку С# с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
- 2 Стенли Б. Липпман. С# для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
- 3 Бруно Бабэ. Просто и ясно о Borland С# Пер. с англ. - Москва: БИНОМ, 1994. 400с.
- 4 В.В. Подбельский. Язык С#: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
- 5 Ирэ Пол. Объектно-ориентированное программирование с использованием С++: Пер. с англ. - Киев: НИИПФ ДиаСофт Лтд, 1995. 480с.
- 6 Т. Фейсон. Объектно-ориентированное программирование на Borland С# 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
- 7 Т. Сван. Освоение Borland С# 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
- 8 Г. Шилдт. Самоучитель С#: Пер. с англ. - Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.
- 9 У. Сэвитч. С++ в примерах: Пер. с англ. - Москва: ЭКОМ, 1997. 736с.
- 10 К. Джамса. Учимся программировать на языке С#: Пер. с англ. - Москва: Мир, 1997. 320с.
- 11 В.А. Склярков. Язык С# и объектно-ориентированное программирование: Справочное издание. - Минск: Вышэйшая школа, 1997. 480с.
- 12 Х. Дейтел, П. Дейтел. Как программировать на С#: Пер. с англ. - Москва: ЗАО "Издательство БИНОМ", 1998. 1024с.
- 13 А.И. Рофе. Экономика труда: учебник. тире М.: КиноРус, 2010. тире 400 с.
- 14 Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 тире Вычислительная техника и программное обеспечение /Еркешева З.Д., Г.Ш. Боканова. тире Алматы: АУЭС, 2013. тире 40с.
- 15 Матросов А, Сергеев А, Чаунин М. HTML 4.0. - СПб.: ВHV Санкт-Петербург, 2000. - 672 с.

Приложение А

Исходные данные для проведения предварительных расчётов

Для работы с программой должны быть подготовлены исходные данные и проведены предварительные расчёты для определения объёмов и тарифов на электроэнергию. Варианты исходных данных выбираются из таблиц 1-3.

Вариант выбирается по последней цифре номера зачётной книжки.

Т а б л и ц а 1 - Исходные данные

Вариант энергообъединения	Отпущено в	Потреблено отраслями народного хозяйства, процент					Потери в сетях	Топливо	
		Промышлен	Коммунально-бытовое	Сельское	Транспортное	Тра		Цена франко-потребителя, тенге/т.н.т	Теплотворная способность ккал/кг.н.т
	19180	3	9				8	3000	4100
	16380	9	10				9	3035	3750
	14260	3	11				01	3070	4600
	12700	4	12				8	3105	4850
	10970	3	9				8	3140	4098
	15630	3	10	5			9	3175	4600
	17200	3	11				01	3210	4650
	12950	2	10	0			21	3245	3750
	7563	7	9				11	3280	4100

Т а б л и ц а 2 типе Число использования максимумов потребления электрической энергии

В	Число часов использования эл. нагрузки по отраслям
---	--

варианты	Промышленность	коммунально-бытовое хозяйство	Сельское хозяйство	Транспорт
1	6000	3000	3400	4500
2	7000	3500	3600	5000
3	5500	3400	3500	4600
4	5500	3400	3500	4600
5	6900	3400	3500	6000
6	6800	3500	3600	6000
7	7000	3500	3600	6000
8	6500	3500	3600	5500
9	6000	3000	3400	5500

Т а б л и ц а 3 типе Типовые суточные графики электрической нагрузки народного хозяйства (в процент от максимума)

Потребитель	Часы суток	Часы суток						
		-4	-8	-12	2-16	6-20	0-24	2
Промышленность	при: Тпр=5500-6000	2	7	00	0	5	5	4
	Тпр=6100-6500	5	0	00	5	0	0	8
	Тпр=6600-7000	0	0	0	00	5	0	6
Коммунально-бытовая	Тн-б=до 3500	5	5	0	5	00	0	5
	Тн-б=3500 и выше	2	5	0	5	00	0	6
Сельское хозяйство	Тс/ж=3400-3600	0	0	5	0	00	0	6
Транспорт	Ттр=4000-5000	2	7	0	00	5	5	5
	Ттр=5000-6500	2	7	0	00	5	5	4

Приложение Б

Данные для выбора оборудования и расчётов себестоимости производства э/э на ГЭС

Стандартный ряд мощностей генераторов для ГЭС, МВт:
1, 1,6, 2,5, 4, 6,3, 10, 16, 25, 32, 40, 63, 80, 100, 125, 200, 250, 400

Удельные капиталовложения в ГЭС на 1 кВт установленной мощности колеблются в пределах от 1200 до 1500 долларов США в восточных районах, и от 950 до 1200 тире в европейской части страны.

Приложение В
Модель энергосистемы

Т а б л и ц а 1 тире Области применения воздушных ЛЭП переменного тока высокого напряжения

U н, кВ	Передаваемая мощность на цепь, мВт	Длина линии, км	Сечение линии, мм ²
3 5	5 - 15	60 - 30	70-150
1 10	25 - 50	150 - 50	70-240
2 20	100 - 200	250 - 150	240 и выше
3 30	300 - 400	300 - 200	240 и выше
5 00	700 - 900	1200 - 800	300-500
7 50	1800 - 2200	2000 - 1200	400

Т а б л и ц а 2 - Прогнозные тарифы НЭС и РЭК с 2012 года

	2012	2013	2014	2015
Тариф КЕГОС	1,19- 1,98	1,31 тире 2,18	1,44 тире 2,40	1,59 -2,64
Тарифы РЭКов	2,38 тире 4,27	2,86 -4,7	3,15 -5,1	3,5 тире 5,5
Торговая надбавка ЭСО	0,36 тире 0,82	0,38- 0,87	0,40 -0,92	0,42 -0,97

