Министерство образования и науки РФ

ФГБОУ ВО «Калужский государственный университет

им. К. Э. Циолковского»

Физико-технологический институт

Кафедра информатики и информационных технологий

КУРСОВАЯ РАБОТА

На тему: «Моделирование и расчет работы солнечной панели, создание Web ресурса»

Выполнил: студент 3 курса

группы Б-ИСиТ-31

Критенко Валентин Григорьевич

Научный руководитель:

Белаш Виктория Юрьевна

Калуга 2018

Содержание:

[Введение 3](#_Toc517075151)

[Глава 1. Теоретические основы разработки программных продуктов 5](#_Toc517075152)

[1.1 Назначение и область применения разработки 5](#_Toc517075153)

[1.2 Установление требований к разрабатываемой системе 8](#_Toc517075154)

[1.3 Анализ существующих разработок 10](#_Toc517075155)

[Глава 2. Разработка программного продукта «Моделирование и расчет выработки солнечных панелей(МРСП)» Разработка Web ресурса 13](#_Toc517075156)

[2.1. Описание и обоснование выбора состава технических и программных средств разработки 13](#_Toc517075157)

[2.2 Алгоритм программы 16](#_Toc517075158)

[2.3 Руководство разработчика 18](#_Toc517075159)

[2.4 Руководство администратора 30](#_Toc517075160)

[2.5. Руководство пользователя 31](#_Toc517075161)

[Заключение 33](#_Toc517075162)

[Список литературы 34](#_Toc517075163)

[Приложения 35](#_Toc517075164)

# Введение

Солнечная энергетика представляет собой одно из перспективных направлений возобновляемой энергетики, основанное на непосредственном использовании солнечного излучения с целью получения энергии для отопления, электроснабжения и горячего водоснабжения. Солнце - неисчерпаемый, экологически безопасный и дешевый источник энергии. Как заявляют эксперты, количество солнечной энергии, которая поступает на поверхность Земли в течение недели, превышает энергию всех мировых запасов нефти, газа, угля и урана. По мнению академика Ж.И. Алферова, «человечество имеет надежный естественный термоядерный реактор - Солнце. Оно является звездой класса «Ж-2», очень средней, каких в Галактике до 150 миллиардов. Но это - наша звезда, и она посылает на Землю огромные мощности, преобразование которых позволяет удовлетворять практически любые энергетические запросы человечества на многие сотни лет». Причем, солнечная энергетика является «чистой» и не оказывает отрицательного влияния на экологию планеты. Немаловажным моментом является тот факт, что сырьем для изготовления солнечных батарей является один из самых часто встречающихся элементов - кремний. В земной коре кремний - второй элемент после кислорода (29,5% по массе). По мнению многих ученых, кремний - это «нефть двадцать первого века»: в течение 30 лет один килограмм кремния в фотоэлектрической станции вырабатывает столько электричества, сколько 75 тонн нефти на тепловой электростанции. По данным Института Энергетической стратегии, теоретический потенциал солнечной энергетики в России составляет более 2300 млрд. тонн условного топлива, экономический потенциал - 12,5 млн. т.у.т. Потенциал солнечной энергии, поступающей на территорию России в течение трех дней, превышает энергию всего годового производства электроэнергии в нашей стране. В России основным направлением развития стали поликристаллические и монокристаллические панели. Поликристаллические больше распространены в хозяйствах и домах, где нужна дешевизна и есть место для постановки большего количества панелей. Монокристаллические, для предприятий и станций, где нужна эффективность и малый размер площадей.

Цель работы: создать программный продукт позволяющий проводить работу с БД и вести расчет-модуляцию солнечных панелей, создание Web ресурса о видах солнечных панелей, истории развития и калькулятор.

Задачи:

- определить назначение и область применения программного продукта и web сайта;

- собрать и проанализировать информацию по программным продуктам схожим или близким к затрагиваемой теме;

- изучить литературу и иные источники информации по теме курсовой работы;

- выбрать технические и программные средства для разработки;

-реализовать приложение по управлению базой данных;  
 - написать информационный сайт с калькулятором;

-написать руководство к созданным продуктам и сайту

- подвести итоги проделанной работы, указать возникшие затруднения и шаги для их устранения.

Работа состоит из введения, двух глав, заключения, списка использованной литературы и приложений.

# Глава 1. Теоретические основы разработки программных продуктов

# 1.1 Назначение и область применения разработки

На сегодня есть несколько основных технологий производства солнечных батарей, которые основаны на применении какого-либо материала при создании пластины. Базируется это на разном поглощении различными материалами солнечного излучения. Наибольшей популярностью среди используемых материалов пользуются поли- и монокристаллический кремний, CdTe, GaAs, аморфный кремний и другие. В зависимости от выбранного материала используется определенная технология, отличающаяся стадиями производства и комплексом оборудования. Чаще всего как сырье применяется поли- и монокристаллический кремний. КПД пластин из данного материала колеблется в диапазоне от 12 до 19%. Данные пластины довольно хрупкие, им необходима дополнительная защита, но они намного дешевле, чем пластины из других материалов. Тонкопленочная технология базируется на применении таких материалов: GaAs, аморфный кремний и CdTe. КПД этих пластин тоже не выше 20%, хотя в будущем есть планы повышения его до 22%. В зависимости от подложки, которая используется, эти батареи могут гнуться, герметичны, устойчивы к механическим воздействиям. Но и их стоимость превышает стоимость кремниевых систем. Как всем известно, солнечные батареи представляют собой фотоэлектрические элементы, которые под воздействием солнечных лучей генерируют электрическую энергию. И не важно, где используются эти панели, на приусадебном участке или крупном предприятии, принцип их работы не изменится, различие между ними состоит лишь в размере и количестве вырабатываемого электричества, то есть для работы предприятий необходимы более мощные солнечные батареи. Стоит отметить, что многие мировые производители специализируются именно на промышленных энергетических системах, наиболее известные из них Naps Systems Oy и Kyocera Solar. Объясняется это нехваткой основных источников энергии (газ, нефть, уголь), особенно заметно это в Испании, Германии, США, Объединенных Арабских Эмиратах - в странах, где солнечная энергетика уже давно развивается стремительными темпами. В России же на сегодняшний день солнечные батареи скорее редкость, чем правило, но продвижения все же есть, и очень даже заметные. В России наибольшее распространение солнечные батареи получили в малом бизнесе, например в фермерских хозяйствах. Причина этого заключается в удаленности от центральной энергосети, ведь зачастую предпринимателям гораздо выгоднее установить фотоэлектрические батареи, чем прокладывать линии электропередач. Если же рассматривать крупные промышленные энергетические системы в России, то наиболее ярким примером выступают метеостанции, которые в рамках реализации проекта «Модернизация и техническое перевооружение сети метеостанций РФ» оснащаются солнечными панелями производства Naps Systems Oy. Помимо прочего солнечные батареи можно встретить: на очистных сооружениях; на автозаправочных станциях в различных мировых странах, в том числе и в России на одной из АЗС ЛУКОЙЛ в Красной Поляне; на промышленных складах и телекоммуникационных системах; в газовой и нефтяной промышленности на системах катодной защиты; на железных дорогах и т.д. Как Вы видите, солнечные батареи могут эффективно использоваться в различных областях, но, несмотря на это, говорить о повсеместном использовании альтернативных источников энергии на крупных предприятиях пока еще рано. Швеция, Финляндия, Италия, Греция, Германия, Австрия - вот лишь некоторые из стран, которые активно пользуются энергией солнца и экономят при этом немалые деньги. К примеру, в Хельсинки на заводе АВВ солнечная энергосистема мощностью 181 кВт используется для зарядки батареи грузоподъемников, а в Италии солнечные панели разместились на крышах промышленных складов. Но лидирующие позиции занимает все-таки Германия. Немцы с их врожденной бережливостью экономят каждый Ватт. Примером может служить так называемое пассивное офисное здание nZEB, все его инженерные системы работают за счет энергии, генерируемой установленными на крыше солнечными батареями. В Европе и США солнечные батареи уже давно в массовом использовании, а вот страны Средней Азии или наши соседи - дело другое. У кого-то альтернативная энергетика находится в самом зародыше, а у кого-то есть неплохие продвижения в этой области. Так, в Узбекистане проводится широкая программа по переходу крупных хозяйственных объединений и промышленных предприятий на солнечные батареи и коллекторы с целью снижения расхода первичных топливно-энергетических ресурсов. По официальным данным за 2013 год в республике было введено в эксплуатацию 29 солнечных коллекторов и 9 фотоэлектрических станций, в нынешнем же году ведутся работы по запуску еще 88 солнечных коллекторов и 60 фотоэлектрических станций. Но это лишь «капля в море», ведь для масштабного перехода на альтернативные источники энергии требуются более высокие мощности. Хотя начало положено, а это главное! С недавнего времени солнечные батареи пользуются популярностью во всем мире. Применение солнечных батарей в микроэлектронике: (как зарядное устройство) для обеспечения электричеством аккумуляторов разной бытовой электроники - плееров, калькуляторов, фонариков и других, для подзарядки электромобилей. Например, в автомобиле Skoda Superb в одной из комплектаций можно установить солнечную батарею на крышу автомобиля и тогда в жаркие дни, салон автомобиля будет проветриваться встроенным вентилятором, работающим от этой батареи, пока автомобиль находится на стоянке. Применение солнечных батарей для энергообеспечения зданий - большие батареи работают как солнечные коллекторы, особенно популярны в субтропических и тропических регионах с большим числом солнечных дней. Основной целью создание ПО и web-ресурса является предоставление информации, расчет по заданным критериям работы и модуляции солнечной панели.

# 1.2 Установление требований к разрабатываемой системе

Разработанный программный продукт предназначен для обычных людей. Условно разделены на пользователя и администратора, для удобства работы с данными. Разработанный сайт должен быть прост и понятен пользователю, информативен.

Для нормальной работы программы необходим IBM совместимый компьютер с тактовой частотой процессора не ниже 1.2ГГц и выше, оперативной памятью 4 ГБ, жесткий диск объемом не менее 30 ГБ, Windows 7(x64) и выше. Для работы сайта требуется браузер Google Chrome или его аналоги.

Требования к дизайну разрабатываемого программного продукта:

* Структурированная информация;
* Приятный дизайн.

Технические требования.

Программный продукт «МРСП» должен:

хранить информацию о солнечных панелях.

предоставлять человеку возможность выбора данных

хранить информацию о производителе, типе, погодных условиях, добавлять информацию и дополнять/редактировать имеющуюся информацию о них ;

Функциональные требования.

Программный продукт «МРСП» должен:

производить расчет из БД;

производить расчет из введенных данных ;

добавлять данные в БД;

изменять данные БД;

удалять данные БД;

Web-ресурс не должен быть перегружен слишком большим количеством кнопок и пунктов. Интерфейс должен быть “Дружелюбным” – это значит, что он должен быть выполнен в неярких тонах, без навязчивых окон и предельно понятным и простым. Дружелюбный интерфейс позволяет пользователю фокусироваться на собственных задачах, а не особенностях взаимодействия с системой.

Сайт должен ознакомить с актуальной информацией о солнечных панелях; производить расчет данных в калькуляторе вводимым пользователем.

Требования к дизайну разрабатываемому программному продукту:

* + Структурированная информация;
  + Корректное отображение содержимого web-страниц;
  + Приятный дизайн.

Функциональные требования к сайту:

Понятные информационные блоки

Написание инструкции к полям и обозначениям калькулятора

Написание калькулятора для расчетов

# 1.3 Анализ существующих разработок

Проанализировав рынок программных продуктов, были найдены следующие продукты:

Свет on - есть 2 калькулятора, онлайн и доступный для скачки, но он доступен только покупателям компании. Согласно описанию настольной программы, он имеет возможности: «Калькулятор выдаёт данные в виде графиков. Вы можете наглядно оценить, какой угол наклона необходим для каждого сезона. На графике видно, сколько энергии в среднем выдаёт солнечная электростанция или панель в любое время года. Вы можете получить оптимальный угол наклона солнечных батарей для своей местности, а также можете рассчитать наилучший постоянный угол, чтобы получить максимальную выработку от солнечных панелей в конкретное время года (например, только для летнего проживания) В отличие от других схожих программ, данные которых ограничены городами, наш калькулятор выполняет расчёт для любой местности. Это могут быть горы, поля и степи. Наша страна обладает огромной территорией. Особенно это актуально для русского Севера, где населённые пункты находятся очень далеко друг от друга. Если Вы введёте конкретные координаты, то калькулятор даст информацию именно для этого места.» Онлайн калькулятор дает возможность посчитать сколько основных потребителей будет в квартире (телевизор , холодильник, освещение, насос).

