МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий

Кафедра математического и программного обеспечения ЭВМ

КУРСОВАЯ РАБОТА

по дисциплине «Программирование, модуль ООП»

на тему Объектно-ориентированное программирование на языке C++

Выполнил студент группы

1ПИб-01-31оп

*группа*

направления подготовки

09.03.04 Программная инженерия

*шифр, наименование*

Краев Алексей Александрович

*фамилия, имя, отчество*

Руководитель

Ершов Евгений Валентинович

*фамилия, имя, отчество*

Заведующий кафедрой ИИТ

*должность*

Дата представления работы

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

Заключение о допуске к защите

Оценка \_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*количество баллов*

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Череповец, 2020

Содержание:

[Введение 3](#_Toc31035443)

[1. Объектно-ориентированный анализ предметной области 5](#_Toc31035444)

[2. Проектирование классов 7](#_Toc31035445)

[3. Логическое структура программы 9](#_Toc31035446)

[4. Модульная структура программы. 11](#_Toc31035447)

[5. Тестирование программы 19](#_Toc31035448)

[Заключение 24](#_Toc31035449)

[Список литературы 25](#_Toc31035450)

[Приложение 1 26](#_Toc31035451)

[Приложение 2 31](#_Toc31035452)

[Приложение 3 32](#_Toc31035453)

# Введение

Объектно-ориентированное программирование (ООП) — это такой подход к программированию, основанный на представлении программы в виде совокупности объектов и компонентных функций, осуществляющих обработку, являющихся экземплярами определенного класса. ООП базируется на 3 принципах: инкапсуляция, наследование и полиморфизм.

Инкапсуляция — это механизм программирования, который позволяет скрыть внутреннее устройство класса и предоставить методы работы с экземпляром класса пользователю, призвана защитить от внешнего вмешательства.

Наследование — это процесс, посредством которого один объект может приобретать свойства другого. Точнее, новый или производный класс может быть определен на основе уже имеющегося класса, или базового. Наследование позволяет выстраивать иерархию классов.

Полиморфизм — это свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач. Целью полиморфизма, является использование одного имени для задания общих для класса действий. Выполнение каждого конкретного действия будет определяться типом данных. Полиморфизм позволяет манипулировать объектами различной степени сложности путём создания общего стандартного интерфейса с реализации похожих действий.

Полиморфная функция – это группа функций с одним и тем же именем, но выполняющие различные действия в зависимости от условий вызова.

Абстрактный класс – это класс, который может использоваться лишь в качестве базового класса для производных классов. Класс является абстрактным, если он содержит хотя бы одну абстрактную функцию.

Целью данной курсовой работы является создания программного продукта, предназначенного для работы с объектами предметной области "Энергетические установки". В ходе разработки были использованы принципы и возможности объектно-ориентированного программирования. При проектировании была создана иерархия родственных классов предметной области и были использованы механизмы наследования и принципы инкапсуляции, произведена полиморфная обработка объектов данных классов.

В расчетно-пояснительной записке описано функциональное назначение данного программного продукта, его логическая структура, а также принцип работы программы.

## Объектно-ориентированный анализ предметной области

Энергетические установки – это комплекс взаимосвязанного оборудования и сооружений, предназначенный для производства или преобразования, энергии. В данной работе установки были классифицированы по принципу их работы, разделение представлено на контекстной диаграмме классов(рис.1).



Рис.1 Контекстная диаграмма классов

Выделяют следующие энергетические установки:

* [Внутреннего](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE-%D0%BB%D1%83%D1%87%D0%B5%D0%B2%D0%B0%D1%8F_%D1%82%D1%80%D1%83%D0%B1%D0%BA%D0%B0) сгорания — это установки основанные на принципах сгорания углеводородного сырья в качестве источника энергии;
* [Альтернативные](https://ru.wikipedia.org/wiki/%D0%96%D0%9A-%D0%B4%D0%B8%D1%81%D0%BF%D0%BB%D0%B5%D0%B9) установки — это все установки, где в качестве источника энергии не выступает горения углеводородного сырья;
* Газо-турбинные — это установки, работающие на природном газе подаваемым вместе с кислородом на лопасти турбины и воспламеняемым там же;
* [ДВС (двигатель внутреннего сгорания)](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD) — энергетическая установка, в которой преобразования химической энергии в механическую (детонация) происходит в камере сгорания цилиндра;
* Роторный двигатель — энергетическая установка, в которой преобразования химической энергии в механическую (детонация) происходит в камере сгорания ротора;
* [Паро-турбинные](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD) — энергетическая установка, в которой источником механической энергии служит подаваемый на лопасти пар под высоким давлением;
* Электрические установки — энергетическая установка, в которой преобразуемым источником энергии является электричество.

## Проектирование классов

Исходя из составленной иерархии родственных классов были спроектированы следующие классы с компонентными функциями, полями данных, представленных в табл.1 и на детальной диаграмме классов(рис.2).

Для создания иерархии родственных типов используется механизм одиночного наследования. Количество уровней иерархии – 4.

Таблица.1

Классы предметной области “Энергетические установки”

|  |  |  |
| --- | --- | --- |
| Название класса | Компонентные функции | Поля(е) данных |
| CInterface | Вывод информации на экран и поиск экземпляра класса. | - |
| EnergyPlants | Конструктор, конструктор со всеми параметрами, копирующий конструктор, деструктор, перегрузка оператора =, запись объектов в файл, чтение объектов из файла, поиск объекта, вывод статуса объекта на экран, установка текущей мощности и ошибки, удаление объекта | Модель, габариты, максимальная мощность, наличие ошибки, текущая мощность, количество оборотов |
| Описанные ниже классы наследуют все поля и функции базового класса | | |
| InternalCombustion | Функция расчёта расхода топлива | Данные о топливе |
| GasTurbine | Функция анализа горения | Содержание кислорода |
| DVS | Функция определения типа двигателя | Тип топлива |
| Rotor | Функция сравнения с ДВС | Удельная мощность |
| AlternativePlants | Функция расчёта сезонной эффективности | Источник энергии |
| SteamTurbine | Функция расчёта теплопотери | Температура пара на входе, температура пара на выходе |
| ElectricPlants | Функция расчёта класса энергоэффективности | Использовано электроэнергии |



Рис.2 Детальная диаграмма классов

## Логическое структура программы

В программе предусмотрены следующие компонентные функции.