Т а б л и ц а 3 тире Действующие и прогнозные тарифы ЭС РК

Группа	Ба зовый тариф	Предельные тарифы по годам			
		012	013	014	015
1-группа (ЭГРЭС-1,2, ЕЭК)	3, 5	,5	,3	,0	,8

2-группа (ЖГРЭС)	9	5,	,9	,3	,5	,7
3-группа (КарТЭЦ-1-3, Павл ТЭЦ-1,3, УКТЭЦ, Астана-Энергия ТЭЦ 1,2)	6	3,	,9	,4	,9	,5
4-группа (Кар.ГРЭС-2, Пав.ТЭЦ-2, Балхаш, Жезказг. ТЭЦ)	25	3,	,55	,1	,5	,0
5-группа (Петропавл. ТЭЦ, Риддер. ТЭЦ, Рудненс. ТЭЦ)	0	3,	,45	,25	,15	,05
6-группа Согр.ТЭЦ, КарГРЭС-1, Степн. ТЭЦ	51	5,	,9	,1	,2	,3
7-группа (Шымк.ТЭЦ-3, Атырау, Актобе, Жамб.ТЭЦ-4, Кзылорд. ТЭЦ)	7	4,	,3	,7	,0	,3
8-группа (Кентау ТЭЦ, Текел. ТЭЦ, Шахт ТЭЦ, Экиб. ТЭЦ)	8	3,	,98	,6	,2	,5
9-группа (Аркалык. ТЭЦ, Кустан. ТЭЦ, Уральская ТЭЦ)	89	4,	,28	,7	,12	,6
10-группа «АлЭС»	38	4,	,4	,8	,2	,6
11-группа «МАЭК»	23	7,	,23	,23	,23	,83
12-группа ГТЭС (Актурбо, Жанажолская)	1	4,	,0	,7	,4	,8
13-группа ГЭС (Бухтарм., Усть-Каменог., Шульбинская, Шардар.)	7	2,	,63	,9	,3	,5

Приложение Г

Подготовка исходных данных и предварительные расчёты

Предварительная обработка задания и расчёт исходных данных для дальнейшей работы с моделью рынка производится по описанной ниже методике.

Г.1 Определение годовой потребности в э/э

Суммарная годовая потребность в электрической энергии на планируемый год в установившейся энергосистеме определяется на основе:

- фактического электропотребления абонентами системы за отчетный год;
- удельного веса групп потребителей в общем годовом потреблении электрической энергии;
- годового прироста электропотребления абонентами энергосистемы;
- потерь э/э в сетях.

$$\mathcal{E} = \mathcal{E}_{п.о.} + \mathcal{E}_{сет}, \text{ МВт}\cdot\text{ч/год}, \quad (1)$$

где $\mathcal{E}_{п.о.}$ - планируемый полезный отпуск электрической энергии по отраслям;

$\mathcal{E}_{сет}$ - потери э/э в сетях.

Формулу (1) поэтому можно написать в виде:

$$\mathcal{E} = \mathcal{E}_{пр} + \mathcal{E}_{кб} + \mathcal{E}_{сх} + \mathcal{E}_{тр} + \mathcal{E}_{сет}, \quad (2)$$

где $\mathcal{E}_{пр}$ - тире промышленности;

$\mathcal{E}_{кб}$ - тире коммунально-бытовым потребителям;

$\mathcal{E}_{сх}$ - тире сельскохозяйственному производству;

$\mathcal{E}_{тр}$ - тире электрифицированному транспорту.

Необходимые исходные данные для определения планируемого полезного отпуска в производстве э/э приведены в приложении А.

Г.2 Построение суточных графиков электрической нагрузки энергосистемы

Суточные графики электрической нагрузки для зимнего (декабрь) и летнего (июль) дня по каждой отрасли народного хозяйства строятся в процентах от годового максимума

Годовой (зимний) суточный максимум электрической нагрузки для каждой отрасли определяется:

$$P_{max}^{omp} = \frac{\mathcal{E}^{omp}}{h_{max}^{omp}}, \quad (3)$$

где \mathcal{E}^{omp} - величина годового электропотребления данной отрасли;

h_{max}^{omp} - годовое число часов использования максимумов электрической нагрузки для данной отрасли.

Летний максимум электрической нагрузки обычно принимают равным 75 процент от зимнего максимума. Типовые суточные графики нагрузки приведены в приложении А, таблица 3.

На основе типовых отраслевых суточных графиков строится (в табличной и графической формах) суммарный суточный зимний график электрической нагрузки энергосистемы. Для пикового значения суммарного зимнего суточного графика нагрузки привести значения потребляемой мощности по отраслям (для этого интервала времени) и их процент (от суммарного пика). Полученные значения занести в таблицу 1.

Т а б л и ц а 1 тире Нагрузка по отраслям в период зимнего максимума

Отрасль	Мощность нагрузок, МВт	проце нт от суммы
Промышленность	565	31
Коммунально-бытовое хозяйство	688	38
Сельское хозяйство	479	27
Транспорт	68	4

Σ	1801	100
----------	------	-----

Г.3 Определение максимальной величины мощности электростанций энергосистемы и выбор генераторов э/э на различных типах станций для покрытия нагрузки региона

С целью покрытия графиков электрической нагрузки потребителей принимаем в энергообъединении (ЭО) три типа электростанций. Покрытие пиковой нагрузки осуществляется ГЭС, число часов их работы лежит в пределах 2500-3000 часов в год. Мощность ТЭЦ принимают в соответствии с уровнем промышленной нагрузки. Для случая, когда величина промышленной нагрузки менее 40 процент, мощность ТЭЦ составляет 20-25 процент общей мощности ЭО, а в случае, если нагрузка превышает 50 процент, мощность ТЭЦ принять 25-35 процент. Остальная часть требуемой мощности покрывается КЭС.