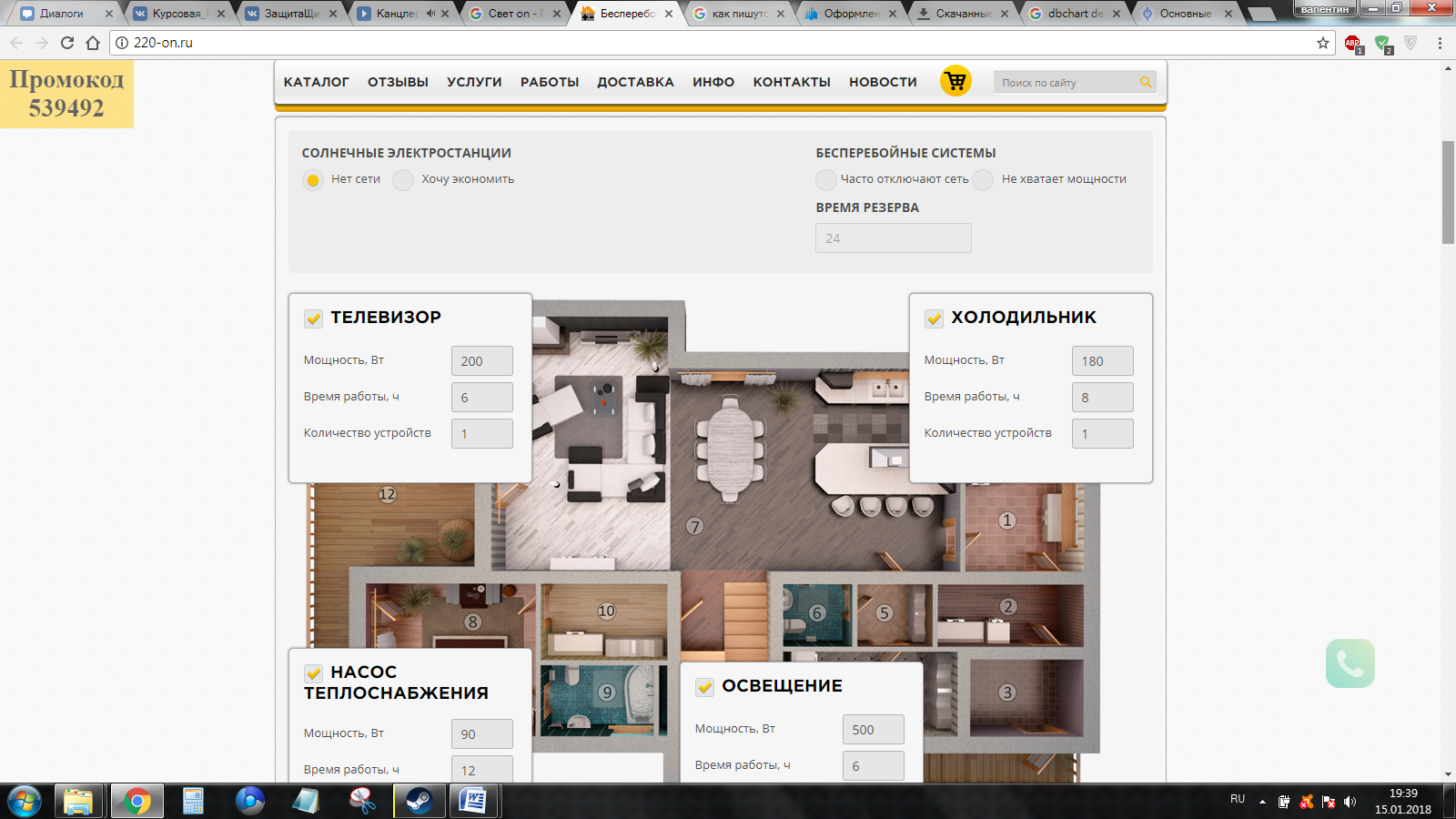


Рис 1. Комнаты (Свет on)

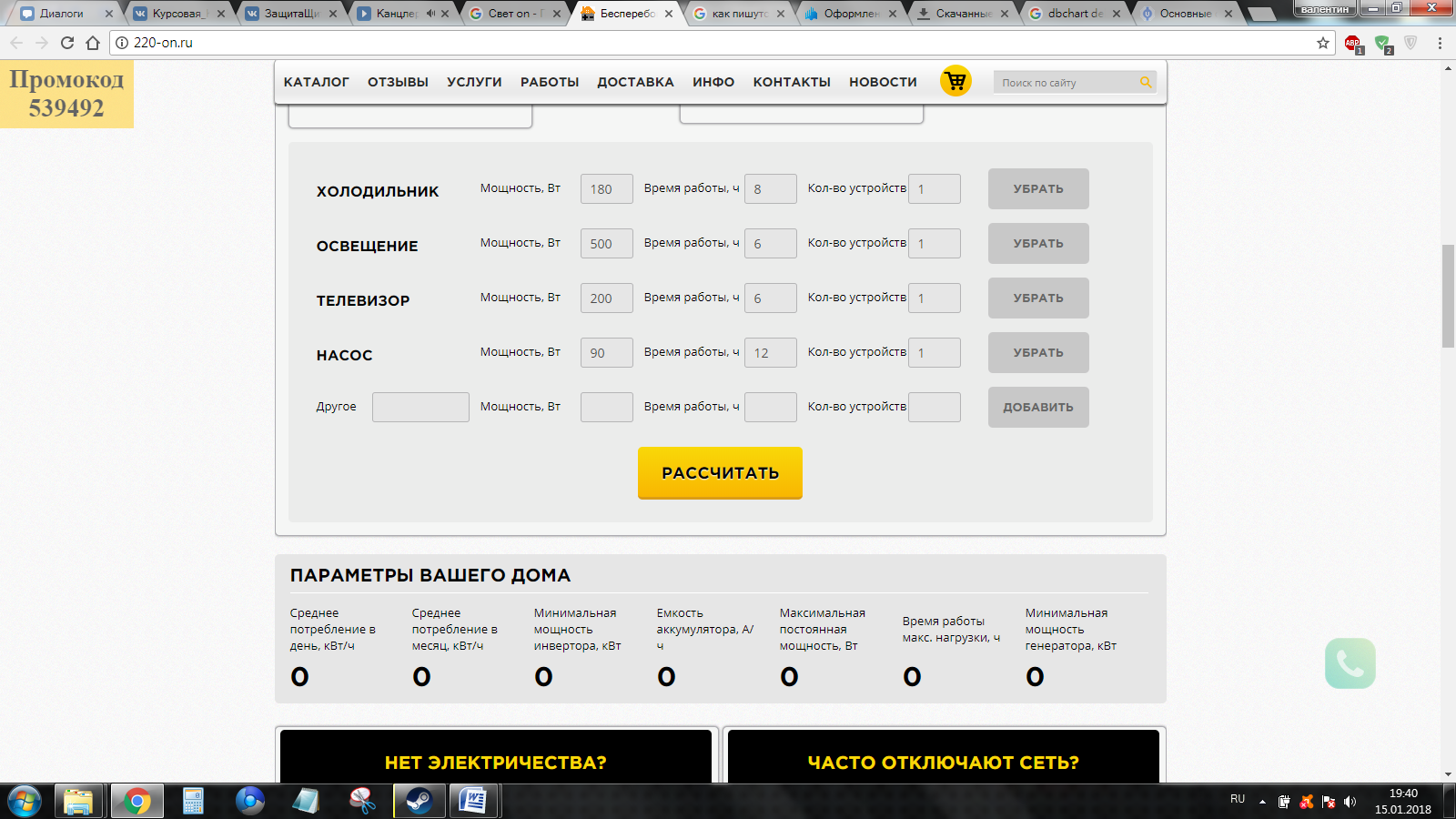
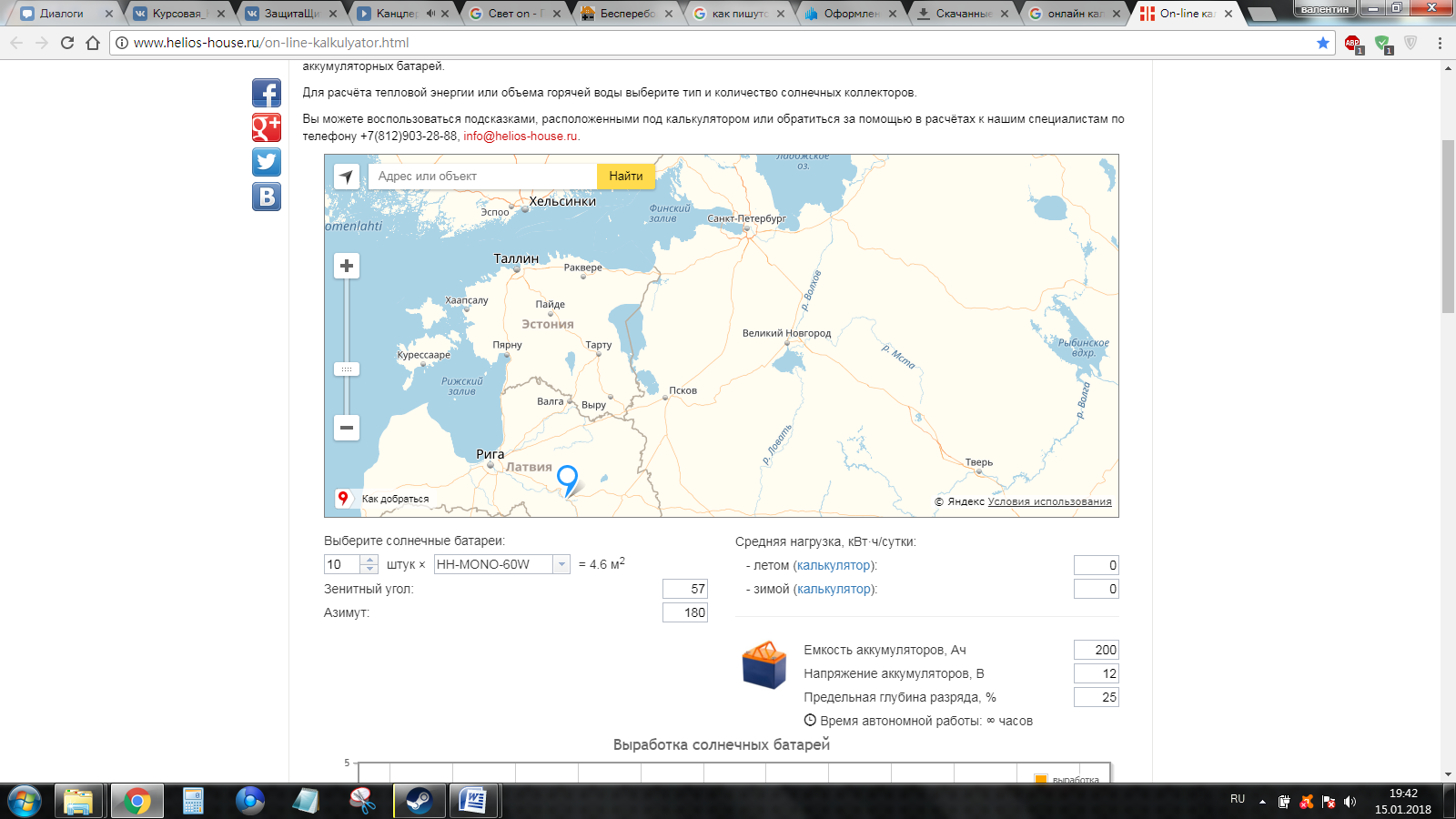


Рис 2. Потребители (Свет on)

Из минусов, для работы данной программы нужен интернет. Она фиксирована к потребителям.

Рис 3. Расположение, мощность ( helios-house)

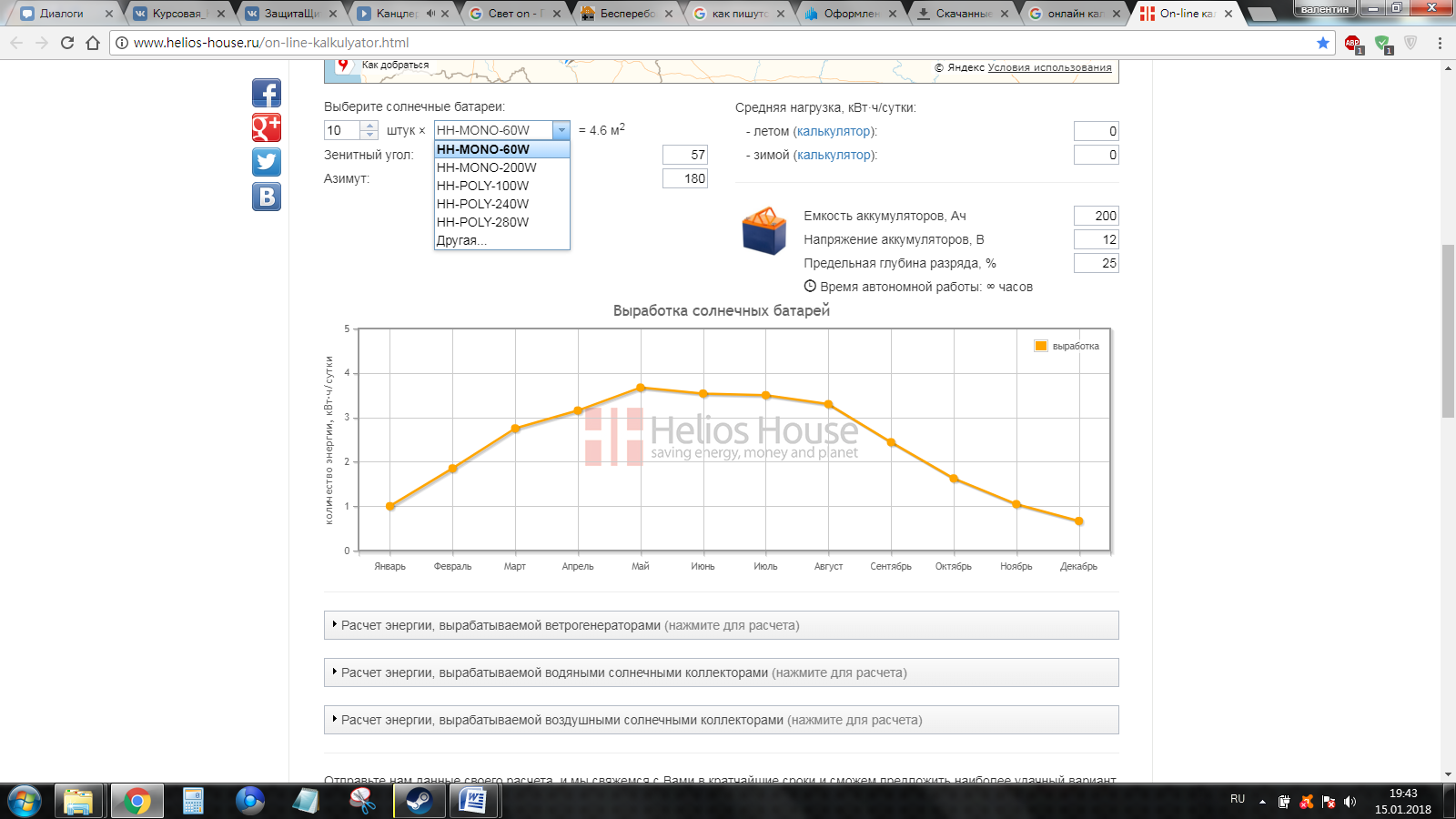


Рис 4. Выбор солнечной панели ( helios-house)

Онлайн калькулятор( helios-house) Позволяет произвести полностью расчет разных источников энергии и рассчитать потребление и выработку. Из минусов, привязаны к сети интернет. Разработка программного продукта даст возможность любителям и профессиональным рабочим без проблем сделать расчет и определиться с выбором той или иной солнечной панели.

# Глава 2. Разработка программного продукта «Моделирование и расчет выработки солнечных панелей(МРСП)» Разработка Web ресурса

# 2.1. Описание и обоснование выбора состава технических и программных средств разработки

При создании программы применялись: Microsoft Visual Studio 2017 — 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, Xbox, 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)

C#— простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов — это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие декларативные сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# отлично подходит для создания и применения программных компонентов.

Вот лишь несколько функций языка C#, обеспечивающих надежность и устойчивость приложений: сборка мусора автоматически освобождает память, занятую уничтоженными и неиспользуемыми объектами; обработка исключений дает структурированный и расширяемый способ выявлять и обрабатывать ошибки; строгая типизация языка не позволяет обращаться к неинициализированным переменным, выходить за пределы массива или выполнять неконтролируемое приведение типов.