Search – выполняет поиск объекта в таблице по полю модель, является виртуальной. Print– выводит информацию об объекте, функция является виртуальной, работает механизм замещения для заданного класса. ExtractFromFile *-* загрузка из файла, в качестве формального параметра передается контейнер для хранения объектов и таблица для вывода информации об объектах. WriteToFile- сохранение в файл, в качестве формального параметра передается контейнер. Operator = - перегруженный оператор присваивания.

Добавление экземпляров классов предметной области производится вручную с клавиатуры или при загрузке из текстового файла, данные экземпляра заносятся в соответствующую таблицу. При нажатии кнопки "Записать в файл" объекты контейнера добавляются в файл.

Обработка исключительных ситуаций осуществляется при создании объектов классов и при загрузке, записи файлов.

Логическая структура программы представлена на рис.3.



Рис.3 Логическая структура программы

## Модульная структура программы.

Программа включает в себя 13 модулей, содержащая в себе описание каждого из классов. Модульная структура программы представлена на риc.4.



Рис.4 Модульная структура программы

Таблица 2. Выбор структур данных

|  |  |  |  |
| --- | --- | --- | --- |
| Имя модуля | Заголовок процедуры или функции | Формальные параметры | Выполняемое действие |
| Stack | void pop | - | Достать элемент из стэка |
| void push | <typename T> data | Добавить элемент в стэк |
| void clear | - | Очистить стэк |
| int GetSize | - | Получить размер стэка |
| T% operator[] | const int index | Перегрузка оператора [] |
| ~MyStack() | - | Деструктор |
| MyStack () | - | Конструктор по умолчанию |
| Node | <typename T> data | Конструктор с одним параметром |
| Node | T data, Node^ pNext | Конструктор со всеми параметрами |
| Node | Node% value | Копирующий конструктор |
| Vec (Container) | void pop\_back | - | Достать элемент из конца векотра |
| void push\_back | void\*data | Поместить элемент в конец вектора |
| void pop\_front | - | Достать элемент из начала вектора |
| void push\_front | void\* data | Поместить элемент в начало вектора |
| clear | - | Очистить список |
| GetSize | - | Получить размер списка |
| insert | void\*value, int index | Поместить элемент в нужную позицию списка |
| removeAt | int index | Удалить необходимый элемент списка |
| T% operator[] | const int index | Перегрузка оператора [] |
| MyVec() | - | Конструктор по умолчанию |
| ~ MyVec() | - | Деструктор |
| Node | void\* data | Конструктор с одним параметром |
| Node | void\* data, Node^ pNext | Конструктор со всеми параметрами |
| Node | Node% value | Копирующий конструктор |
| EnergyPlants | EnergyPlants() | - | Констркутор по умолчанию |
| EnergyPlants | String^ model,double maxPower, double length, double width, double height, double weight | Конструктор с параметрами |
| EnergyPlants | EnergyPlants%value | Копирующий конструктор |
| ~ EnergyPlants () | - | Деструктор |
| EnergyPlants %operator = | const EnergyPlants % right | Перегрузка оператора равно |
| static void WriteToFile | MyStack<EnergyPlants>% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack<EnergyPlants>% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack<EnergyPlants>% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack<EnergyPlants>% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack<EnergyPlants>% myqueue, DataGridView^ dgv | Статус установки |
| InternalCombustion | InternalCombustion () | - | Конструктор по умолчанию |
| InternalCombustion | String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel | Конструктор с параметрами |
| InternalCombustion | InternalCombustion % value | Копирующий конструктор |
| InternalCombustion % operator = | const InternalCombustion % right | Перегрузка оператора равно |
| ~ InternalCombustion () | - | Деструктор |
| static void WriteToFile | MyStack< InternalCombustion >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< InternalCombustion >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< InternalCombustion >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< InternalCombustion >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< InternalCombustion >% myqueue, DataGridView^ dgv | Статус установки |
| String^ CalculatedFuel() | - | Рассчет топлива |
| GasTurbine | GasTurbine () | - | Конструктор по умолчанию |
| GasTurbine | String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, double oxyConcentration | Конструктор с параметрами |
| GasTurbine | GasTurbine % value | Копирующий конструктор |
| GasTurbine % operator = | const GasTurbine % right | Перегрузка оператора равно |
| ~ GasTurbine () | - | Деструктор |
| static void WriteToFile | MyStack< GasTurbine >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< GasTurbine >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< GasTurbine >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< GasTurbine >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< GasTurbine >% myqueue, DataGridView^ dgv | Статус установки |
| String^ AnaliticsBurning(); | - | Анализ горения |
| DVS | DVS () | - | Конструктор по умолчанию |
| DVS | String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, String^ typeFuel | Конструктор с параметрами |
| DVS | DVS % value | Копирующий конструктор |
| DVS % operator = | const DVS % right | Перегрузка оператора равно |
| ~ DVS () | - | Деструктор |
| static void WriteToFile | MyStack< DVS >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< DVS >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< DVS >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< DVS >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< DVS >% myqueue, DataGridView^ dgv | Статус установки |
| String^ TypeDVS () | - | Тип ДВС |
| Rotor | Rotor () | - | Конструктор по умолчанию |
| Rotor | String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, double udPower | Конструктор с параметрами |
| Rotor | Rotor % value | Копирующий конструктор |
| Rotor % operator = | const Rotor % right | Перегрузка оператора равно |
| ~ Rotor () | - | Деструктор |
| static void WriteToFile | MyStack< Rotor >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< Rotor >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< Rotor >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< Rotor >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< Rotor >% myqueue, DataGridView^ dgv | Статус установки |
| String^ SravnSDVS () | - | Сравнение с ДВС |
| AlternativePlants | AlternativePlants () | - | Конструктор по умолчанию |
| AlternativePlants | String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy | Конструктор с параметрами |
| AlternativePlants | AlternativePlants % value | Копирующий конструктор |
| AlternativePlants % operator = | const AlternativePlants % right | Перегрузка оператора равно |
| ~ AlternativePlants () | - | Деструктор |
| static void WriteToFile | MyStack< AlternativePlants >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< AlternativePlants >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< AlternativePlants >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< AlternativePlants >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< AlternativePlants >% myqueue, DataGridView^ dgv | Статус установки |
| String^ SeasonEffective () | - | Сезонная эффективность |
| SteamTurbine | SteamTurbine () | - | Конструктор по умолчанию |
| SteamTurbine | String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy,double tempStreamIn, double tempStreamOut | Конструктор с параметрами |
| SteamTurbine | SteamTurbine % value | Копирующий конструктор |
| SteamTurbine % operator = | const SteamTurbine % right | Перегрузка оператора равно |
| ~ SteamTurbine() | - | Деструктор |
| static void WriteToFile | MyStack< SteamTurbine >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< SteamTurbine >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< SteamTurbine >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< SteamTurbine >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< SteamTurbine >% myqueue, DataGridView^ dgv | Статус установки |
| String^ TempRash() | - | Разница температур |
| ElectricPlants | ElectricPlants () | - | Конструктор по умолчанию |
| ElectricPlants | ElectricPlants(String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy, double usingEnergy | Конструктор с параметрами |
| ElectricPlants | ElectricPlants % value | Копирующий конструктор |
| ElectricPlants % operator = | const ElectricPlants % right | Перегрузка оператора равно |
| ~ ElectricPlants () | - | Деструктор |
| static void WriteToFile | MyStack< ElectricPlants >% mySt | Записьданных в файл |
| static void ExtractFromFile | MyStack< ElectricPlants >% mySt, DataGridView^ a | Чтение данных из файла |
| void Search override | DataGridView^ dgv, String^ str | Поиск объекта |
| void Print override | DataGridView^ dgv, int Rows | Вывод информации об объекте |
| static void PowerMoment | MyStack< ElectricPlants >% mySt,double pw, DataGridView^ dgv | Установка текущей мощности |
| static void Errors | MyStack< ElectricPlants >% mySt, DataGridView^ dgv | Установка ошибки |
| static void Status | MyStack< ElectricPlants >% myqueue, DataGridView^ dgv | Статус установки |
| String^ ClassEnergySave () | - | Класс энергоэффектиности |
| ClassInterface | virtual void Print | DataGridView ^dgv, int Rows | Вывод информации об объекте |
|  | virtual void Search | DataGridView^ dgv, String^ str | Поиск объекта |
| Cache | T% operator[] | const int index | Перегрузка оператора [] |