Общее количество ЭС рекомендуется принять равным не менее чем 6 или 7. Из них 2 ГЭС, 3 (4) ТЭЦ, 2 КЭС. Четыре ТЭЦ принимаем в случае, если после распределения требуемых мощностей по типам электростанций выяснится, что на долю ТЭЦ приходится **более 1200 МВт**. При этом мощность отдельных станций выбирается, исходя из следующего соотношения:

40 ПРОЦЕНТ -45 ПРОЦЕНТ суммарной мощности, вырабатываемой электростанциями определённого типа, должно вырабатываться на одной станции, 55 ПРОЦЕНТ -60 процент - на оставшейся (или в сумме на оставшихся типе для ТЭЦ).

При выборе единичной мощности агрегатов для тепловых электростанций необходимо ориентироваться на современные энергоблоки и турбоагрегаты мощностью не менее 50 МВт. При этом на долю одного агрегата должно приходиться 40 процент -45 процент мощности всех энергопроизводящих агрегатов станции, а 55 процент -60 процент - на долю оставшихся энергоагрегатов.

Выбранное оборудование свести в таблицу 2, расписав подробно по каждому блоку каждой ЭС.

Т а б л и ц а 2 - Состав и мощность оборудования электростанций

Электростанции и блоки	Установленная мощность
ГЭС	340
ГЭС 1	140
блоки 1 и 2	2x40
блок 3	1x60
ГЭС 2	200
блоки 1 и 2	2x60
блок 3	1x80
ТЭЦ	1655

ТЭЦ 1	495
блоки 1 и 2	2x180
блок 3	1x135
ТЭЦ 2	495
блоки 1 и 2	2x180
блок 3	1x135
ТЭЦ 3	665
блоки 1 и 2	2x250
блок 3	1x165
КЭС	2280
КЭС 1	680
блоки 1 и 2	2x250
блок 3	1x180
КЭС 2	1600
блоки 1 и 2	2x500
блок 3	2x300
$\sum N_{уЭС}$ всех ЭС	4275

Мощность турбин для ГЭС выбирается из приведённого в приложении Б стандартного ряда мощностей генераторов, а установленные на ТЭС турбины выбираются из таблиц 1-6 приложения Б.

При определении суммарной мощности станции необходимо учесть расход э/э на собственные нужды электростанций. Для ТЭЦ расход на СН принять 5-10 ПРОЦЕНТ от $N_{уст.}$ станции; для КЭС тире 3-5 процент; для ГЭС тире 1-2 процент.

Структуру и состав основного оборудования электростанций принимать из следующих требований норм технологического проектирования станций:

а) выдачу мощности станциями осуществлять не менее чем от двух открытых распределительных устройств (ОРУ) различного класса напряжения.

б) распределение нагрузки между ОРУ принять:

1) 60-65 процент от установленной мощности электростанции ($N_{уст.}$) тире для покрытия нагрузки потребителей РЭК (класс напряжения тире до 110 кВ);

2) 35-40процент тире передача э/э в направлении связи с энергосистемой (в НЭС) через второе ОРУ на напряжении 220 кВ и выше.

В этом разделе следует привести теоретические принципы экономического распределения электрической нагрузки между электростанциями энергосистемы с учетом энергоэкономических характеристик различных типов электростанций. В основе принять, что в отдельных интервалах времени совмещенного суточного графика нагрузки энергосистемы предпочтение отдается той или иной электростанции на основе критерия минимума расхода топлива на единицу выработки э/э.

При проектировании энергосистем аварийный резерв мощности размещается на ГЭС или крупных маневренных КЭС и ГТЭС. В данной работе резерв мощности размещаем на ГЭС.

Установленная мощность ЭО определяется по формуле:

$$N_y = \sum N_{расп} + N_{рез}, \quad (4)$$

где $N_{рез}$ тире суммарный энергетический резерв мощности, сконцентрированный на ГЭС ЭО, МВт.

В расчёте, в целях упрощения, принимаем величину резерва мощности равной 15 процент суммарной мощности ГЭС. Суммарный энергетический резерв мощности ЭО составит:

$$N_{рез} = 0,15 \cdot N_{расп.ГЭС}. \quad (5)$$

Резервная мощность не участвует в покрытии общего графика нагрузок, но используется в покрытии пиковой нагрузки, так как вероятность выхода из строя основного генерирующего оборудования в часы пика мала и поэтому не учитывается в расчётах. Это необходимо учитывать в дальнейшем. Тогда суммарная мощность ГЭС составит:

$$N_{ГЭС} = 1,15 N_{расп}. \quad (6)$$

Г.4 Долевое участие электростанций в покрытии графика нагрузок и построение суточных рабочих графиков ЭПО. Определение отпускных цен и прогнозных тарифов на услуги по передаче и распределению электроэнергии в НЭС и РЭК

На основании выбранного энергопроизводящего оборудования электростанций необходимо рассчитать долевое участие D каждой электростанции в покрытии суточного графика нагрузок для зимнего максимума и летнего минимума.

$$D_{ГЭС} = \sum N_{ГЭС} \cdot d - N_{сн}, \quad (7)$$

где $\sum N_{ГЭС}$ тире суммарная мощность ГЭС ЭО;

d тире доля данной ГЭС в сумме мощностей ГЭС ЭО. $d = N_{ГЭС} / \sum N_{ГЭС}$.