Microsoft Office Access или просто Microsoft Access - реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных. В базе данных Access (если ее специально не настраивали для работы с данными или кодом, принадлежащими другому источнику) все таблицы хранятся в одном файле вместе с другими объектами, такими как формы, отчеты, макросы и модули. Для файлов баз данных, созданных в формате Access 2007 (который также используется в Access 2016, Access 2013 и Access 2010), используется расширение ACCDB, а для баз данных, созданных в более ранних версиях Access, - MDB. С помощью Access 2016, Access 2013, Access 2010 и Access 2007 можно создавать файлы в форматах более ранних версий приложения (например, Access 2000 и Access 2002-2003). Использование Access позволяет: добавлять новую информацию в базу данных, например, новый артикул складских запасов; изменять информацию, уже находящуюся в базе, например, перемещать артикул; удалять информацию, например если артикул был продан или утилизирован; упорядочивать и просматривать данные различными способами; обмениваться данными с другими людьми с помощью отчетов, сообщений электронной почты, внутренней сети или Интернета.

Bootstrap 3 — фреймворк, набор HTML+CSS инструментов и шаблонов для верстки и более эффективного и быстрого создания сайтов и веб-приложений более эффективно и быстро. Bootstrap 3 — современный помощник, разработчиков интерфейсов, дизайнеров и вебмастеров, доступный для использования по открытой лицензии. Этот фреймворк очень динамичный и регулярно обновляемый. Bootstrap 3 сейчас на пике популярности и можно уверенно сказать, что на его основе можно сделать любой веб-интерфейс. Так же он полностью отвечает критериям для создания сайта.

# 2.2 Алгоритм программы

Приложение для пользователей дает возможность рассчитать данные автоматически , задав параметры и погодные условия, нужные для симуляции. Приложение для Администратора дает ему полный доступ к данным БД. Можно вносить изменения, удалять и изменять данные.

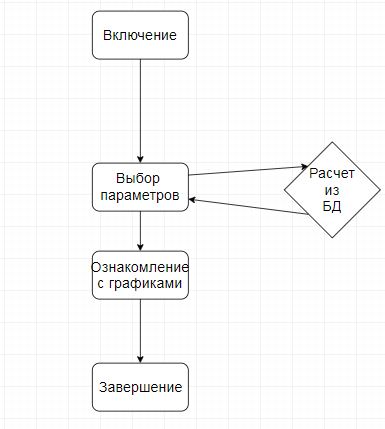


Рис 5.Алгоритм пользовательской программы

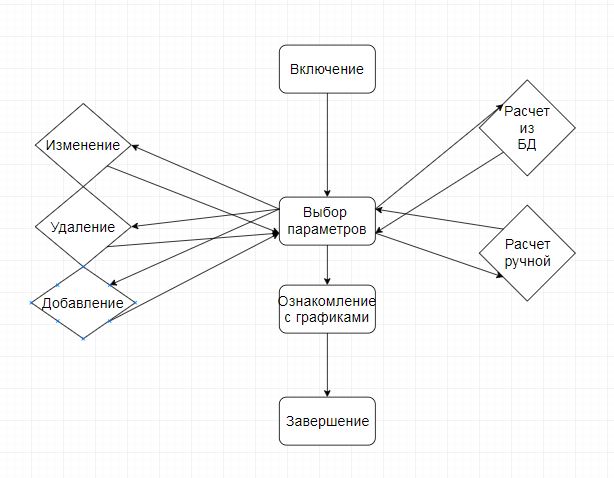


Рис 6.Алгоритм программы администратора

# 2.3 Руководство разработчика

В настоящее время жизнь человека зависит от различного рода информации, для управления которой требуются создания огромного количества баз и банков данных различного назначения.Система управления базами данных (СУБД) - это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. СУБД MS ACCESS представлена в виде таблиц. По сути БД- это некоторое подобие картотеки, электронного хранилища данных, которые хранятся в компьютере в виде одного или нескольких файловДанная курсовая работа рассматривает создание базы данных «Моделирование и расчет выработки солнечных панелей». В курсовой работе разрабатывается БД, с помощью которой, можно провести расчет-симуляцию работы солнечных панелей. Проектирование базы данных.База данных (БД)- совокупность взаимосвязанных и организованных по определенным правилам данных, которое отражает состояние объектов какой-то конкретной предметной области и отношения между этими объектами. СУБД используется разработчиками и пользователями, обращающимися к СУБД напрямую или косвенно, через прикладные программы. Базы данных состоят из четырех основных компонентов: Данных пользователя, метаданных, индексов и метаданных приложений. Данные пользователей представляются в виде отношений - двумерных таблиц данных. Столбцы таблицы содержат поля, или атрибуты, а строки содержат записи о конкретных объектах делового мира .База данных состоит как из данных, так и из метаданных. Метаданные - это данные, которые являются описанием структуры данных, находящихся внутри базы. Зная, как расположены данные, можно их получить. Так как описание структуры базы данных находится в самой базе, то она является самоописательной. База данных является интегрированной, ибо содержит в себе не только элементы данных, но и существующие между ними взаимосвязи. В базе данных метаданные хранятся в области, которая называется словарь данных. Он описывает таблицы, столбцы, индексы, ограничения и другие компоненты, из которых состоит база данных.Индексы, которые хранятся в базе данных, призваны улучшить её производительность и доступность. Метаданные приложений описывают структуру и формат пользовательских форм, отчетов, запросов и других компонентов приложений. В Access в полной мере реализовано управление реляционными базами данных. Система поддерживает первичные и внешние ключи и обеспечивает целостность данных на уровне ядра (что предотвращает несовместимые операции обновления или удаления данных). Кроме того, таблицы в Access снабжены средствами проверки допустимости данных, предотвращающими некорректный ввод вне зависимости от того, как он осуществляется, а каждое поле таблицы имеет свой формат и стандартные описания, что существенно облегчает ввод данных. Access поддерживает все необходимые типы полей, в том числе текстовый, числовой, счетчик, денежный, дата/время, MEMO, логический, гиперссылка и поля объектов OLE. Если в процессе специальной обработки в полях не оказывается никаких значений, система обеспечивает полную поддержку пустых значений. Реляционная обработка данных в Access за счет гибкой архитектуры системы способна удовлетворить любые потребности. При этом Access может использоваться как автономная СУБД в режиме файл-сервера или клиентского компонента таких продуктов, как SQL Server. Основная цель системы управления базами данных заключается в том, чтобы предложить пользователю абстрактное представление данных, скрыв конкретные особенности хранения и управления ими. Следовательно, отправной точкой при проектировании БД должно быть общее описание информационных потребностей пользователей, которые должны найти свое отражение в создаваемой базе данных.Прежде чем создавать таблицы, формы и другие объекты необходимо задать структуру базы данных. Хорошая структура базы данных является основой для создания адекватной требованиям, эффективной базы данных.

**Проектирования физической базы данных**

Основным структурным элементом БД является таблица. В ней хранятся вводимые данные, а также структура базы (поля, их типы и свойства). Данная база состоит из нескольких таблиц созданных в режиме конструктора. Между собой таблицы не соединены. БД выступает только в роли хранилища. Вся обработка происходит внутри приложения.

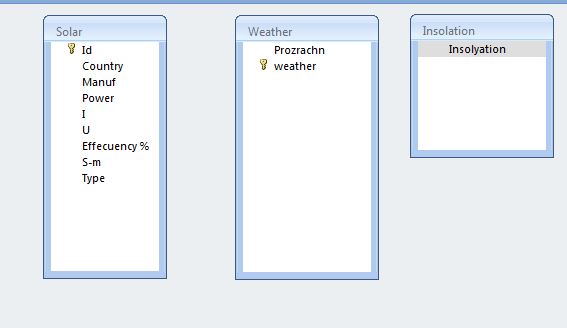


Рис 7. физическая модель базы данных

**Загрузка данных.**

После создания всех полей и определения их свойств в таблицу можно вводить информацию. Для этого необходимо войти в режим таблицы. Новая таблица Access состоит из одной пустой записи. Чтобы её заполнить, необходимо ввести несколько строк с данными. После ввода данных пустая запись смещается в конец таблицы. На листе данных активная запись обозначается треугольным маркером, а пустая - звездочкой. Для обозначения записи, в которой выполняется ввод, используется изображение карандаша. Все маркеры появляются в левой части листа данных. Запись таблицы активизируется при выполнении на ней щелчка. Переходить от записи к записи и от поля к полю таблицы позволяют также клавиши управления курсором. С помощью клавиш Tab и Enter можно перемещаться по полям слева направо, а посредством клавиш Shift+Tab - в обратном направлении. В активном поле появляется мерцающий курсор ввода, свидетельствующий о том, что можно начинать ввод. Переход в другое поле расценивается программой как подтверждение ввода, выполненного в предыдущем поле. После активизации любого поля записи в строке состояния появляется комментарий, который введен пользователем в поле Описание при составлении таблицы. Также загрузка данных на этапе проектирования может производиться с помощью форм. Довольно часто в таблицу вводятся некорректные данные. Чтобы избежать таких ошибок, можно задать условия и значения. После создания базы данных и заполнении её данными необходимо начать разработку приложения по обработке информации.

**Создание приложения:**

Открываем Visual Studio 2017. Создаем проект Windows form.   
На форму закидываем элементы :  
**binding navigator** - Представляет пользовательский интерфейс для перехода и обработки для элементов управления в форме, которые привязываются к данным.

**DataGridView** предоставляет настраиваемую таблицу для отображения данных. Класс DataGridView допускает настройку ячеек, строк, столбцов и границ с помощью использования свойств, таких как DefaultCellStyle, ColumnHeadersDefaultCellStyle, CellBorderStyle и GridColor. Дополнительные сведения см. в разделе Базовое форматирование и оформление элемента управления DataGridView в Windows Forms.

Можно воспользоваться элементом управления DataGridView для отображения данных вместе с источником базовых данных или без него. Button

**Chart** встроенная область для рисования графиков

**Textbox** Предоставляет элемент управления "текстовое поле" Windows.  
может хранить значения  
**Label** – надпись которую можно редактировать или задавать значение

**Button** – обычная кнопка выполняющая запрограммированные действия по нажатию.  
Приложение пользователя обладает функциями автоподсчета из данных БД.  
Приложение администратора дает возможность автоподсчета из БД, ручного, изменения и удаления данных.  
Пример кода из кнопки «образец 1»

double pr1, in1, po1, s1, r1 = 0;

pr1 =Convert.ToDouble(prozrachnTextBox.Text);

in1 = Convert.ToDouble(insolyationTextBox.Text);

po1 = Convert.ToDouble(powerTextBox.Text);

s1 = Convert.ToDouble(s\_mTextBox.Text);

r1 = pr1\*in1\*po1/100/1000;

label2.Text = prozrachnTextBox.Text;

label3.Text = insolyationTextBox.Text;

label4.Text = powerTextBox.Text;

label5.Text = s\_mTextBox.Text;

label11.Text = Convert.ToString(r1);

chart1.Series["obj1"].Points.AddXY(1,Convert.ToInt64(r1));

chart2.Series["obj1"].Points.AddXY(1, Convert.ToInt64(s1));

кнопка Очистить  
chart1.Series[0].Points.Clear();

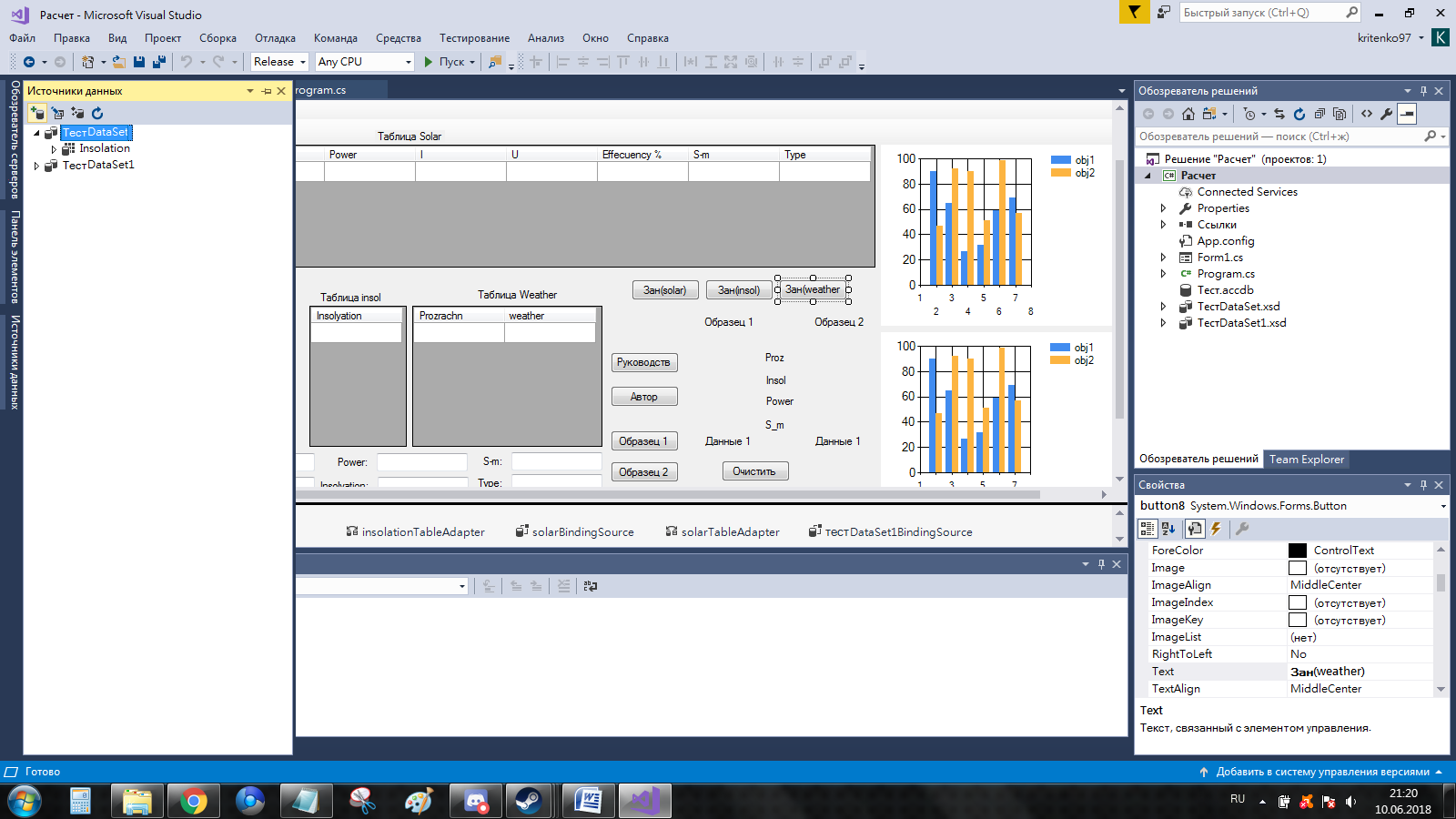
chart1.Series[1].Points.Clear();

chart2.Series[0].Points.Clear();

chart2.Series[1].Points.Clear();