Выбор структур данных для решения поставленной задачи представлен в таблице 3.

Таблица 3. Выбор структур данных

|  |  |  |
| --- | --- | --- |
| Наименование | Обозначение | Тип данных |
| Модель | model | String^ |
| Габбариты | length,width,height,weight | double |
| Максимальная мощность | maxPower | double |
| Текущая мощность | powerMoment | double |
| Обороты | rotation | int |
| Наличие ошибки | er | bool |
| Тип матрицы | MatrixType | String^ |
| Стэк для объектов класса EnergyPlants | sEP | MyStack< EnergyPlants > |
| Стэк для объектов класса InternalCombustion | sIC | MyStack< InternalCombustion > |
| Стэк для объектов класса AlternativePlants | sAP | MyStack< AlternativePlants > |
| Стэк для объектов класса SteamTurbine | sST | MyStack< SteamTurbine > |
| Стэк для объектов класса DVS | sDVS | MyStack< DVS > |
| Стэк для объектов класса GasTurbine | sGT | MyStack< GasTurbine > |
| Стэк для объектов класса Rotor | sR | MyStack< Rotor > |
| Стэк для объектов класса ElectricityPlants | sElP | MyStack< ElectricityPlants > |
| Список указателей на интерфейс | LISTPOINTER | Vec<CInterface^> |

## Тестирование программы

Для проверки работоспособности программы проводилось тестирование с различными входными данными.

Тестирование 1. Ввод данных в программе.

Входные данные:

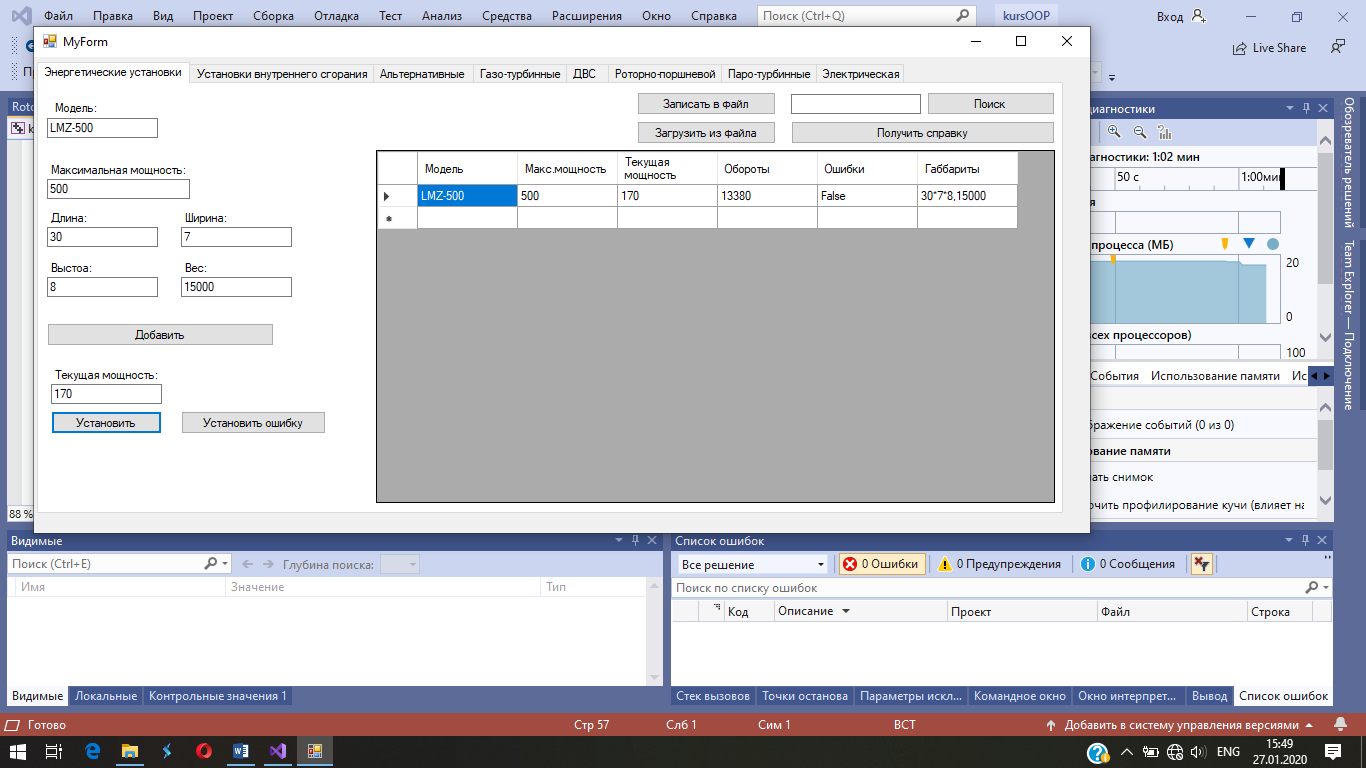


Рис.5 Входные данные № 1

Тестирование 2. Создание объекта без ввода данных о нем.

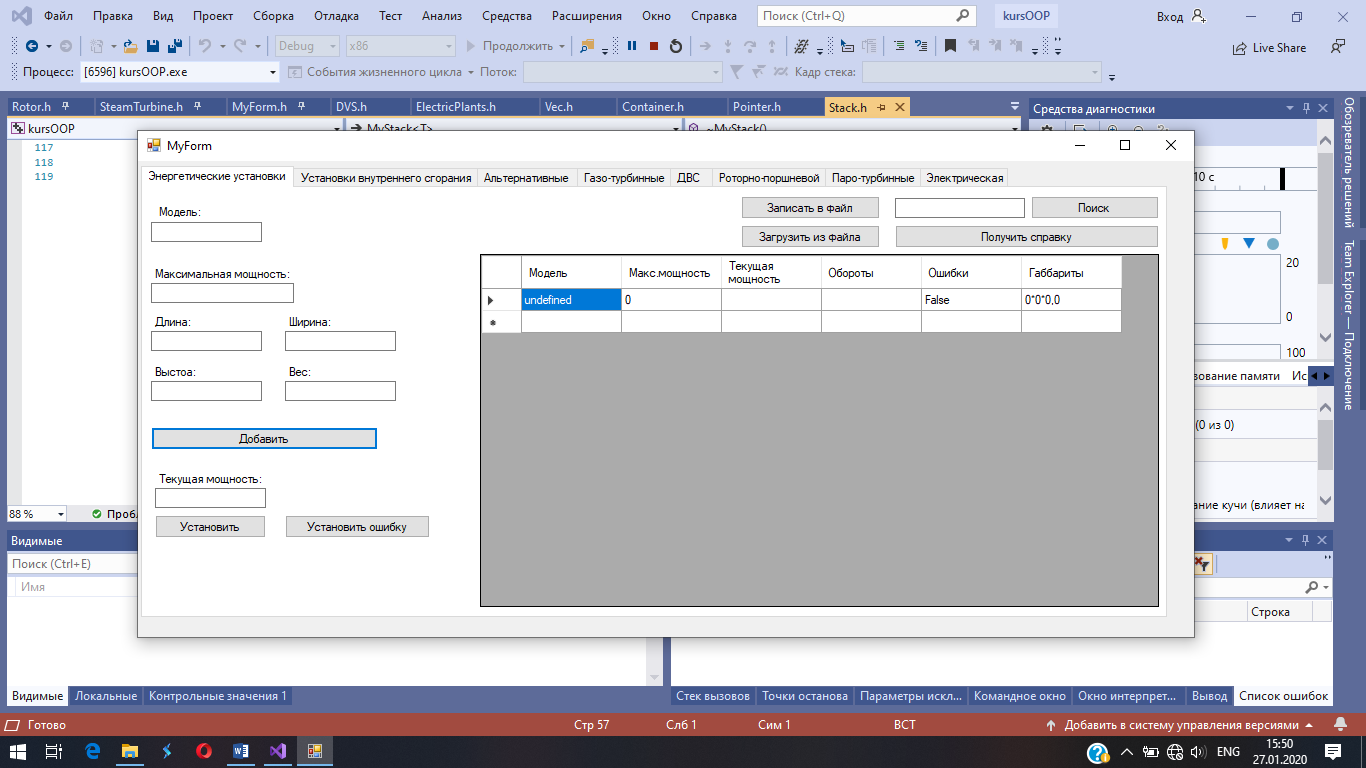


Рис.6 Входные данные № 2

Тестирование 3. Поиск объекта по полю модель.