$N_{сн}$ тире мощность, расходуемая на собственные нужды электростанции. Для ТЭЦ и КЭС собственные нужды в рабочем графике не учитываем.

При расчётах нужно учитывать, что полная мощность ГЭС используется в период пиковой нагрузки, в остальное время ГЭС выдаёт электроэнергию, вырабатываемую при обязательном техническом пропуске воды из водохранилищ тире 10 процент от $N_{ГЭС}$.

$$D_{ТЭЦ} = \sum N_{ТЭЦ} \cdot d - N_{сн}, \quad (8)$$

где $\sum N_{ТЭЦ}$ тире суммарная мощность ТЭЦ ЭО;

d тире доля данной ТЭЦ в сумме мощностей ТЭЦ ЭО.

$N_{сн}$ тире мощность, расходуемая на собственные нужды электростанции.

$$D_{КЭС} = (N_{наг} - \sum D_{ГЭС} - \sum D_{ТЭЦ}) \cdot d, \quad (9)$$

где $N_{наг}$ тире мощность нагрузки в данный момент времени;

d типе доля данной КЭС в сумме мощностей КЭС ЭО.

Суммарная мощность, выдаваемая всеми станциями, должна соответствовать мощности нагрузки для рассчитываемого периода времени. Таким образом, рассчитанная мощность соответствует мощности, выдаваемой с шин электростанций в сети РЭКов и НЭСа по суточному графику нагрузок.

Результаты расчёта суточных рабочих графиков свести в таблицу 3.

Т а б л и ц а 3 - Суточные рабочие графики электростанций

Станции / Тип	Доля от мощности всех ЭС этого типа, отн. е.	-4 ас.	-8 ас.	-12 час.	2-16 ас.	6-20 ас.	0-24 ас.	Средне- суточная в час, МВт	сутки, МВт	В сезон, М Вт
ЭС1 Г	0,648	4	4	4	4	41	4	49	8	54475
ЭС2 Г	0,352					7		60	4	3752
ЭС Σ Г	1	2	2	2	2	18	2	309	1	38227
ЭЦ1 Т	0,206	2	2	2	2	2	2	208	2	01856
ЭЦ2 Т	0,206	2	2	2	2	2	2	208	2	01856
ЭЦ3 Т	0,588	62	62	62	62	62	62	293	6	145290
ЭЦ Σ Т	1	46	46	46	46	46	46	0709	1	949002
ЭС1 К	0,605	6	10	75	58	87	42	1071	1	014837
ЭС2 К	0,395	3	02	41	29	49	23	227	7	315241
ЭС Σ К	1	58	12	116	087	136	65	8297	1	330078
мощность Σ		26	80	584	555	801	033	0315	3	517307

На основании таблицы построить суммарный зимний суточный рабочий графики с отражением долевого участия **каждого** типа электростанций.

Г.5 Расчет себестоимости производства э/э на различных электростанциях, по типам станций

Полная себестоимость э/э - это годовые издержки энергосистемы по производству $U_{ст}$, оказанию услуг по передаче э/э $U_{сет.}$ и внепроизводственных затрат $U_{пр.}$

$$U = U_{ст} + U_{сет.} + U_{пр.} \text{ тыс. тенге/год.} \quad (12)$$

Годовые издержки электростанций определяются как сумма издержек по отдельным электростанциям, учитывающих расходы за топливо U_T , заработную плату $U_{з.п.}$, амортизацию основных фондов U_a , текущий ремонт $U_{т.р.}$, общестанционные нужды $U_{общ.}$:

$$U_{ст.} = \Sigma (U_T + U_{з.п.} + U_a + U_{т.р.} + U_{общ.}) \text{ тыс. тенге/год.} \quad (13)$$

Издержки топлива на тепловых ЭПО определяются как:

$$U_T = \mathcal{E}_{о.с.}^T \cdot V_{уд} \cdot (7000/Q_{п.}^H) \cdot C_T \text{ тыс. тенге/год,} \quad (14)$$

где $\mathcal{E}_{о.с.}^T$ - полезный отпуск э/э электростанциями, кВт·ч

$Q_{п.}^H$ - теплотворная способность топлива, Ккал/кг (приложение 1);

C_T - цена тонны топлива франко-потребителя, тенге/т;

$V_{уд}$ - средний удельный расход условного топлива на единицу выработанной ЭЭ электростанциями, г. у.т./кВт·ч.

При определении $V_{уд}$ для КЭС учитывают удельные расходы топлива с учётом мощности блоков. Учитывая, что средний КПД КЭС лежит в диапазоне 0,35-0,4, по выражению (15) результат получим в кг условного топлива на 1 кВт·ч:

$$v_{уд} = 0,124/\eta_{КЭС}. \quad (15)$$

При расчётах необходимо учитывать так же, что КПД электростанции снижается с уменьшением единичной мощности блока. Поэтому для станции с блоками мощностью 60 МВт КПД нужно принимать меньшим, чем для станции с блоками мощностью 210 МВт.

Поскольку КЭС работает в режиме регулирования графика нагрузки, то, соответственно, с течением суток КПД КЭС изменяется. Таким образом, в зоне провала КПД КЭС принимать равным 0,25 - 0,3, а в пике тире 0,37 - 0,43

Для ТЭЦ КПД принять равным 0,45 - 0,5 вне зависимости от времени суток тире ТЭЦ покрывает базовую часть графика нагрузок и работает в условно-постоянном режиме выдачи мощности. Удельный расход топлива по типу турбин приведён в приложении.