пример кнопки Зан(weather)

this.weatherTableAdapter.Update(this.тестDataSet1.Weather);  
Как происходит подключение проекта к БД   
Открываем проект. Слева ищем «источник данных» клавиши Shift+Alt+D

Рис 8 источник данных

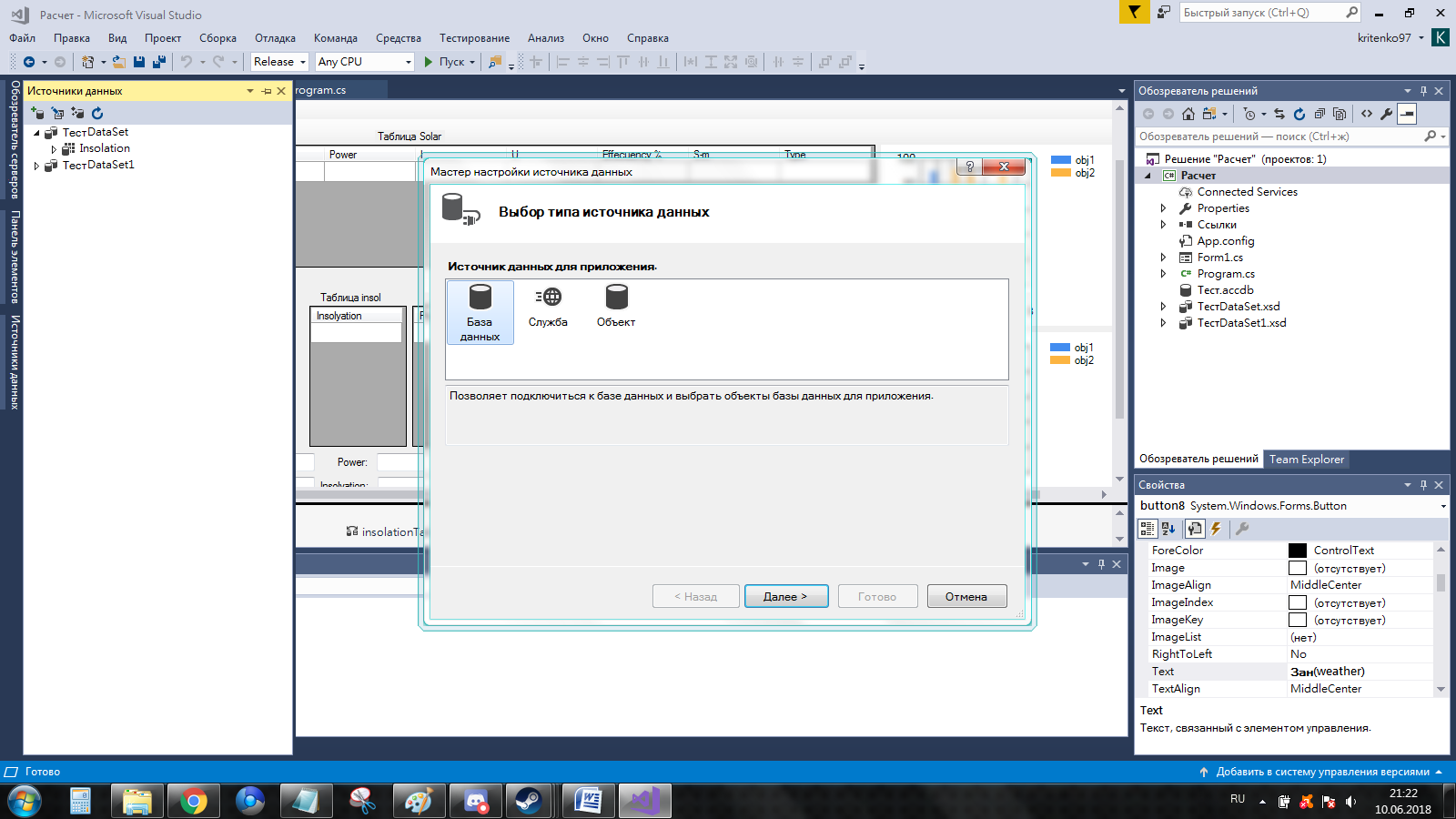


Рис 9 подключение БД

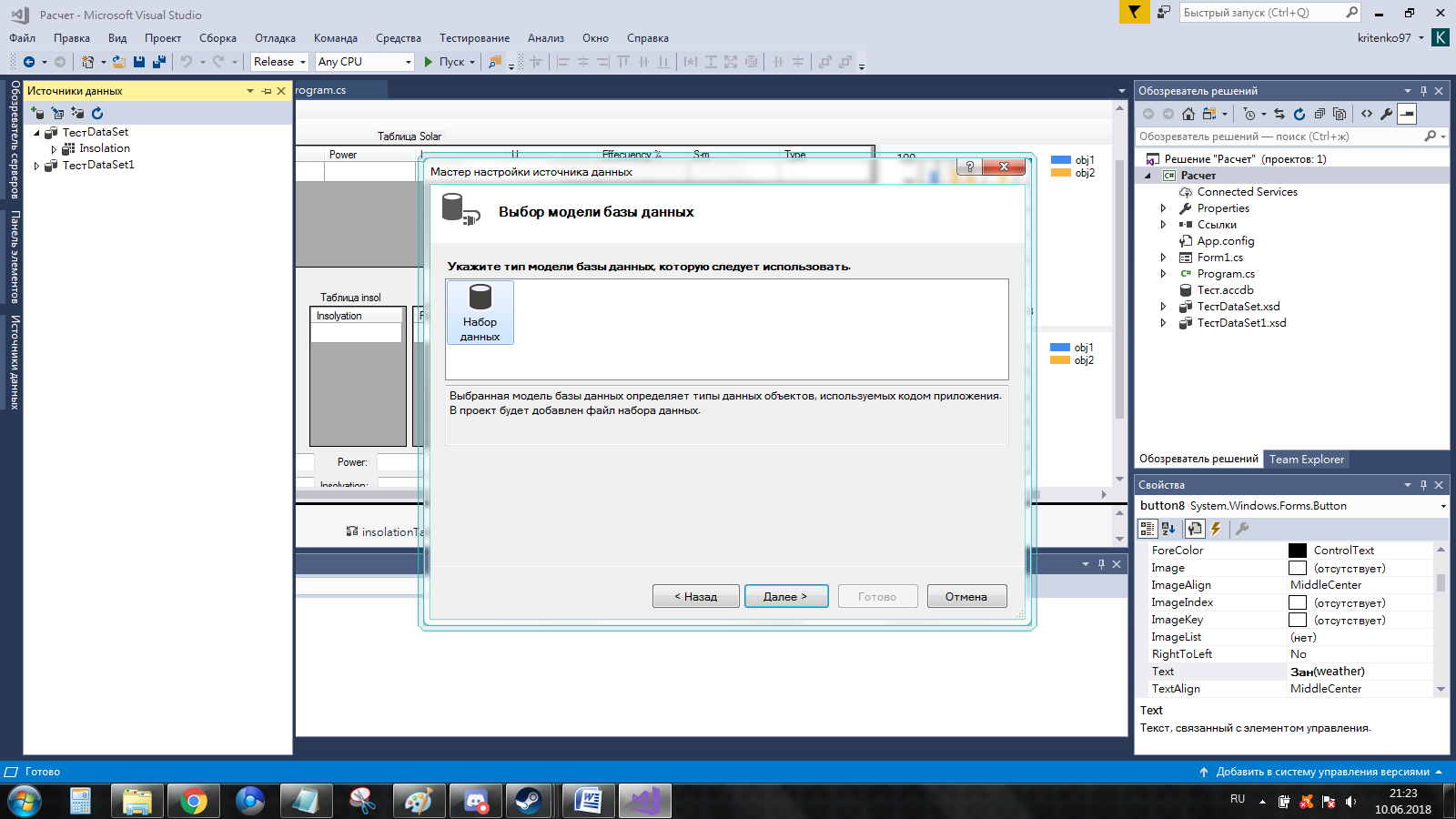


Рис 10 подключение БД

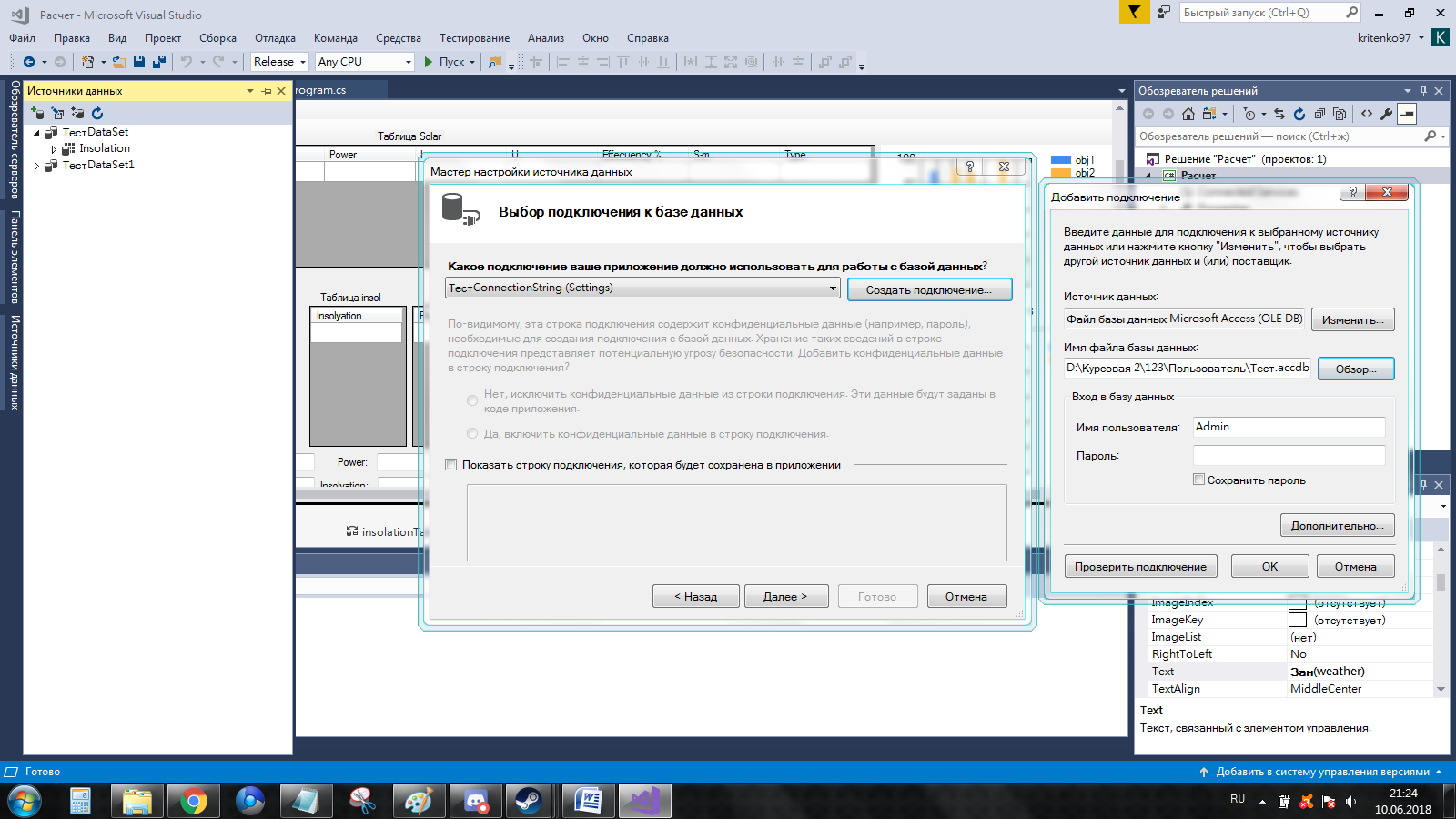


Рис 11 подключение БД

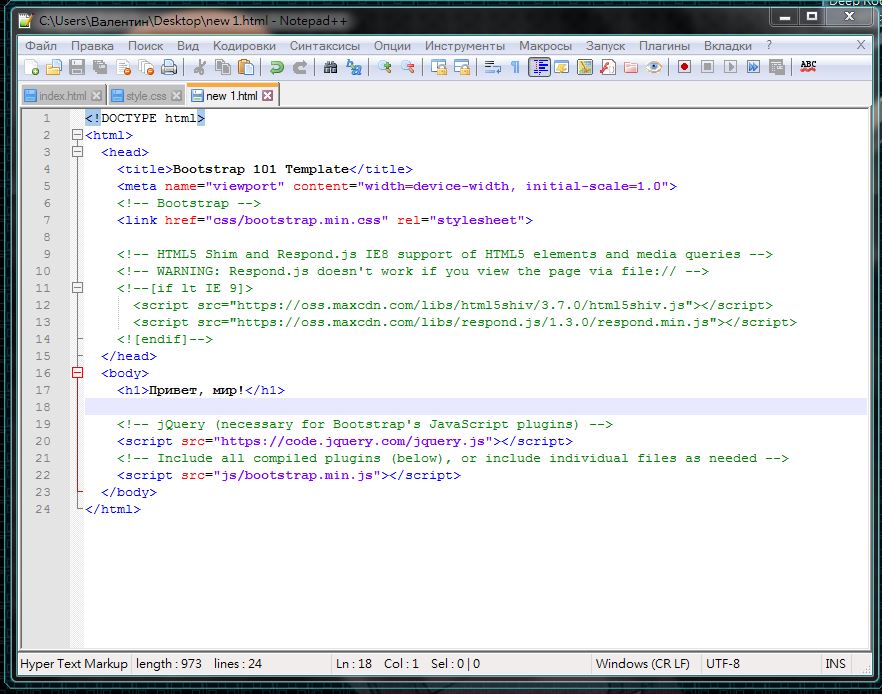
**Написание сайта**В интернете находим базовый шаблон на Bootstrap 3  


Рис 12 шаблон

Дальше подключаем свой файл CSS расширений. Ставим язык сайта «ru». Автоматически к нашему сайту подключены библиотеки стилей Bootstrap, JS.