Входные данные:

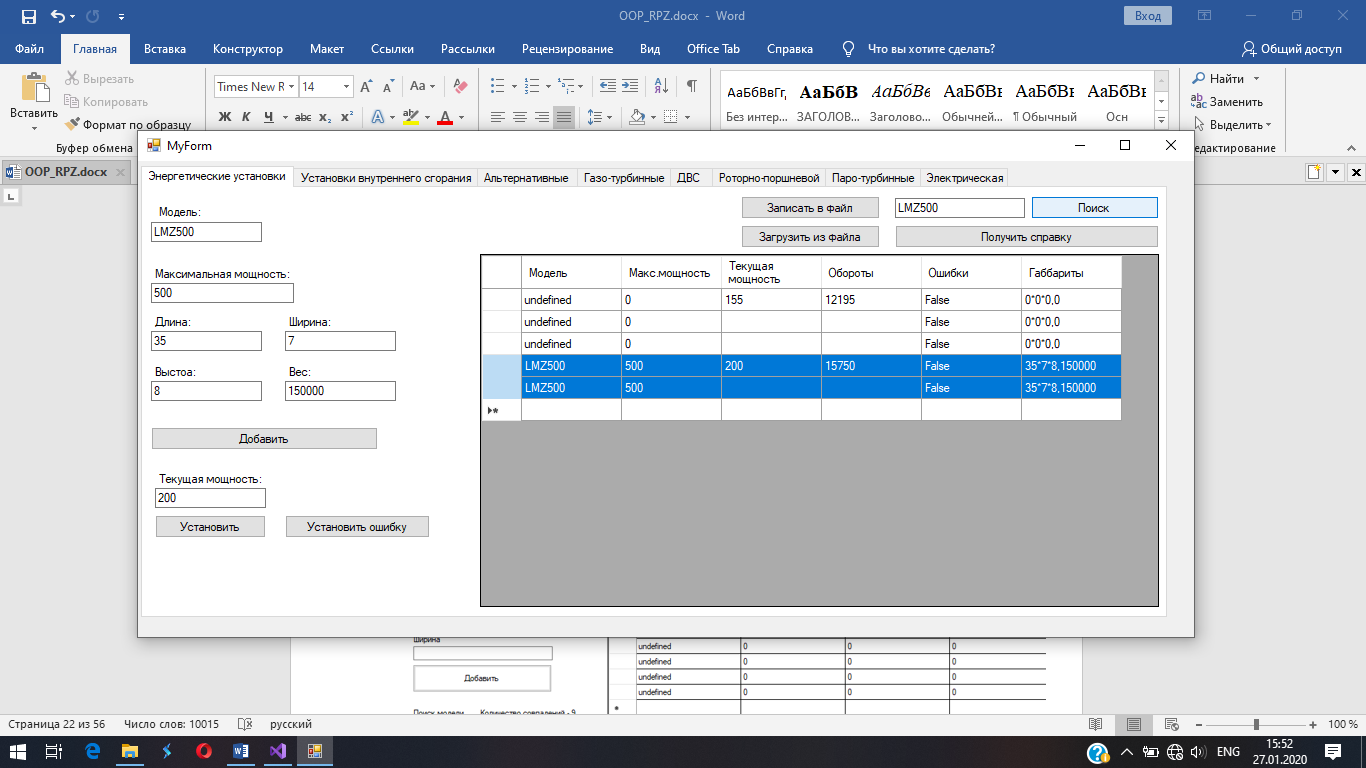


Рис.7 Входные данные № 3

Тестирование 4. Загрузка входных данных в файл.

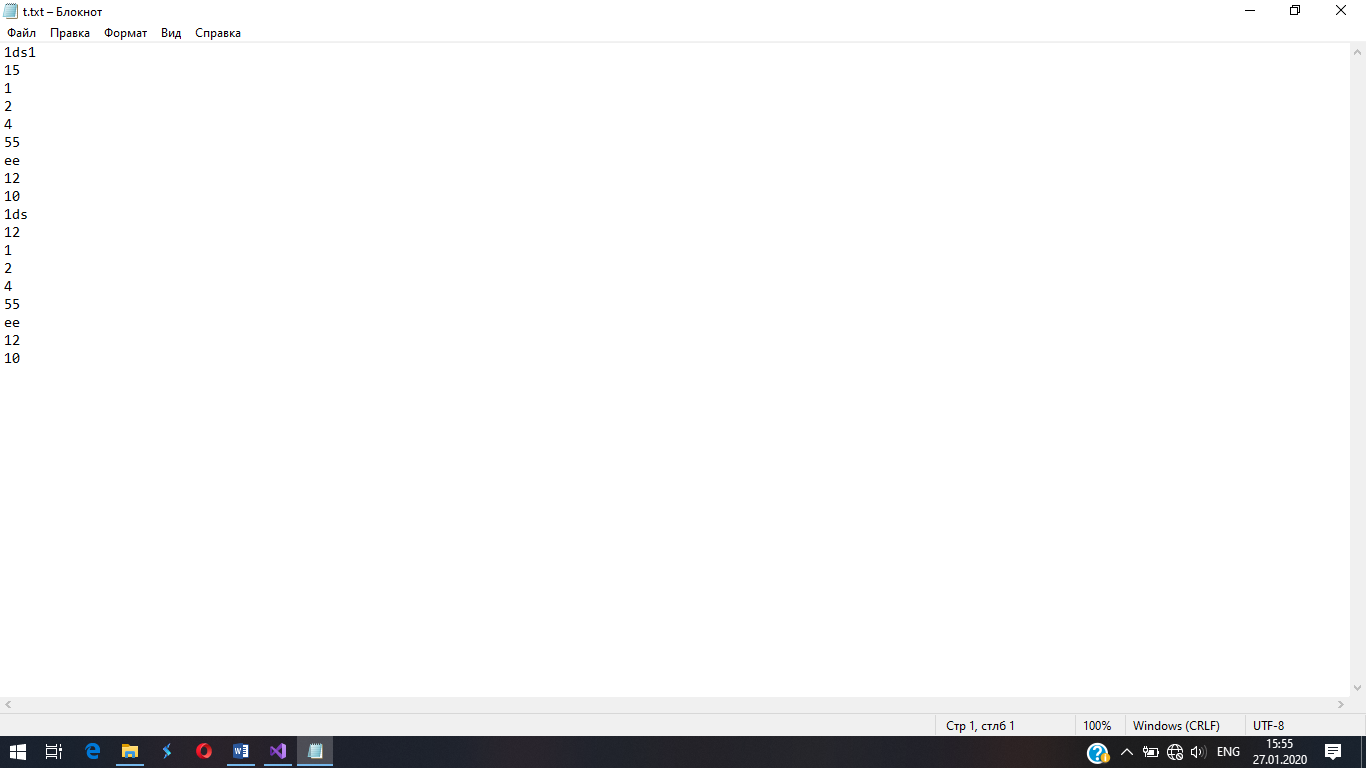


Рис.8 Результат 4 тестирования

Тестирование 5. Загрузка данных из файла.