С окончанием отопительного сезона КПД ТЭЦ, из-за прекращения отопительного отбора, снижается (принять снижение кпд равным 15 процент). Так же, вследствие общего снижения электрической нагрузки в летний период, КПД тепловых электростанций падает ещё на 25 процент.

Ниже, в целях общего ознакомления, приведены расчётные формулы по определению других составляющих издержек ЭПО (формулы 16-21), в данной работе не использующиеся, за исключением формулы 17 для ГЭС.

Издержки на заработную плату персонала станции $U_{з.п.}$ определяются:

$$U_{з.п.} = n_{шт} \cdot N_y \cdot Z_{ср}, \text{ тыс. тенге/год,} \quad (16)$$

где $n_{шт}$ - штатный коэффициент для данной электростанции, чел·МВт (приложение 8);

N_y - установленная мощность электростанции, МВт;

$Z_{ср}$ - среднегодовой фонд заработной платы одного человека промышленно-производственного персонала, 35000-65000тенге/мес.

Издержки на амортизацию основных фондов U_a на электростанциях:

$$U_a = \sum (d_i/100) \cdot K_i, \text{ тыс. тенге/год,} \quad (17)$$

где d_i - средневзвешенная норма амортизации, процент (для КЭС тире 5,8-7,1, для ТЭЦ тире 6,7-7,9, для ГЭС тире 2-4);

K_i — стоимость основных производственных фондов электростанции, тыс.тенге/год.

Затраты на текущий ремонт $U_{т.р.}$ принимаются

$$U_{т.р.} = (0,1-0,2) \cdot U_a, \text{ тыс. тенге/год.} \quad (18)$$

Общестанционные издержки $U_{общ.}$ определяются по выражению

$$U_{общ.} = (0,16-0,32)(U_{з.п.} + U_a + U_{т.р.}), \text{ тыс. тенге/год.} \quad (19)$$

Издержки на амортизацию и обслуживание электросетей, приближенно оцениваемые пропорционально стоимости основных производственных фондов электрических сетей:

$$U_{сет} = \sum ((d_{сi}/100) \cdot K_{в.л.}), \text{ тыс. тенге/ год,} \quad (20)$$

где $d_{сi}$ - норма ежегодных издержек на амортизацию и обслуживание сетей, принимаем 5,0-7,0 процент.

Внепроизводственные общестанционные издержки определяются как:

$$U_{пр.} = (0,017 - 0,025) \cdot (U_{с.т.} + U_{сет}), \text{ тыс. тенге/ год.} \quad (21)$$

Для упрощения расчётов примем, что на ТЭС топливная составляющая затрат лежит в пределах 50-60 процент. Тогда остальные составляющие издержек примем равными 50-40 процент соответственно.

Для ГЭС основная составляющая издержек тире амортизация (формула номер 17), которую примем равной 40-50 процент. Тогда, найдя амортизацию, можно будет найти и остальные издержки, равные, соответственно, 60-50 процент.

Таким образом, полная (коммерческая) себестоимость полезно отпущенного киловатт-часа на электростанции определится как:

$$S = U/\mathcal{E}_{п.о.} \text{ тенге/кВт·ч.} \quad (22)$$

При этом э/э, потребляемая на СН электростанций, не входит в издержки.

Отпускная цена э/э с шин станции, с учётом доходности 10 процент, вычисляется как:

$$Ц_{ЭС} = (S_{э.с.} + 0,1 \cdot S_{э.с.}). \quad (23)$$

Необходимо рассчитать себестоимость и отпускную цену 1 кВтч э/э для всех ЭС. Результаты расчётов свести в таблицу 4.

Т а б л и ц а 4 - Прогнозные отпускные цены с шин ЭС.

Отпускная цена	Ц _{о.т.} тенге/кВт·ч
ГЭС 1	10,71
ГЭС 2	9,7
ТЭЦ1	4,3
ТЭЦ2	5,2
ТЭЦ3	5,2
КЭС1	5,6
КЭС2	6

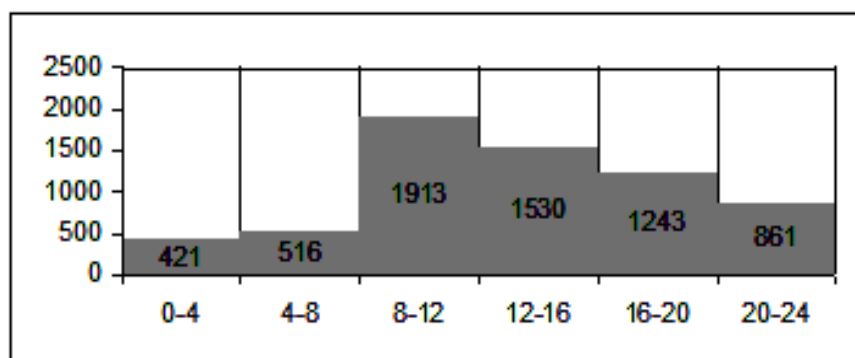
Г.6 Прогнозные тарифы на услуги по передаче и распределению э/э

Субъектами рынка э/э РК являются НЭС («КеGOK») и РЭК, оказывающие услуги по передаче и/или распределению э/э. Прогнозные тарифы на оказание услуг по передаче э/э в НЭС на среднесрочный период принять в соответствии с данными, приведёнными в таблице 4. Указанные тарифы включают покрытие затрат по диспетчеризации в сетях и расходы, связанные с балансированием мощности. В рассматриваемой модели прогнозирования конечных тарифов у потребителей учитываются и тарифы на оказание услуг по передаче и распределению э/э в РЭК. Количество РЭКов соответствует количеству функционирующих в регионе ЭПО. Прогноз изменения тарифов РЭКов принять в соответствии с таблицей 4 приложения А.