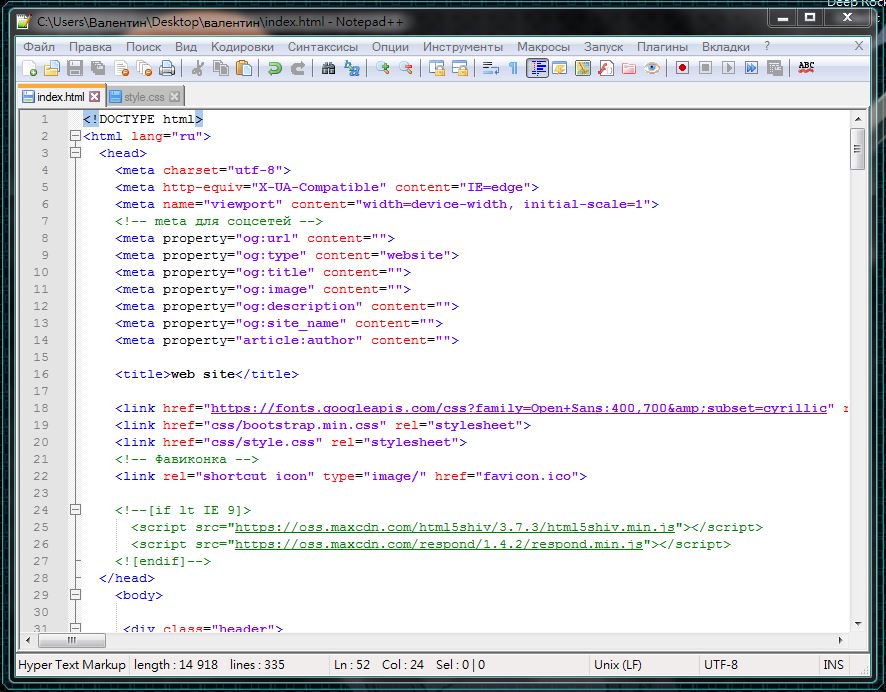


Рис 13 подключаем стили

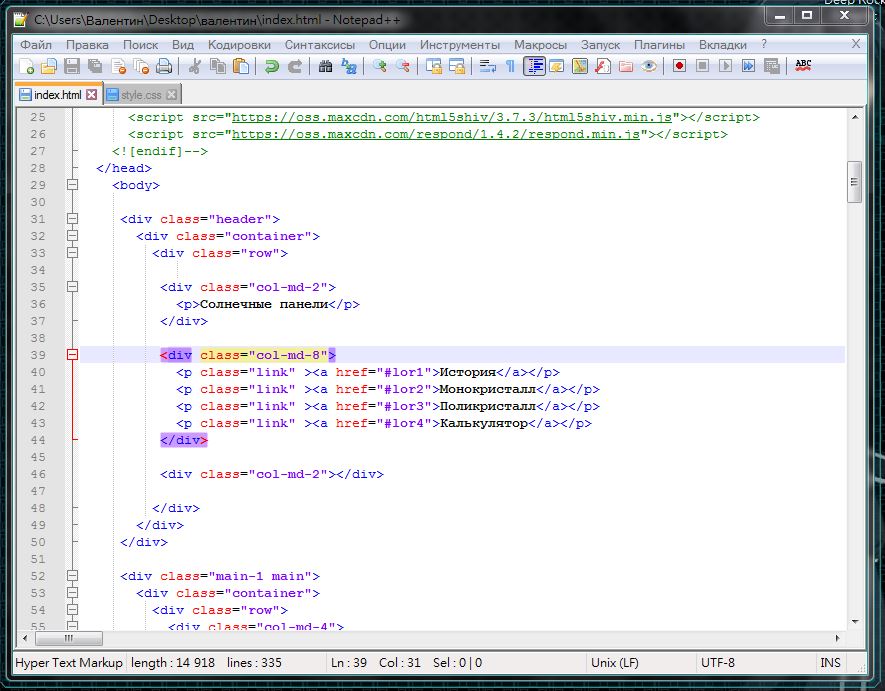


Рис 14 Шапка сайта

Здесь мы создаем главное меню сайта. Разбиваем его на колонки. Делаем кликабельными разделы меню, плавный переход на разделы, ссылки на них.

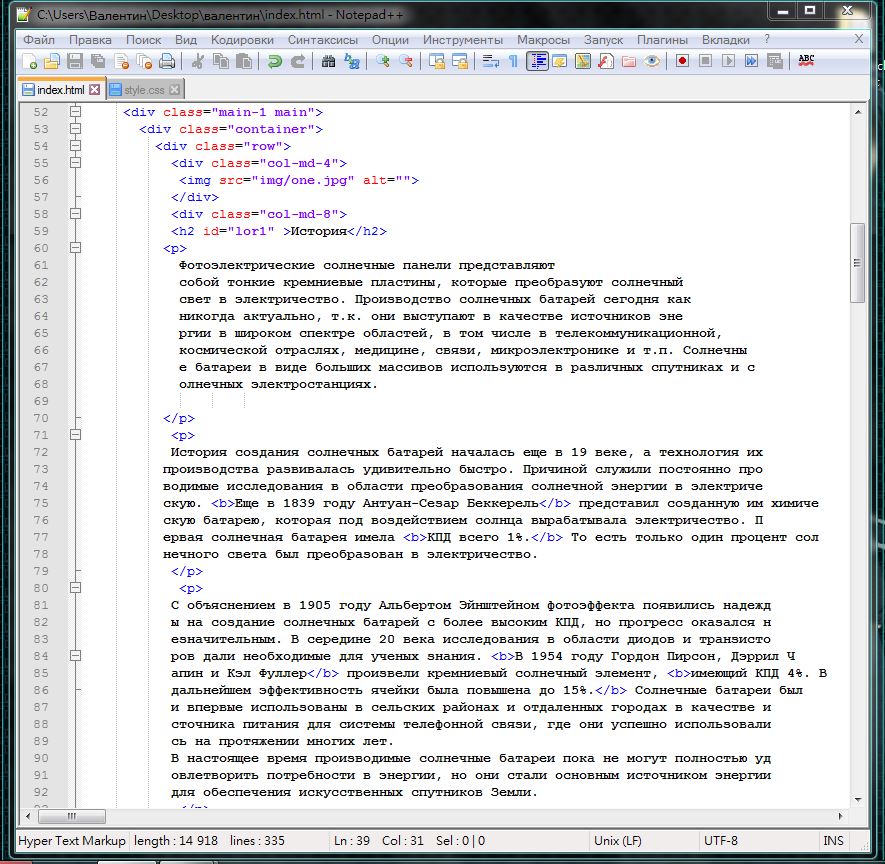


Рис 15 история

Пункт меню история. Представляет собой контейнер.

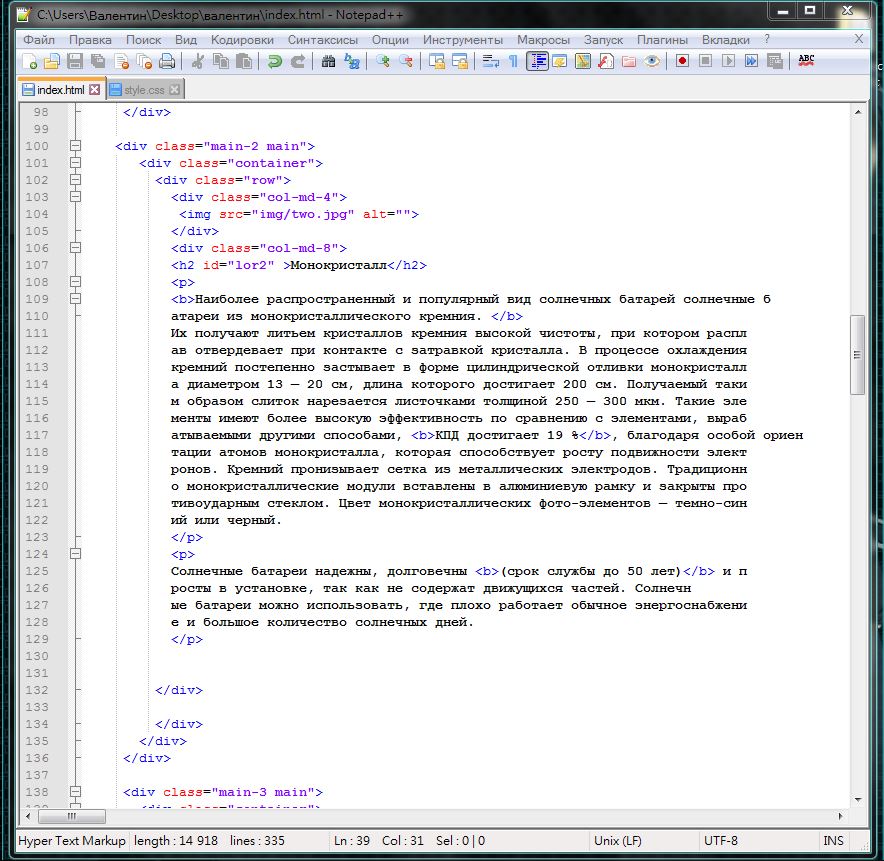


Рис 16 Моноскристалл

Развитие, кпд, характеристики монокристаллических фотоэлементов

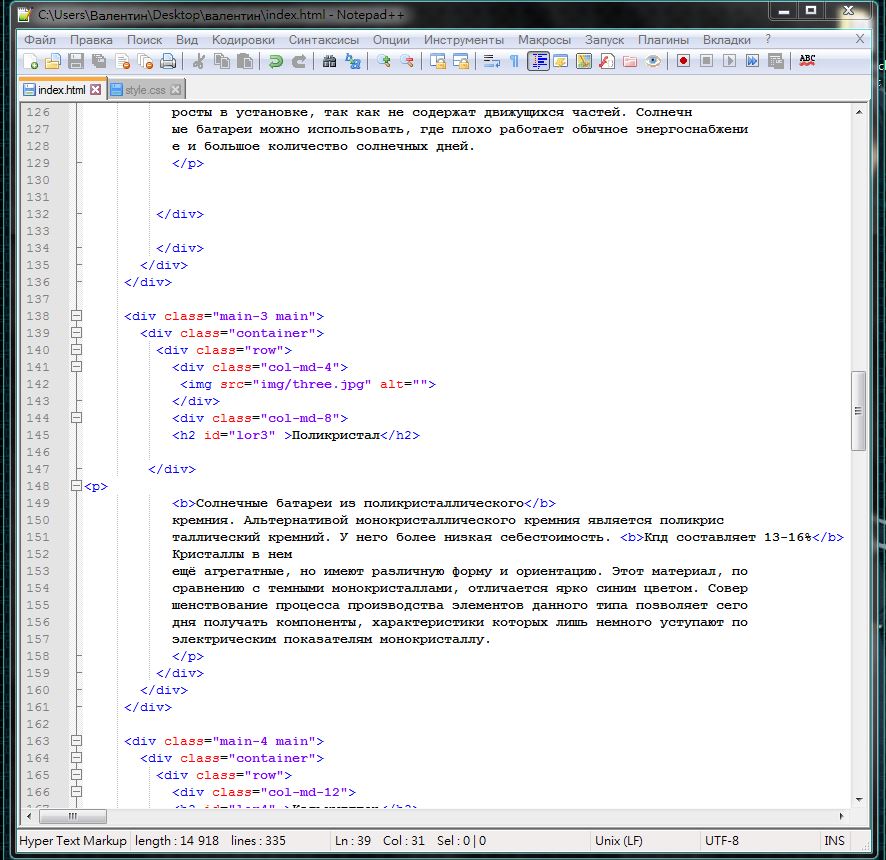


Рис 17 поликристалл

Поликристаллические фотоэлементы

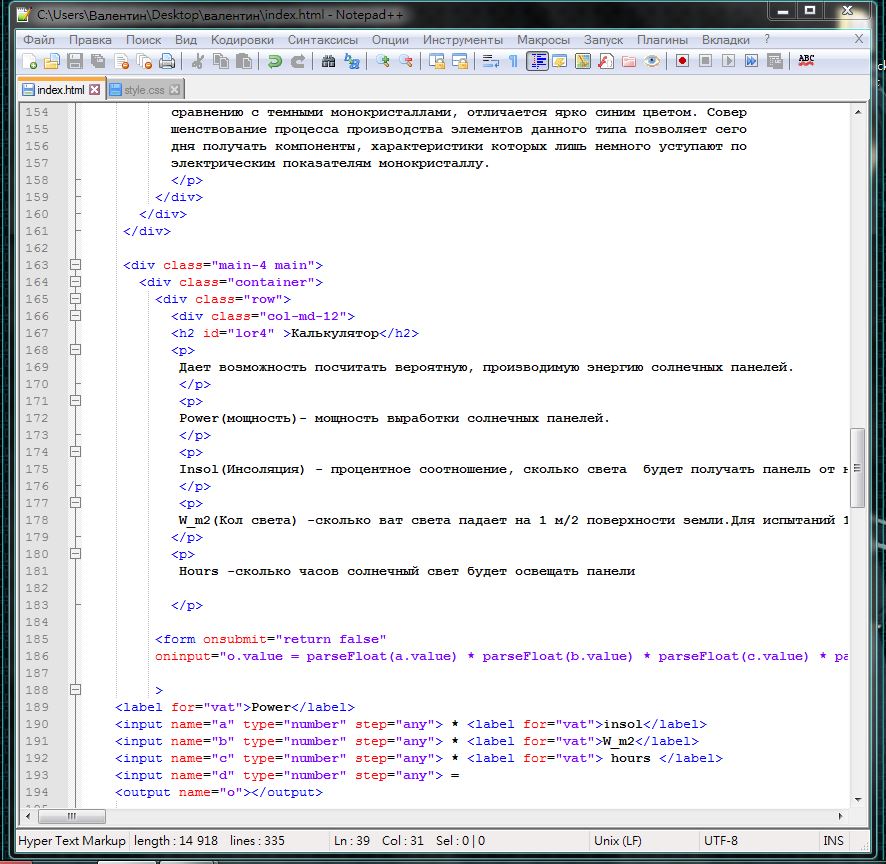


Рис 18 калькулятор

На фото здесь и ниже представлен фрагмент калькулятора. Расчет, форма.