Входные данные:

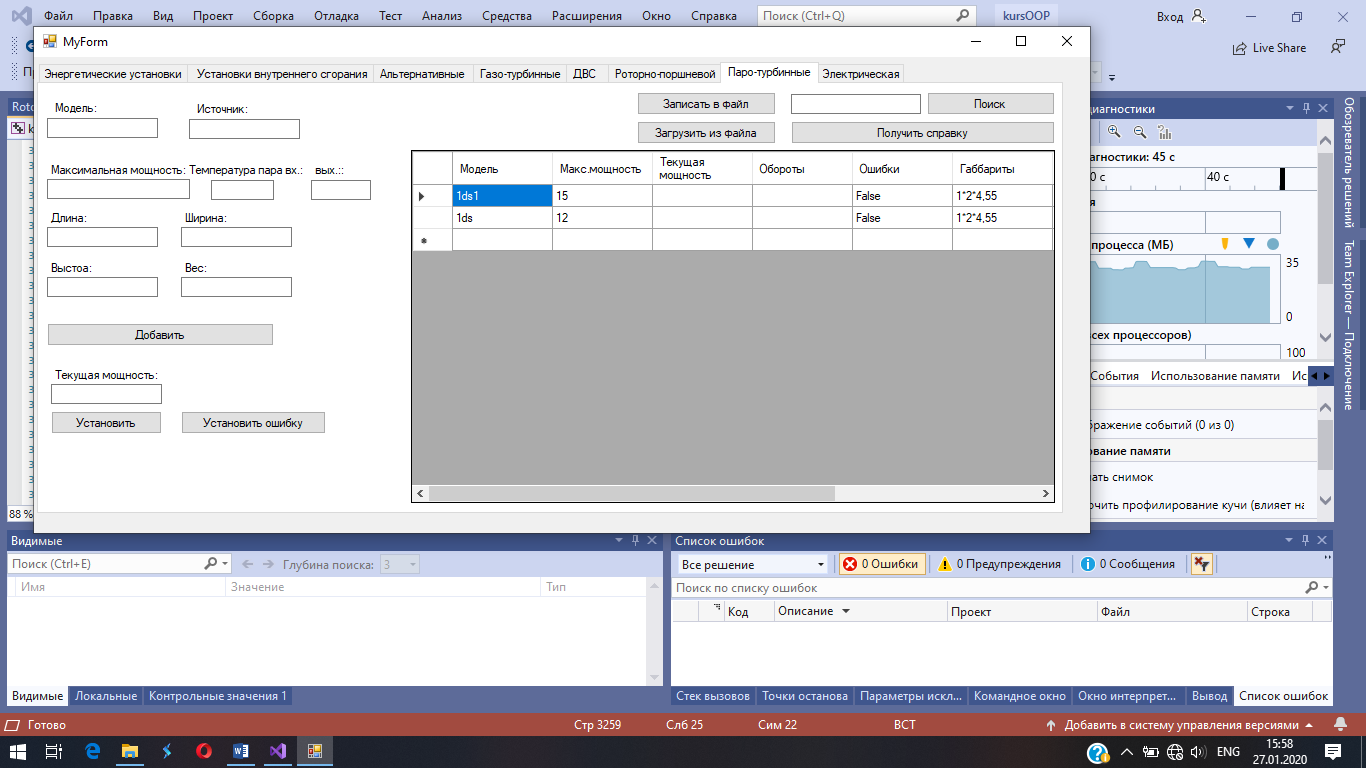


Рис.9 Результат 5 тестирования

Тестирование 6. Интерактивный ввод некорректных данных.

Входные данные:

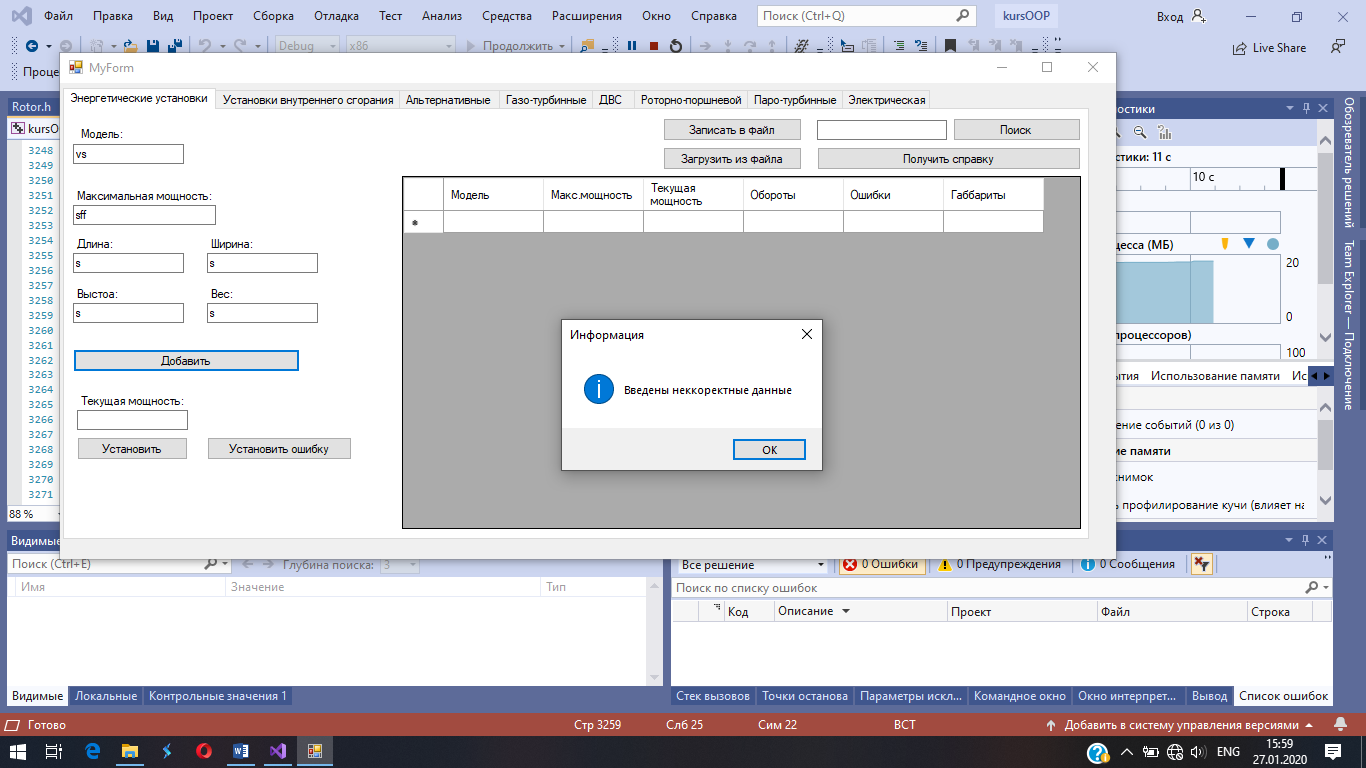


Рис.10 Входные данные № 5

Тестирование 7. Загрузка из файла некорректных данных.

Входные данные:

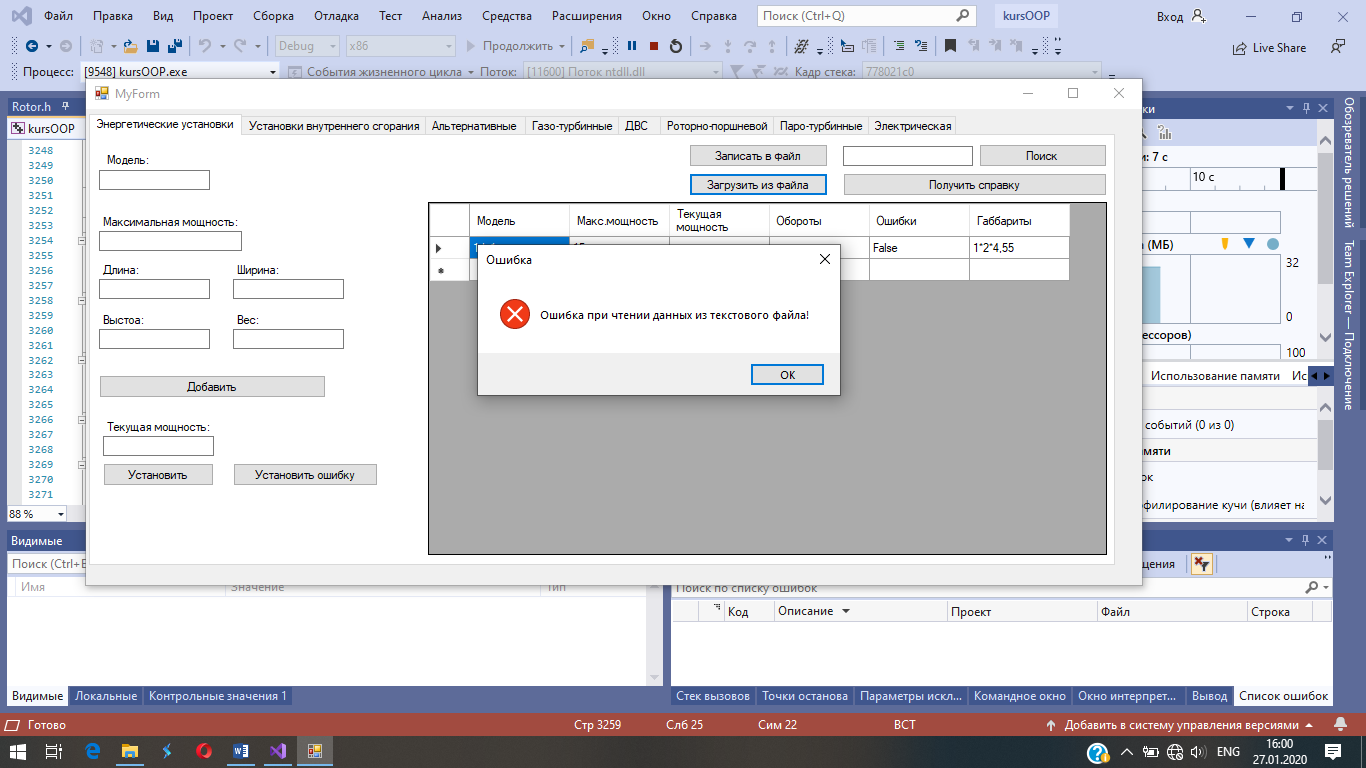


Рис.11 Входные данные № 6

Программа прошла тесты и соответствует всем требованиям, согласно техническому заданию.

Таблица 3. Протокол тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата и время тестирования | Тест | Кто проводил тестирование | Описание теста | Результаты тестирования |
| 29.12.2019 | Добавление объектов | Ответственное лицо | Объекты добавляются в контейнер, также выводится информация о них | Успех |
| 29.12.2019 | Поиск объектов | Ответственное лицо | Успешно выводиться количество найденных объектов | Успех |
| 29.12.2019 | Сохранение объектов в файл | Ответственное лицо | Успешно создается файл и поля объектов построчно хранятся в файле | Успех |
| 29.12.2019 | Загрузка объектов из файла | Ответственное лицо | Объекты успешно загружаются в контейнер и таблицу | Успех |
| 29.12.2019 | Работа с компонентными функциями | Ответственное лицо | Компонентные функции работают корректно | Успех |

# Заключение

В результате курсовой работы было разработано программное обеспечение для работы с объектами предметной области – «Энергетические установки». В рамках данной работы проводилась работа с механизмами одиночного наследования, указателями, шаблонами и полиморфными функциями. Так же был разработан визуальный интерфейс для работы с объектами предметной области, с функциями поиска, записи данных в файл, вывода из файла, добавлением объектов (с праметрами и без), установкой ошибки и текущей мощности для выбранного объекта, вывода информации о нем. Был создан контейнер для хранения объектов предметной области. Созданы механизмы работы с исключительными ситуациями. В качестве приобретенных компетенций были получены дополнительные навыки программирования на языке C++, закреплены полученные знания дисциплины объектно-ориентированное программирование.