Приложение Д
Зимние суточные графики нагрузки

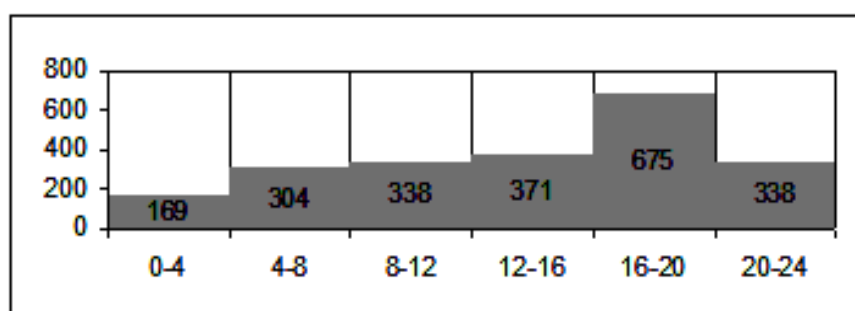
Промышленность

Часы	МВт
0-4	421
4-8	516
8-12	1913
12-16	1530
16-20	1243
20-24	861
Итого в сутки	25934



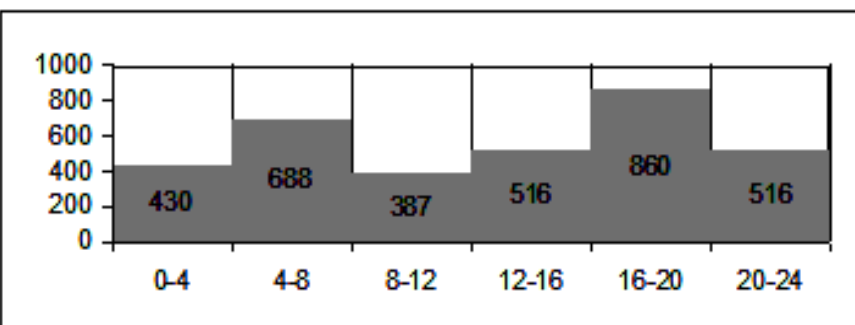
Коммунально-бытовое хозяйство

Часы	МВт
0-4	169
4-8	304
8-12	338
12-16	371
16-20	675
20-24	338
Итого в сутки	8775



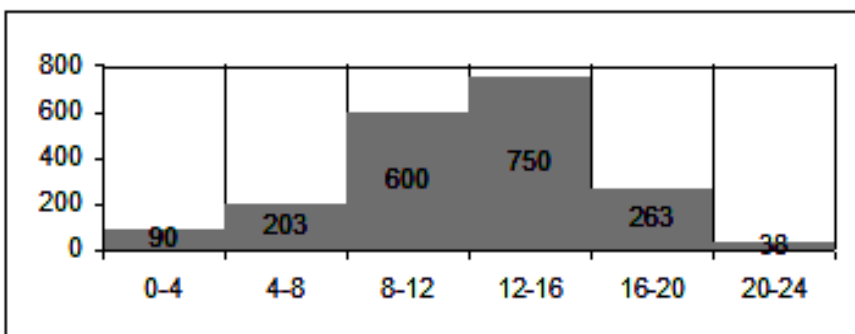
Сельское хозяйство

Часы	МВт
0-4	430
4-8	688
8-12	387
12-16	516
16-20	860
20-24	516
Итого в сутки	13593



Транспорт

Часы	МВт
0-4	90
4-8	203
8-12	600
12-16	750
16-20	263
20-24	38
Итого в сутки	7770



Продолжение приложения Д

Суммарный суточный график потребления (зимний) (с учётом потерь)

Часы	МВт
0-4	1243
4-8	1916
8-12	3626
12-16	3548
16-20	3406
20-24	1962
Итого в сутки	62800
Итого в сезон	11429543

Потребление по отраслям для зимнего суточного максимума

Отрасль	Мощность	% от суммы
Пром	1392	41%
Ком-быт	756	22%
Сельхоз	964	28%
Транспорт	294	9%
Σ	3406	100%



4 Определение мощности и выбор энергопроизводящего оборудования электростанций энергосистемы

В нашей расчетно-графической работе участвуют три типа электростанций: ГЭС, ТЭЦ и КЭС. ГЭС покрывает только пиковую нагрузку. Для определения покрытия пиковой нагрузки ГЭС воспользуемся формулой:

$$N_{ГЭС} = 1,15 \cdot N_{вер} \cdot 1,1 \cdot 1,015, \quad (3)$$

где $N_{вер}$ – разность между часовым максимумом нагрузки и следующей ступени нагрузки по суммарному графику нагрузки;

1,15 – резервная мощность;

1,015 – собственные нужды;

1,1 – 10% мощности ГЭС вырабатывает при обязательном спуске воды.

Мощность ТЭЦ берем как 30% от часового максимума нагрузки по графику, т.к. величина промышленной нагрузки превышает 50%.

КЭС покрывает остальную часть требуемой мощности.