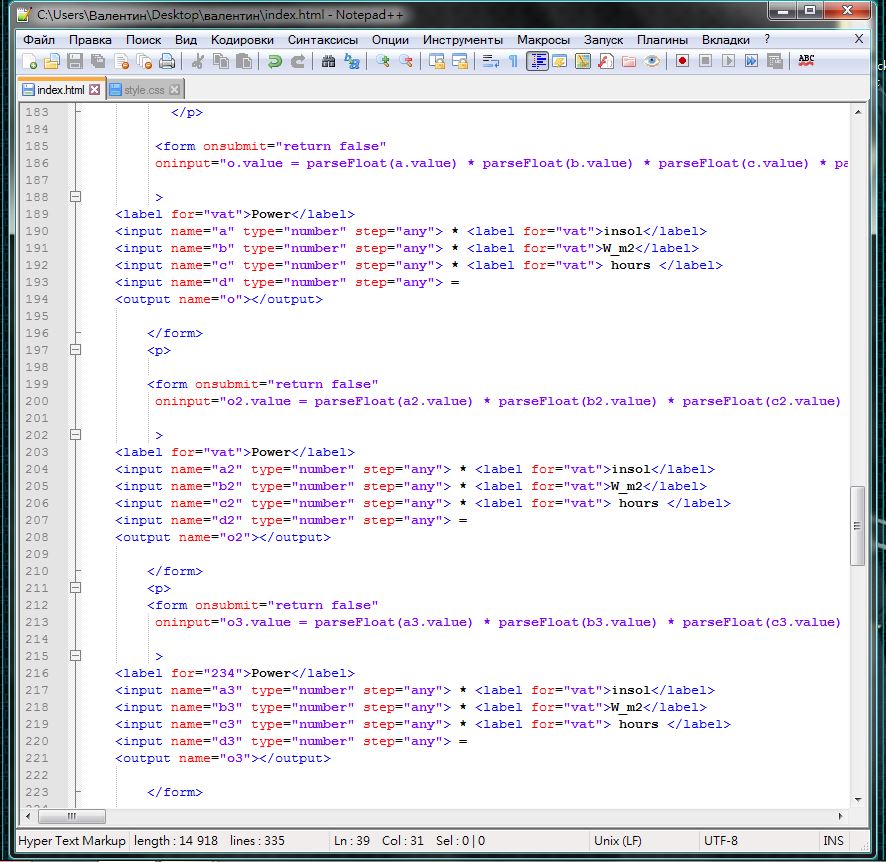


Рис 19 калькулятор

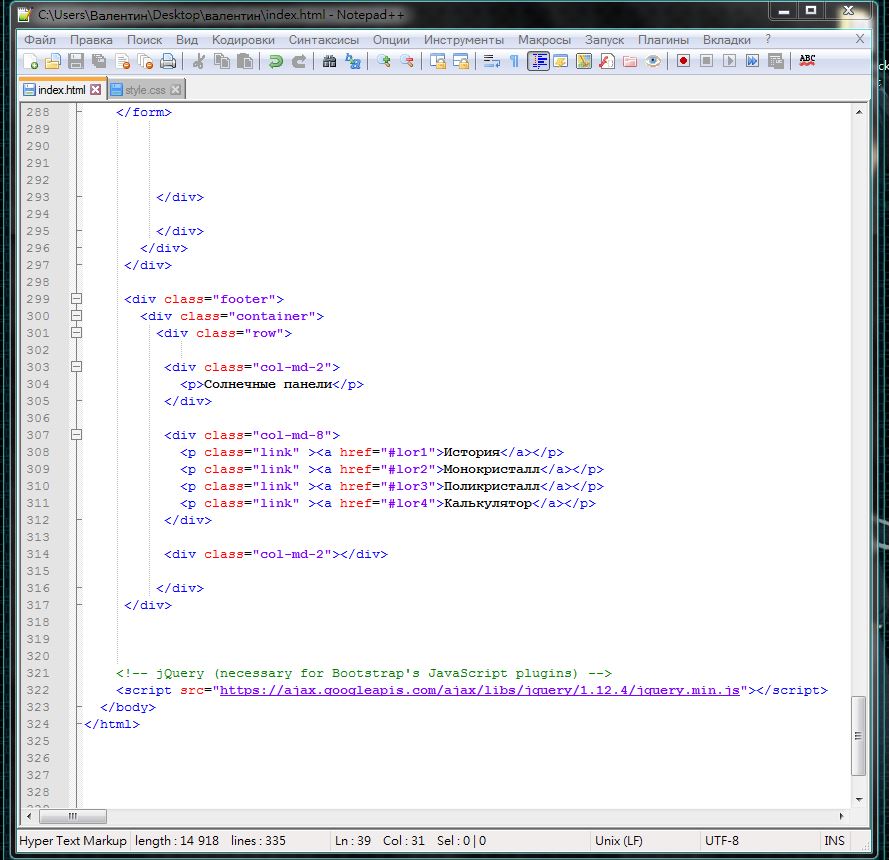


Рис 20 нижняя шапка меню.

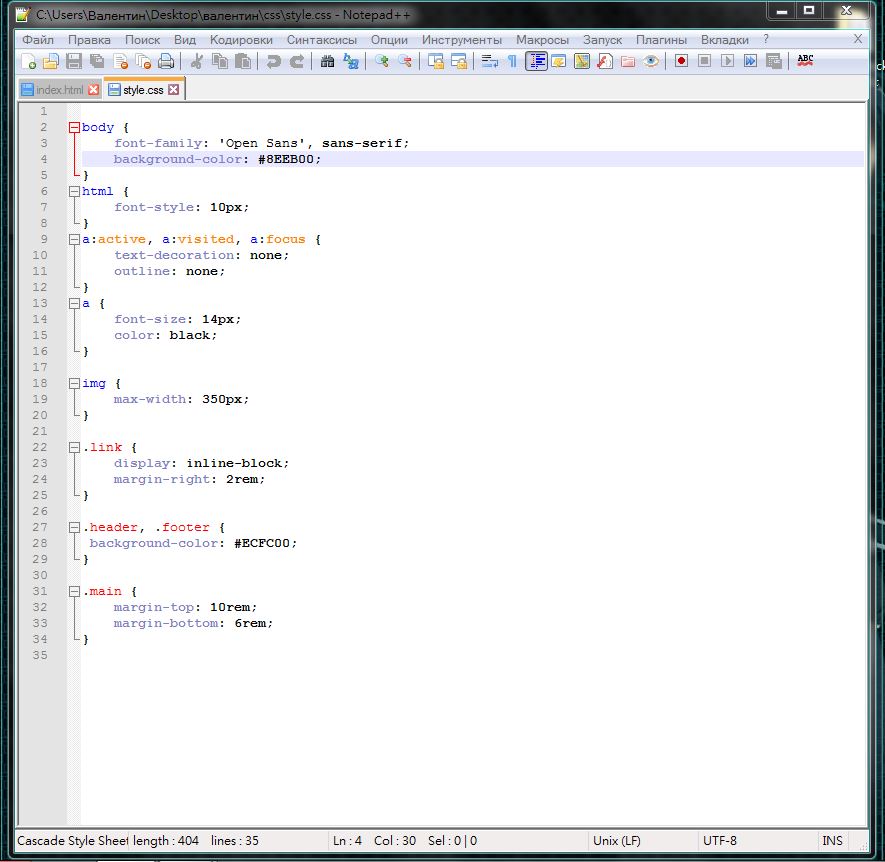


Рис21 CSS

Файл CSS. Здесь основные настройки цвета меню, текста, заднего фона, отступы.

# 2.4 Руководство администратора

Программа администратора. С помощью неё, он может вносить, удалять, изменять данные в БД. Производить расчеты вручную или с помощью данных из БД.

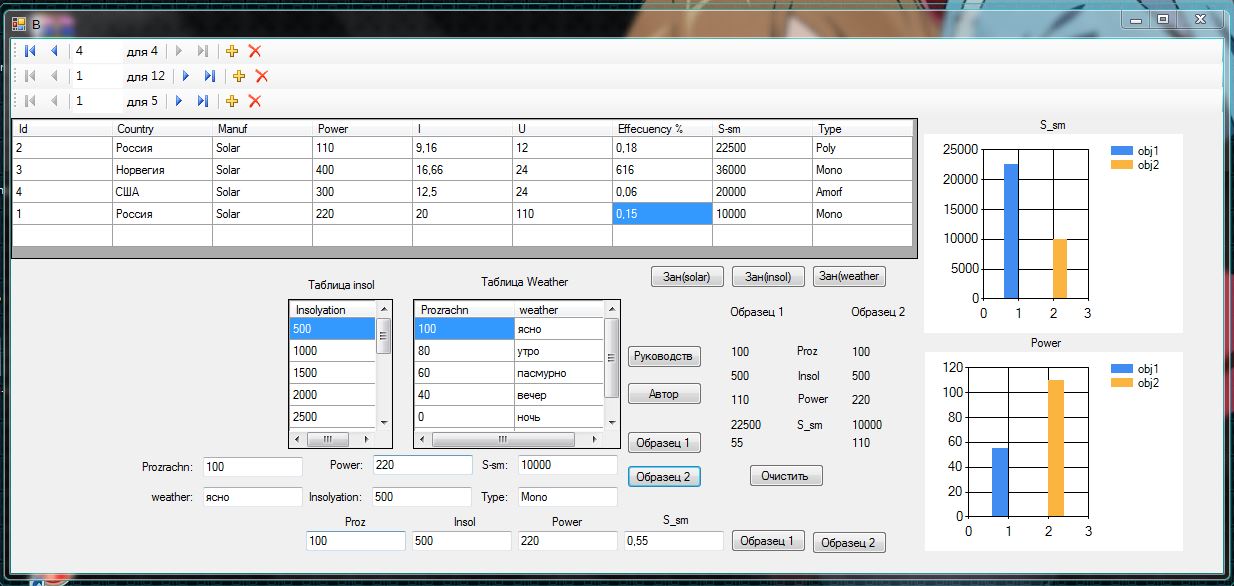


Рис 22 администратор

На верху есть три панели навигатора. Служат для навигации по табличкам(Солнечная панель, Инсоляция, Погода), добавление, удаление. Для редактирования можно кликнуть несколько раз по табличке. Есть кнопка руководства. Небольшая справка. Для сохранения информации нужно кликнуть по кнопке Зан (наименование таблицы).Внизу так же есть раздел для расчета данных , которые нужны администратору

# 2.5. Руководство пользователя

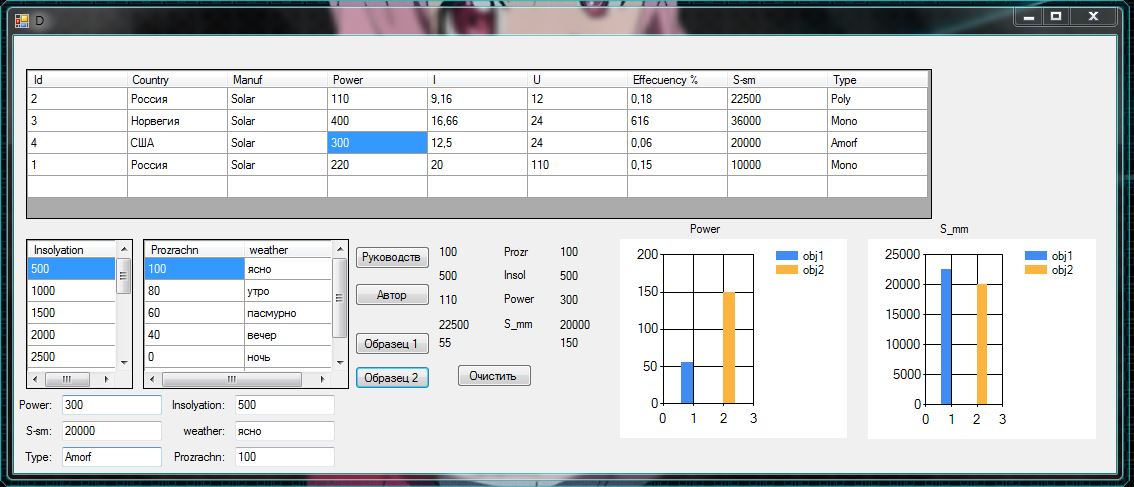
Так выглядит окно пользователя, в котором можно сделать расчет-симуляцию работы солнечной панели в реальных условиях. Выбрав поля: Погода, Солнечная панель, Инсоляция нажать кнопку Образец и выдаст результат.