Продолжение приложения Г

4 Определение максимальной величины мощности электростанций энергосистемы и выбор генераторов электроэнергии на различных типах станций для покрытия нагрузки региона

Мощность с учётом резерва и расхода на собственные нужды

Тип станций	СН, в %	Общая N	С учётом СН	С резервом (15 от N)
ГЭС	2	220	224	258
ТЭЦ	8	851	920	
КЭС	4	2335	2428	
Располагаемая мощность ЭО		3406	3572	3605

Выбор оборудования

Электростанции и блоки	Установленная мощность	Стандартный ряд мощностей генераторов (приложение Е):
ГЭС		
ГЭС №1	166	для ГЭС, МВт: 1, 1,6, 2,5, 4, 6,3, 10, 16, 25, 32, 40, 63, 80, 100, 125, 200, 250, 400
блоки 1-2	126	
блок 3	40	
ГЭС №2	104	
блоки 1-2	64	для КЭС и ТЭЦ_1, МВт: 25/30, 50/60, 110/120, 180/210, 250/300
блок 3	40	
Суммарная N ГЭС	270	
Суммарная N ГЭС за вычетом резерва	234,8	для ТЭЦ и КЭС_2, МВт 25/30, 50/60, 60/75, 80/100, 135/165
ТЭЦ		
ТЭЦ №1	260	
блоки 1-2	150	
блок 3	110	
ТЭЦ №2	260	
блоки 1-2	150	
блок 3	110	
ТЭЦ №3	450	
блоки 1-2	270	
блок 3	180	
Суммарная N ТЭЦ	970	
КЭС		
КЭС №1	1350	
блоки 1, 2, 3	750	
блок 4, 5	600	
КЭС №2	1100	
блоки 1, 2	500	
блок 3, 4	600	
Суммарная N КЭС	2450	

Суммарная N ЭО, МВт (без резерва).	3655
------------------------------------	------

Требуемое распределение по станциям	1 станция	2 станция	3 станция
	40%	30%	30%
Требуемое распределение по блокам	3 блока	3 блока	3 блока
	1x40%	1x40%	1x40%
	2x30%	2x30%	2x30%

Продолжение приложения Д

5 Определение долевого участия электростанций в покрытии графика нагрузок и построение суточных рабочих графиков ЭПО

Долевое участие для ГЭС определяется как:

$$D_{ГЭС} = \Sigma N_{ГЭС} \cdot d - N_{св}, \quad (4)$$

где $\Sigma N_{ГЭС}$ – суммарная мощность ГЭС ЭО;

d – доля данной ГЭС в сумме мощностей ГЭС ЭО. $d = N_{ГЭС} / \Sigma N_{ГЭС}$;

$N_{св}$ – мощность, расходуемая на собственные нужды электростанции.

Для ТЭЦ:

$$D_{ТЭЦ} = \Sigma N_{ТЭЦ} \cdot d - N_{св}, \quad (5)$$

где $\Sigma N_{ТЭЦ}$ – суммарная мощность ТЭЦ ЭО;

d – доля данной ТЭЦ в сумме мощностей ТЭЦ ЭО;

$N_{св}$ – мощность, расходуемая на собственные нужды электростанции.

Для КЭС:

$$D_{КЭС} = (N_{наг} - \Sigma D_{ГЭС} - \Sigma D_{ТЭЦ}) \cdot d - N_{св}, \quad (6)$$

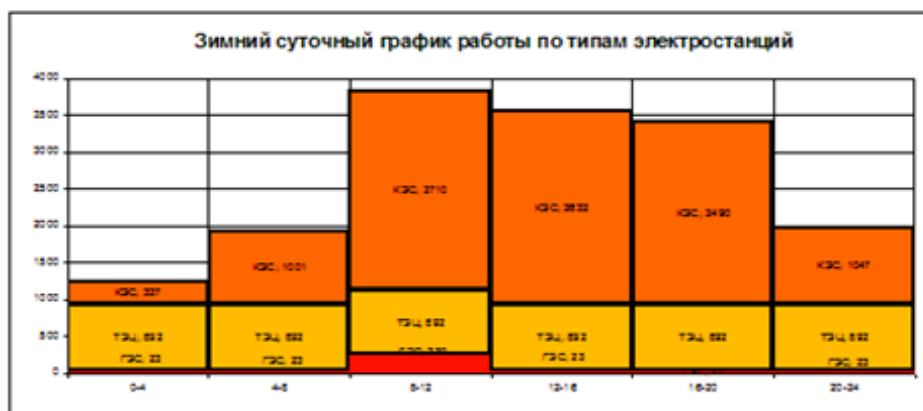
где $N_{наг}$ – мощность нагрузки в данный момент времени;

d – доля данной КЭС в сумме мощностей КЭС ЭО;

$N_{св}$ – мощность, расходуемая на собственные нужды электростанции.

Суточные графики выдачи мощности электростанциями в зимний максимум

Станции	Доля от суммы данного типа ЭС	0-4	4-8	8-12	12-16	16-20	20-24	В сутки	Среднесуточная часовая	В сезон
ГЭС1	0,61	14	14	141	14	14	14	849	35	154475
ГЭС2	0,39	9	9	89	9	9	9	532	22	96780
ГЭС Σ	1	23	23	230	23	23	23	1381	58	251255
ТЭЦ1	0,268	239	239	239	239	239	239	5741	239	1044826
ТЭЦ2	0,268	239	239	239	239	239	239	5741	239	1044826
ТЭЦ3	0,464	414	414	414	414	414	414	9936	414	1808352
ТЭЦ Σ	1	892	892	892	892	892	892	21418	892	3898003
КЭС1	0,551	180	551	1493	1450	1372	577	22498	937	4094653
КЭС2	0,449	147	449	1217	1182	1118	470	18332	764	3336384
КЭС Σ	1	327	1001	2710	2632	2490	1047	40830	1701	7431037
Σ выдача мощности		1243	1916	3833	3548	3406	1962	63628	2651	11580296



Продолжение приложения Д

6 Расчет себестоимости электроэнергии

Полная себестоимость электроэнергии определяется по формуле:

$$U = U_{cr} + U_{czt} + U_{np}, \text{ тыс. тенге / год} \quad (7)$$

где U_{cr} – годовые издержки энергосистемы по производству;

U_{czt} – оказанию услуг по передаче электроэнергии;

U_{np} – внепроизводственные затраты.