Рис 23пользователь

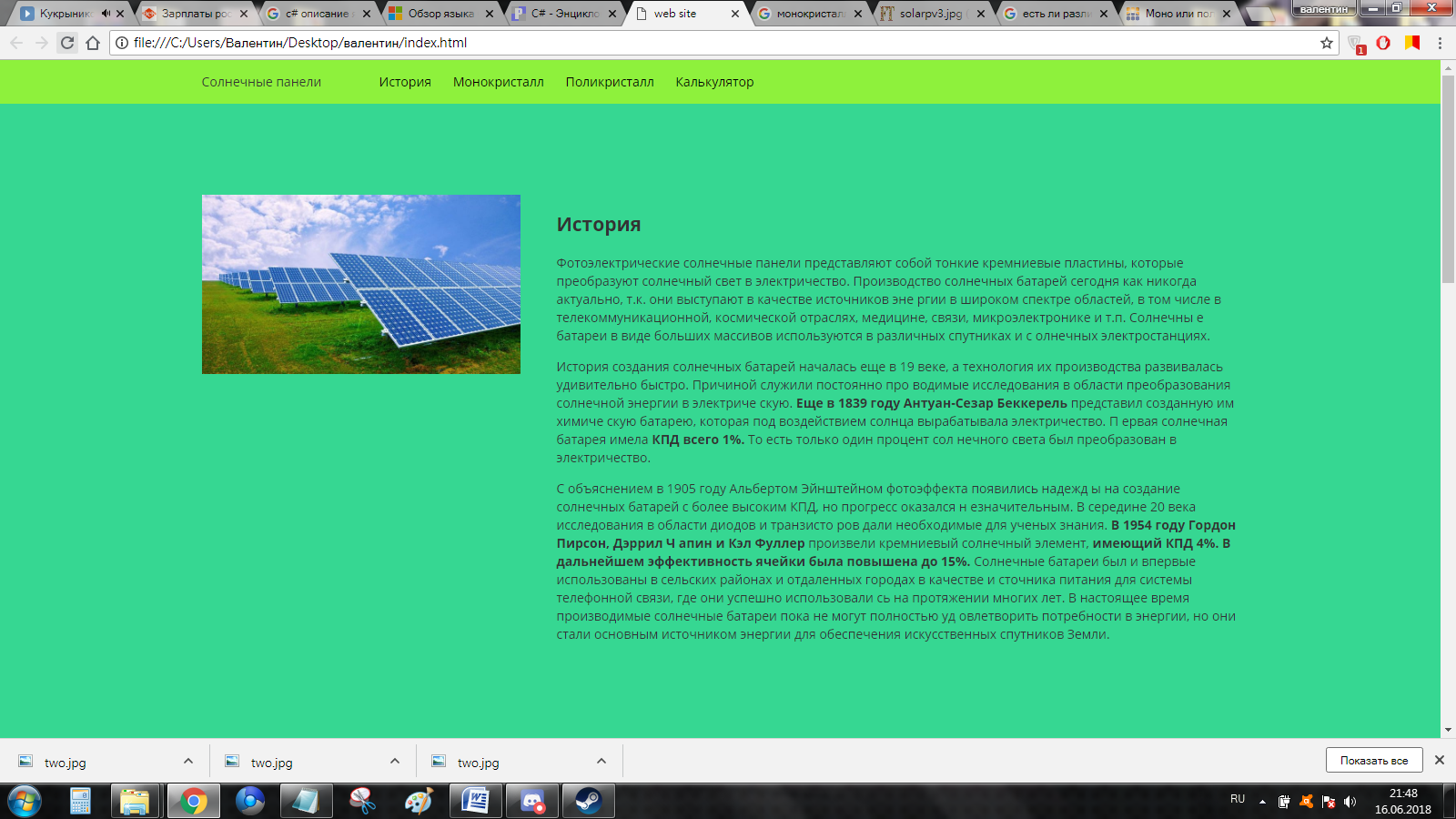


Рис 24 Верх сайта

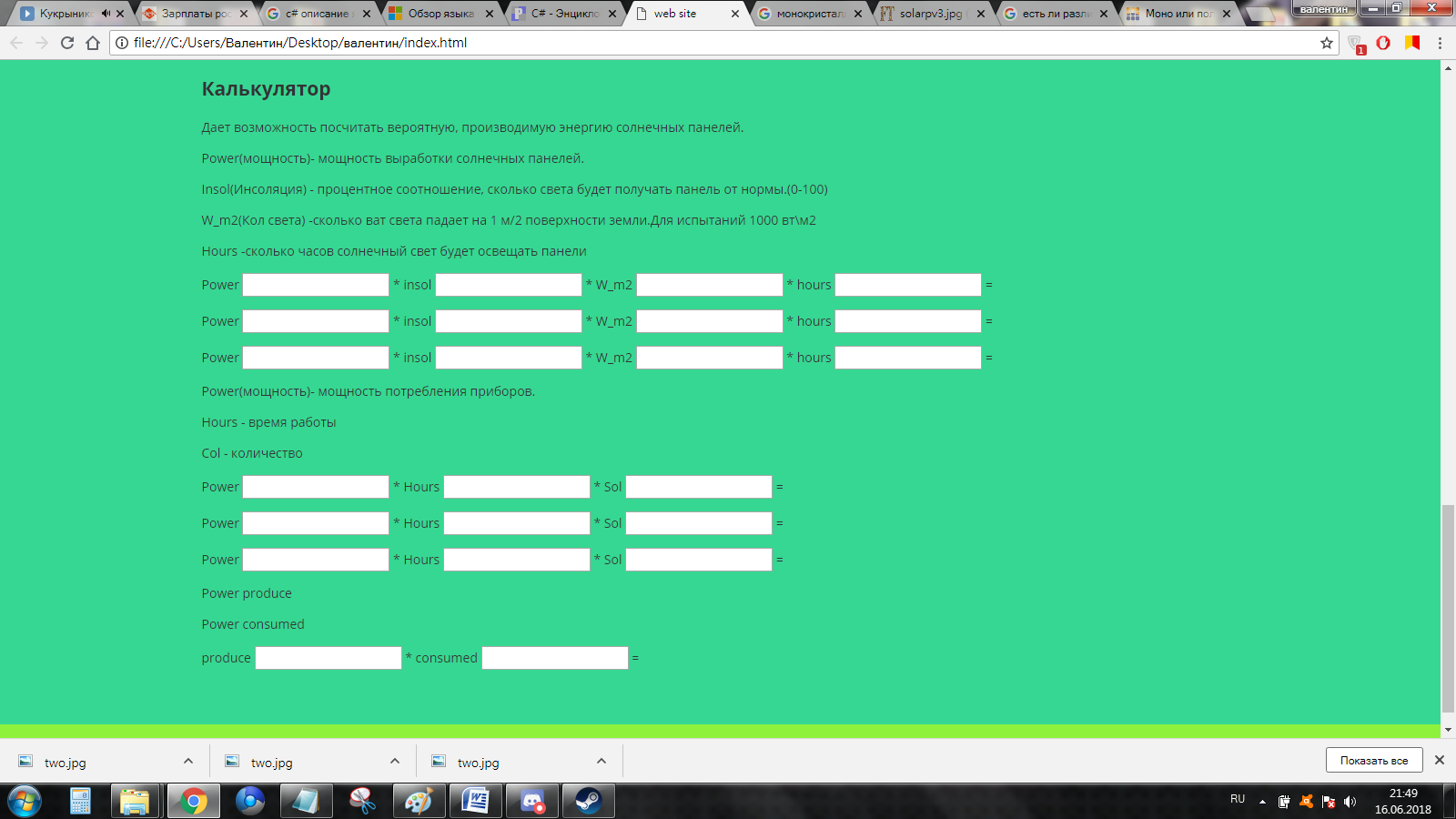


Рис 25 Калькулятор

# Заключение

В ходе работы над курсовой работой было прочитано и проанализированное много ресурсов. Полностью с чистого листа шло освоение C#.

Освоено Visual Studio 2017. Решена проблема с рисованием графиков в элементах chart. Углубленны знания в верстке сайтов и элементов верстки:HTML, JS, CSS.Так же был освоен Bootstrap 3- фреймворк для быстрой верстки сайтов с адаптацией под любые разрешения. Отдельно были получены знания по дизайну. Прочитано много ресурсов по работе солнечных панелей. В ходе погружения в тему работы был найден вопрос по падению производительности солнечных панелей от тени или пасмурной погоды в 2-3 раза. В дальнейшем при чтении технической литературы, форумов, тестов производителей, данный феномен не подтвердился и является ошибочным. Приложение дает возможность посчитать и сделать симуляцию работы солнечных панелей, с сравнительными графиками. Для удобства есть хранение в БД, которая дает возможность сохранять параметры для частой работы. Приложение и сайт отвечают основным требованиям функциональности и могут быть опубликованы в сети интернет.

# Список литературы

1) Албахари Дж., Албахари Б.C# 6.0. Справочник. Полное описание языка. 6-е издание, Издательство: Вильямс Издательский дом, 2016, 1040 с.

2) [Астахова](https://www.ozon.ru/person/1058431/) Ирина, [Вадим Мельников](https://www.ozon.ru/person/1058436/), [Александр Толстобров](https://www.ozon.ru/person/1058433/), [Вадим Фертиков](https://www.ozon.ru/person/1458689/), СУБД. Язык SQL в примерах и задачах. Москва, 2007, 435 с.

3) Албахари Д., Албахари Б. C# 7.0. Справочник. Полное описание языка, Издательство: Альфа - книга, 2018, 1024с.

4) Бекаревич Ю., Пушкина Н.Самоучитель Microsoft Access 2013, Издательство: БХВ-Петербург, 2014, 464 с.

5) Винарский Я., Гутгарц Р. Web-аппликации в интернет-маркетинге: проектирование, создание и применение. Практическое пособие, Издательство: Инфра-М, 2018, 269 с.

6) Дунаев В. В. HTML, скрипты и стили. Спб.: БХВ – Петербург, 2011. 603 с.

7) Дронов В.HTML 5 CSS 3 и Web 2.0 Разработка современных Web-сайтов, Издательство: БХВ-Петербург, 2011, 416 с.

8) Евдокимов П. C# на примерах , Издательство: Наука и Техника СПб,2018, 320 с.

9) Кириллов В.В. Введение в реляционные базы данных. Санкт-Петербург, Издательство «БХВ-Петербург», 2012, 464 с.

10) Прайс М. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов, Питер СПб, 2018, 640 с.

11) Сурядный А.Microsoft Access 2010. Лучший самоучитель, Издательство: Астрель, 2012, 448 с.

12) Фримен. Э. Изучаем HTML, XHTML и CSS. СПб.: Питер, 2013. 608 c.

13) Чебыкин Р. Самоучитель HTML и CSS. Современные технологии. М: Огни, 2012. 624 c.

14) Microsoft Office Access 2003. Шаг за шагом. Москва, Издательство «Эком», 2010, 368 с.

# Приложения

**Код приложения пользователя.**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Расчет

{ public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet4.Weather". При необходимости она может быть перемещена или удалена.

this.weatherTableAdapter1.Fill(this.тестDataSet4.Weather);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet4.Insolation". При необходимости она может быть перемещена или удалена.

this.insolationTableAdapter1.Fill(this.тестDataSet4.Insolation);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet4.Solar". При необходимости она может быть перемещена или удалена.

this.solarTableAdapter1.Fill(this.тестDataSet4.Solar);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Weather". При необходимости она может быть перемещена или удалена.

this.weatherTableAdapter.Fill(this.тестDataSet3.Weather);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Insolation". При необходимости она может быть перемещена или удалена.

this.insolationTableAdapter.Fill(this.тестDataSet3.Insolation);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Solar". При необходимости она может быть перемещена или удалена.

this.solarTableAdapter.Fill(this.тестDataSet3.Solar);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Weather". При необходимости она может быть перемещена или удалена.

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet1.Solar". При необходимости она может быть перемещена или удалена.

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

double pr1, in1, po1, s1, r1 = 0;

pr1 =Convert.ToDouble(prozrachnTextBox.Text);

in1 = Convert.ToDouble(insolyationTextBox.Text);

po1 = Convert.ToDouble(powerTextBox.Text);

s1 = Convert.ToDouble(s\_smTextBox.Text);

r1 = pr1\*in1\*po1/100/1000;

label2.Text = prozrachnTextBox.Text;

label3.Text = insolyationTextBox.Text;

label4.Text = powerTextBox.Text;

label5.Text = s\_smTextBox.Text;

label11.Text = Convert.ToString(r1);

chart1.Series["obj1"].Points.AddXY(1,Convert.ToInt64(r1));

chart2.Series["obj1"].Points.AddXY(1, Convert.ToInt64(s1));

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void prozrachnTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

double pr2, in2, po2, s2, r2 = 0;

pr2 = Convert.ToDouble(prozrachnTextBox.Text);

in2 = Convert.ToDouble(insolyationTextBox.Text);

po2 = Convert.ToDouble(powerTextBox.Text);

s2 = Convert.ToDouble(s\_smTextBox.Text);

r2 = pr2 \* in2 \* po2 / 100 / 1000;

label9.Text = prozrachnTextBox.Text;

label8.Text = insolyationTextBox.Text;

label7.Text = powerTextBox.Text;

label6.Text = s\_smTextBox.Text;

label12.Text = Convert.ToString(r2);

chart1.Series["obj2"].Points.AddXY(2, Convert.ToInt64(r2));

chart2.Series["obj2"].Points.AddXY(2, Convert.ToInt64(s2));

}

private void chart1\_Click(object sender, EventArgs e)

{

chart1.Series["obj1"].Points.AddY(Convert.ToInt64(label11.Text));

chart1.Series["obj2"].Points.AddY(Convert.ToInt64(label12.Text));

}

private void button3\_Click(object sender, EventArgs e)

{

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

chart2.Series[0].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

chart2.Series[1].Points.Clear();

}

private void button4\_Click(object sender, EventArgs e)

{

MessageBox.Show(" Доброго времени суток) Создатель Критенко Валентин Григорьевич ");

}

private void button5\_Click(object sender, EventArgs e)

{

MessageBox.Show(" Программа дает возможность произвести расчет-симуляцию солнечных панелей. Выбрать данные в табличках. Нажать Образец 1 или 2. Данные занесутся в график. Выбрать другие данные и занести другой образец. После очистить график. Solar- таблица с характеристиками панели. Insol таблица - кол солнечного света на м2. Weather таблица - погодные условия и коэффециент прозрачности от нормальных условий. ");

}

private void label14\_Click(object sender, EventArgs e)

{

}

private void dataGridView2\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

}

private void s\_smLabel\_Click(object sender, EventArgs e)

{

}

private void s\_smTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void button6\_Click(object sender, EventArgs e)

{

textBox1.Text = "https://yadi.sk/d/PTLsaCGp3Y2fod";

}

}

}

**Код приложения администратора**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Расчет

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Weather". При необходимости она может быть перемещена или удалена.

this.weatherTableAdapter.Fill(this.тестDataSet3.Weather);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Insolation". При необходимости она может быть перемещена или удалена.

this.insolationTableAdapter.Fill(this.тестDataSet3.Insolation);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Solar". При необходимости она может быть перемещена или удалена.

this.solarTableAdapter.Fill(this.тестDataSet3.Solar);

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Weather". При необходимости она может быть перемещена или удалена.

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet1.Weather". При необходимости она может быть перемещена или удалена.

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet3.Weather". При необходимости она может быть перемещена или удалена.

// TODO: данная строка кода позволяет загрузить данные в таблицу "тестDataSet.Insolation". При необходимости она может быть перемещена или удалена.

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

double pr1, in1, po1, s1, r1 = 0;

pr1 =Convert.ToDouble(prozrachnTextBox.Text);

in1 = Convert.ToDouble(insolyationTextBox.Text);

po1 = Convert.ToDouble(powerTextBox.Text);

s1 = Convert.ToDouble(s\_smTextBox.Text);

r1 = pr1\*in1\*po1/100/1000;

label2.Text = prozrachnTextBox.Text;

label3.Text = insolyationTextBox.Text;

label4.Text = powerTextBox.Text;

label5.Text = s\_smTextBox.Text;

label11.Text = Convert.ToString(r1);

chart1.Series["obj1"].Points.AddXY(1,Convert.ToInt64(r1));

chart2.Series["obj1"].Points.AddXY(1, Convert.ToInt64(s1));

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void prozrachnTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

double pr2, in2, po2, s2, r2 = 0;

pr2 = Convert.ToDouble(prozrachnTextBox.Text);

in2 = Convert.ToDouble(insolyationTextBox.Text);

po2 = Convert.ToDouble(powerTextBox.Text);

s2 = Convert.ToDouble(s\_smTextBox.Text);

r2 = pr2 \* in2 \* po2 / 100 / 1000;

label9.Text = prozrachnTextBox.Text;

label8.Text = insolyationTextBox.Text;

label7.Text = powerTextBox.Text;

label6.Text = s\_smTextBox.Text;

label12.Text = Convert.ToString(r2);

chart1.Series["obj2"].Points.AddXY(2,Convert.ToInt64(r2));

chart2.Series["obj2"].Points.AddXY(2, Convert.ToInt64(s2));

}

private void chart1\_Click(object sender, EventArgs e)

{

chart1.Series["obj1"].Points.AddY(Convert.ToInt64(label11.Text));

chart2.Series["obj2"].Points.AddY(Convert.ToInt64(label12.Text));

}

private void button3\_Click(object sender, EventArgs e)

{

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

chart2.Series[0].Points.Clear();

chart2.Series[1].Points.Clear();

}

private void button4\_Click(object sender, EventArgs e)

{

MessageBox.Show(" Доброго времени суток) Создатель Критенко Валентин Григорьевич ");

}

private void button5\_Click(object sender, EventArgs e)

{

MessageBox.Show(" Внимание! Поля чувствительны к вводимым данным. Поля с числами используют значения (2,13561). Ввести данные в поля или занести из(в) Табличек(и). Выбрать образец 1 ил 2.Ввурху есть панель навигации. Помогает работать с данными в таблицах.Для сохранения изменений нужно нажать на кнопку зан(таблица). Расположение таблиц-навигаторов(сверху - вниз, слева на право): Solar,Insol,Weather. Производит расчет-симуляцию. По заданным критериям. Solar- таблица с характеристиками панели. Insol таблица - кол солнечного света на м2. Weather таблица - погодные условия и коэффециент прозрачности от нормальных условий. ");

}

private void button6\_Click(object sender, EventArgs e)

{

this.solarTableAdapter.Update (this.тестDataSet3.Solar);

}

private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)

{

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

}

private void button6\_Click\_1(object sender, EventArgs e)

{

this.solarTableAdapter.Update(this.тестDataSet3.Solar);

}

private void toolStripButton1\_Click\_1(object sender, EventArgs e)

{

this.solarTableAdapter.Update(this.тестDataSet3.Solar);

}

private void button7\_Click(object sender, EventArgs e)

{

this.insolationTableAdapter.Update(this.тестDataSet3.Insolation);

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

this.insolationTableAdapter.Update(this.тестDataSet3.Insolation);

}

private void button8\_Click(object sender, EventArgs e)

{

this.weatherTableAdapter.Update(this.тестDataSet3.Weather);

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

this.weatherTableAdapter.Update(this.тестDataSet3.Weather);

}

private void button9\_Click(object sender, EventArgs e)

{

double pr1, in1, po1, s1, r1 = 0;

pr1 = Convert.ToDouble(textBox1.Text);

in1 = Convert.ToDouble(textBox2.Text);

po1 = Convert.ToDouble(textBox3.Text);

s1 = Convert.ToDouble(textBox4.Text);

r1 = pr1 \* in1 \* po1 / 100 / 1000;

label2.Text = prozrachnTextBox.Text;

label3.Text = insolyationTextBox.Text;

label4.Text = powerTextBox.Text;

label5.Text = s\_smTextBox.Text;

label11.Text = Convert.ToString(r1);

chart1.Series["obj1"].Points.AddXY(1, Convert.ToInt64(r1));

chart2.Series["obj1"].Points.AddXY(1, Convert.ToInt64(s1));

}

private void button10\_Click(object sender, EventArgs e)

{

double pr2, in2, po2, s2, r2 = 0;

pr2 = Convert.ToDouble(textBox1.Text);

in2 = Convert.ToDouble(textBox2.Text);

po2 = Convert.ToDouble(textBox3.Text);

s2 = Convert.ToDouble(textBox4.Text);

r2 = pr2 \* in2 \* po2 / 100 / 1000;

label9.Text = prozrachnTextBox.Text;

label8.Text = insolyationTextBox.Text;

label7.Text = powerTextBox.Text;

label6.Text = s\_smTextBox.Text;

label12.Text = Convert.ToString(r2);

chart1.Series["obj2"].Points.AddXY(2, Convert.ToInt64(r2));

chart2.Series["obj2"].Points.AddXY(2, Convert.ToInt64(s2));

}

private void button11\_Click(object sender, EventArgs e)

{

textBox5.Text = "https://yadi.sk/d/PTLsaCGp3Y2fod";

}

}

}

**Сайт**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- meta для соцсетей -->

<meta property="og:url" content="">

<meta property="og:type" content="website">

<meta property="og:title" content="">

<meta property="og:image" content="">

<meta property="og:description" content="">

<meta property="og:site\_name" content="">

<meta property="article:author" content="">

<title>web site</title>

<link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700&amp;subset=cyrillic" rel="stylesheet">

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="css/style.css" rel="stylesheet">

<!-- Фавиконка -->

<link rel="shortcut icon" type="image/" href="favicon.ico">

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<div class="header">

<div class="container">

<div class="row">

<div class="col-md-2">

<p>Солнечные панели</p>

</div>

<div class="col-md-8">

<p class="link" ><a href="#lor1">История</a></p>

<p class="link" ><a href="#lor2">Монокристалл</a></p>

<p class="link" ><a href="#lor3">Поликристалл</a></p>

<p class="link" ><a href="#lor4">Калькулятор</a></p>

</div>

<div class="col-md-2"></div>

</div>

</div>

</div>

<div class="main-1 main">

<div class="container">

<div class="row">

<div class="col-md-4">

<img src="img/one.jpg" alt="">

</div>

<div class="col-md-8">

<h2 id="lor1" >История</h2>

<p>

Фотоэлектрические солнечные панели представляют собой тонкие кремниевые пластины, которые преобразуют солнечный свет в электричество. Производство солнечных батарей сегодня как никогда актуально, т.к. они выступают в качестве источников энергии в широком спектре областей, в том числе в телекоммуникационной, космической отраслях, медицине, связи, микроэлектронике и т.п. Солнечны

е батареи в виде больших массивов используются в различных спутниках и солнечных электростанциях.

</p>

<p>

История создания солнечных батарей началась еще в 19 веке, а технология их производства развивалась удивительно быстро. Причиной служили постоянно про водимые исследования в области преобразования солнечной энергии в электрическую. <b>Еще в 1839 году Антуан-Сезар Беккерель</b> представил созданную им химическую батарею, которая под воздействием солнца вырабатывала электричество. Первая солнечная батарея имела <b>КПД всего 1%.</b> То есть только один процент солнечного света был преобразован в электричество. </p>

<p>

С объяснением в 1905 году Альбертом Эйнштейном фотоэффекта появились надежды на создание солнечных батарей с более высоким КПД, но прогресс оказался незначительным. В середине 20 века исследования в области диодов и транзисторов дали необходимые для ученых знания. <b>В 1954 году Гордон Пирсон, Дэррил Чапин и Кэл Фуллер</b> произвели кремниевый солнечный элемент, <b>имеющий КПД 4%. Вдальнейшем эффективность ячейки была повышена до 15%.</b> Солнечные батареи был

и впервые использованы в сельских районах и отдаленных городах в качестве источника питания для системы телефонной связи, где они успешно использовались на протяжении многих лет. В настоящее время производимые солнечные батареи пока не могут полностью удовлетворить потребности в энергии, но они стали основным источником энергиидля обеспечения искусственных спутников Земли.

</p>

</div>

</div>

</div>

</div>

<div class="main-2 main">

<div class="container">

<div class="row">

<div class="col-md-4">

<img src="img/two.jpg" alt="">

</div>

<div class="col-md-8">

<h2 id="lor2" >Монокристалл</h2>

<p>

<b>Наиболее распространенный и популярный вид солнечных батарей солнечные батареи из монокристаллического кремния. </b>

Их получают литьем кристаллов кремния высокой чистоты, при котором расплав отвердевает при контакте с затравкой кристалла. В процессе охлаждения кремний постепенно застывает в форме цилиндрической отливки монокристалл а диаметром 13 — 20 см, длина которого достигает 200 см. Получаемый таким образом слиток нарезается листочками толщиной 250 — 300 мкм. Такие элементы имеют более высокую эффективность по сравнению с элементами, вырабатываемыми другими способами, <b>КПД достигает 19 %</b>, благодаря особой ориентации атомов монокристалла, которая способствует росту подвижности электронов. Кремний пронизывает сетка из металлических электродов. Традиционно монокристаллические модули вставлены в алюминиевую рамку и закрыты противоударным стеклом. Цвет монокристаллических фото-элементов — темно-синий или черный.

</p>

<p>

Солнечные батареи надежны, долговечны <b>(срок службы до 50 лет)</b> и просты в установке, так как не содержат движущихся частей. Солнечные батареи можно использовать, где плохо работает обычное энергоснабжение и большое количество солнечных дней.

</p>

</div>

</div>

</div>

</div>

<div class="main-3 main">

<div class="container">

<div class="row">

<div class="col-md-4">

<img src="img/three.jpg" alt="">

</div>

<div class="col-md-8">

<h2 id="lor3" >Поликристал</h2>

</div>

<p>

<b>Солнечные батареи из поликристаллического</b>

кремния. Альтернативой монокристаллического кремния является поликристаллический кремний. У него более низкая себестоимость. <b>Кпд составляет 13-16%</b>Кристаллы в нем ещё агрегатные, но имеют различную форму и ориентацию. Этот материал, по сравнению с темными монокристаллами, отличается ярко синим цветом. Совершенствование процесса производства элементов данного типа позволяет сего дня получать компоненты, характеристики которых лишь немного уступают по электрическим показателям монокристаллу.

</p>

</div>

</div>

</div>

<div class="main-4 main">

<div class="container">

<div class="row">

<div class="col-md-12">

<h2 id="lor4" >Калькулятор</h2>

<p>

Дает возможность посчитать вероятную, производимую энергию солнечных панелей.

</p>

<p>

Power(мощность)- мощность выработки солнечных панелей.

</p>

<p>

Insol(Инсоляция) - процентное соотношение, сколько света будет получать панель от нормы.(0-100)

</p>

<p>

W\_m2(Кол света) -сколько ват света падает на 1 м/2 поверхности земли.Для испытаний 1000 вт\м2

</p>

<p>

Hours -сколько часов солнечный свет будет освещать панели

</p>

<form onsubmit="return false"

oninput="o.value = parseFloat(a.value) \* parseFloat(b.value) \* parseFloat(c.value) \* parseFloat(d.value) / 100000"

>

<label for="vat">Power</label>

<input name="a" type="number" step="any"> \* <label for="vat">insol</label>

<input name="b" type="number" step="any"> \* <label for="vat">W\_m2</label>

<input name="c" type="number" step="any"> \* <label for="vat"> hours </label>

<input name="d" type="number" step="any"> =

<output name="o"></output>

</form>

<p>

<form onsubmit="return false"

oninput="o2.value = parseFloat(a2.value) \* parseFloat(b2.value) \* parseFloat(c2.value) \* parseFloat(d2.value) / 100000"

>

<label for="vat">Power</label>

<input name="a2" type="number" step="any"> \* <label for="vat">insol</label>

<input name="b2" type="number" step="any"> \* <label for="vat">W\_m2</label>

<input name="c2" type="number" step="any"> \* <label for="vat"> hours </label>

<input name="d2" type="number" step="any"> =

<output name="o2"></output>

</form>

<p>

<form onsubmit="return false"

oninput="o3.value = parseFloat(a3.value) \* parseFloat(b3.value) \* parseFloat(c3.value) \* parseFloat(d3.value) / 100000"

>

<label for="234">Power</label>

<input name="a3" type="number" step="any"> \* <label for="vat">insol</label>

<input name="b3" type="number" step="any"> \* <label for="vat">W\_m2</label>

<input name="c3" type="number" step="any"> \* <label for="vat"> hours </label>

<input name="d3" type="number" step="any"> =

<output name="o3"></output>

</form>

<p>

Power(мощность)- мощность потребления приборов.

</p>

<p>

Hours - время работы

</p>

<p>

Col - количество

</p>

<p>

<form onsubmit="return false"

oninput="o4.value = parseFloat(a4.value) \* parseFloat(b4.value) \* parseFloat(c4.value) "

>

<label for="234">Power</label>

<input name="a4" type="number" step="any"> \* <label for="vat">Hours</label>

<input name="b4" type="number" step="any"> \* <label for="vat">Sol</label>

<input name="c4" type="number" step="any"> =

<output name="o4"></output>

</form>

<p>

<form onsubmit="return false"

oninput="o5.value = parseFloat(a5.value) \* parseFloat(b5.value) \* parseFloat(c5.value) "

>

<label for="234">Power</label>

<input name="a5" type="number" step="any"> \* <label for="vat">Hours</label>

<input name="b5" type="number" step="any"> \* <label for="vat">Sol</label>

<input name="c5" type="number" step="any"> =

<output name="o5"></output>

</form>

<p>

<form onsubmit="return false"

oninput="o6.value = parseFloat(a6.value) \* parseFloat(b6.value) \* parseFloat(c6.value) "

>

<label for="234">Power</label>

<input name="a6" type="number" step="any"> \* <label for="vat">Hours</label>

<input name="b6" type="number" step="any"> \* <label for="vat">Sol</label>

<input name="c6" type="number" step="any"> =

<output name="o6"></output>

</form>

<p>

Power produce - произведено

</p>

<p>

Power consumed - потреблено

</p>

<p>

<form onsubmit="return false"

oninput="o6.value = parseFloat(a6.value) / parseFloat(b6.value) "

>

<label for="Produce">produce</label>

<input name="a6" type="number" step="any"> \* <label for="vat">consumed</label>

<input name="b6" type="number" step="any"> =

<output name="o6"></output>

</form>

<p><b>

Ссылка на ПО. Позволяет производить расчеты в приложении : (https://yadi.sk/d/PTLsaCGp3Y2fod)

</p></b>

</div>

</div>

</div>

</div>

<div class="footer">

<div class="container">

<div class="row">

<div class="col-md-2">

<p>Солнечные панели</p>

</div>

<div class="col-md-8">

<p class="link" ><a href="#lor1">История</a></p>

<p class="link" ><a href="#lor2">Монокристалл</a></p>

<p class="link" ><a href="#lor3">Поликристалл</a></p>

<p class="link" ><a href="#lor4">Калькулятор</a></p>

</div>

<div class="col-md-2"></div>

</div>

</div>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

</body>

</html>

**Style.css**

body {

font-family: 'Open Sans', sans-serif;

background-color: #36D792;

}

html {

font-style: 10px;

}

a:active, a:visited, a:focus {

text-decoration: none;

outline: none;

}

a {

font-size: 14px;

color: black;

}

img {

max-width: 350px;

}

.link {

display: inline-block;

margin-right: 2rem;

}

.header, .footer {

background-color: #8EF13C;

}

.main {

margin-top: 10rem;

margin-bottom: 6rem;

}