Годовые издержки U_{cr} определяются как:

$$U_{cr} = \sum (U_T + U_{z.p.} + U_A + U_{tr} + U_{osw}), \text{ тыс. тенге / год} \quad (8)$$

где U_T – расход за топливо;

$U_{z.p.}$ – заработная плата;

U_A – амортизация основных фондов;

U_{tr} – текущий ремонт;

U_{osw} – общестанционные нужды.

Издержки топлива на тепловых ЭПО определяются как:

$$U_T = \mathcal{E}_{oc}^* \cdot B_{uz} \cdot \left(\frac{7000}{Q_n^*} \right) \cdot C_T, \text{ тыс. тенге / год}, \quad (9)$$

где \mathcal{E}_{oc}^* – полезный отпуск электроэнергии электростанциями, кВт·ч;

Q_n^* – теплотворная способность топлива, Ккал/кг (приложение 1);

C_T – цена тонны топлива франко-станция потребитель, тенге/т;

B_{uz} – средний удельный расход условного топлива на единицу выработанной ЭЭ электростанциями, г.у.т /кВт·ч.

$$B_{uz} \text{ рассчитывается по формуле: } B_{uz} = \frac{0,124}{\eta}. \quad (10)$$

Для КЭС КПД берем в пределах от 0,25 до 0,3 в провале и от 0,37 до 0,43 в пике. КПД для ТЭЦ берем в пределах от 0,45 до 0,5 не зависимо от времени суток, т.к. ТЭЦ в данной РГР работает в постоянном режиме выдачи мощности. КПД электростанции в летний период падает на 25%.

Определение отпускной цены с шин тепловых электростанций

Коэффициент полезного действия

КПД тепловых электростанций	КПД (η) зимой в пике
ТЭЦ1	0,43
ТЭЦ2	0,43
ТЭЦ3	0,46
КЭС1	0,39
КЭС2	0,39

Удельный расход топлива Вуд. = 0,124/ η

Удельный расход топлива Вуд, кг/кВтч	Вуд. зимой в пике
ТЭЦ1	0,288
ТЭЦ2	0,288
ТЭЦ3	0,270
КЭС1	0,318
КЭС2	0,318

Продолжение приложения Д

Расход натурального топлива

$$B_{\text{нат}} = B_{\text{уд}} \cdot (Q_{\text{р.уг.}}/Q_{\text{р.нт}})$$

Расход натурального топлива В нат, кг/кВтч	Вуд. зимой в пике
ТЭЦ1	0,425
ТЭЦ2	0,425
ТЭЦ3	0,397
КЭС1	0,469
КЭС2	0,469

Топливные издержки на 1 кВт (В·Цт)

$$1 \text{ кг н.т. тенге} = 4,98$$

Топливные издержки на 1 кВтч тенге	В зимой в пике
ТЭЦ1	2,116
ТЭЦ2	2,116
ТЭЦ3	1,978
КЭС1	2,333
КЭС2	2,333

Полная себестоимость 1 кВт·ч

(топливная составляющая принимается равной 50%, прочие издержки – 50%)

Себестоимость (P*2)	Сзм в пике, тенге/кВтч
ТЭЦ1	4,2
ТЭЦ2	4,2
ТЭЦ3	4,0
КЭС1	4,7
КЭС2	4,7

Отпускная цена с шин тепловых электростанций (1,1·S)

Цена 1 квтч (1,1·S)	Ц зм в пике тенге/кВтч
ТЭЦ1	4,7
ТЭЦ2	4,7
ТЭЦ3	4,4
КЭС1	5,1
КЭС2	5,1

Продолжение приложения Д

Определение отпускной цены с шин гидроэлектростанций

Себестоимость производства на ГЭС ($S_{гэс} = Z_{ам} + Z_{пр}$)

Удельные капитальные затраты на ГЭС

Куд ГЭС, за 1 кВт	
\$	1200
тенге	175200
Курс тенге к \$	146

Себестоимость производства на ГЭС ($S_{гэс} = Z_{ам} + Z_{пр}$)

Z _{ам}	40%
Z _{пр}	60%

Капитальные затраты на ГЭС ($K_{уд} \cdot N_{гэс}$)

Кап. затраты	тенге
КГЭС 1	29083200000
КГЭС 2	18220800000

Затраты на амортизацию ($Z_{ам} = K_{гэс} \cdot N_{о}$),

$N_{о} = 0,03$

Амортизация	тенге
Z _{ам} ГЭС 1	872496000
Z _{ам} ГЭС 2	546624000

Себестоимость ЭЭ на ГЭС ($Z_{пр} = (0,6 \cdot Z_{ам}) / 0,4$)

Себестоимость	тенге
Z _{пр} ГЭС 1	1308744000
Z _{пр} ГЭС 2	819936000
P _{гэс} 1	2181240000
P _{гэс} 2	1366560000
S _{гэс} 1	7,04
S _{гэс} 2	7,04
Ц _{гэс} (1,1 · S _{гэс})	7,74