# Список литературы

1. Лафоре, Р. Объектно-ориентированное программирование в С++ / Р. Лафоре. - СПб.: Питер, 2019. - 928 c.
2. Павловская, Т. С/С++.Процедурное и объектно-ориентированное программирование / Т. Павловская. - СПб.: Питер, 2018. - 496 c.
3. Ашарина, И.В. Объектно-ориентированное программирование в С++: лекции и упражнения: Учебное пособие для вузов / И.В. Ашарина. - М.: РиС, 2015. - 336 c.
4. Комлев, Николай Юрьевич Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей / Комлев Николай Юрьевич. - М.: Солон-Пресс, 2014. - ****77**0** c.
5. Арлоу, Джим ULISTPOINTER 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Джим Арлоу , Айла Нейштадт. - Москва: **Мир**, 2007. - 624 c.
6. Лаптев В.В. «С++. Объектно – ориентированное программирование: Учебное пособие. – СПб.: Питер, 2008. – 464 с.: ил. – (Серия «Учебное пособие»)»
7. Пол, Айра Объектно-ориентированное программирование на С++ / Айра Пол. - М.: Не указано, Бином, Невский Диалект, 2001. - 464 c.

Приложение 1

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

наименование института (факультета)

Математического и программного обеспечения ЭВМ

наименование кафедры

Объектно-ориентированное программирование

наименование дисциплины в соответствии с учебным планом

УТВЕРЖДАЮ

Зав. кафедрой ,

д.т.н., профессор Ершов Е.В.

« » 2019 г.

Объектно-ориентированное программирование на языке C++

Техническое задание на курсовую работу

Листов 4

Руководитель: заведующий кафедрой Ершов Евгений Валентинович

Исполнитель: студент гр. 1ПИб-01-31оп

Краев А.А.

2019 год

Введение

Задачей данной курсовой работы является создание ПО для работы с объектами предметной области – «Энергетические установки». Приложение должно помогать пользователю работать с объектами иерархии родственных типов.

1. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки является задание на курсовую работу по дисциплине «Объектно-ориентированное программирование», выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Дата утверждения: 20 сентября 2019 года.

Наименование темы разработки: Объектно-ориентированное программирование на языке C++, выданное преподавателем кафедры МПО ЭВМ ИИТ ЧГУ в рамках учебного плана. Общие требования к программному обеспечению для разработки следующие:

* Разработать иерархии родственных типов, корневой класс которых класс-интерфейс, для моделирования и обработки данных предметных областей набором отложенных методов - полиморфная обработка родственных объектов (согласно вариантов А.х.х и А.у.у). Организовать взаимодействие иерархий типов через указатели на интерфейсы и виртуальные функции (отложенные методы).
* Создать обобщенный (void\*) контейнерный класс (базовый) и от него, используя закрытое наследование, производный класс – шаблон для хранения указателей на абстрактный базовый класс - интерфейс (согласно варианта В.х).
* Для хранения объектов каждого производного класса использовать структуру данных (согласно варианта С.х).
* Реализовать функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
* Реализовать файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
* Предусмотреть обработку различных исключительных ситуаций.
* Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

Вариант:

А 31. Энергетические установки

В 1. Статический вектор

С 7. Стек

2. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основным назначением созданного продукта является работа с объектами родственных классов в визуальном интерфейсе. Приложение должно предоставлять пользователю возможность производить ввод данных и отображать эти данные в таблице.

3. ТРЕБОВАНИЯ К РАЗРАБОТКЕ

* 1. Требования к функциональным характеристикам

К программному продукту предъявлены следующие требования:

* создание объектов классов и обмен данными между ними;
* запись данных в файл;
* ввод данных с клавиатуры;
* вывод данных из файла;
* вывод данных и результатов на экран;
* создание классов и демонстрация механизмов работы с ними;
* создание иерархии родственных типов;
* создание обобщенного контейнерного класса - списка, и от него производный класс – шаблон;
* использование очереди для хранения объектов производного класса;
* реализация механизмов наследования;
* реализация функции обработки данных;
* полиморфная обработка родственных объектов;
* обработка исключительных ситуаций.

3.2 Требования к надежности

При создании приложения необходимо предусмотреть:

1. Обработку некорректных действий со стороны пользователя;
2. Работоспособность элементов программы;
3. Адекватное отображение интерфейса и форм.

3.3 Условия эксплуатации

ПО предназначено для работы на одном персональном компьютере одним пользователем, не требует наличия обслуживающего персонала.

Не требуется высокая квалификация пользователя.

3.4 Требования к составу и параметрам технических средств

Требования к параметрам технических средств определяются системными требованиями программного обеспечения Visual Studio 2019.

Минимальные системные требования:

1. процессор - 32- или 64-разрядный процессор с тактовой частотой 1.6 ГГц;
2. операционная система - Windows xp или выше;
3. оперативная память - 1 ГБ ОЗУ;
4. свободное место на жестком диске - 4 ГБ свободного места на диске;
5. монитор разрешение - 1366x 768
6. графика - встроенный графический адаптер;

3.5 Требования к информационной и программной совместимости

Операционная система семейства Windows x32, x64.

3.6 Требования к маркировке и упаковке

Требования не предъявляются.

3.7 Требования к транспортированию и хранению

Данная программа должна храниться на компакт-диске, который позволит пользователю получить доступ к приложению.

4. Специальные требования

Требования не предъявляются.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1 Программная документация должна содержать расчётно-пояснительную записку с содержанием:

1. Титульный лист.
2. Оглавление.
3. Введение.
4. Основная часть
5. Заключение
6. Список литературы.
7. Приложения (содержит ТЗ, руководство пользователя, текст программы; из основного текста должны быть ссылки на номера разделов Приложения).

6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии и этапы разработки представлены в таблице П1.1.

Таблица П1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапа разработки моделей | Сроки разработки | Результат выполнения | Отметка о выполнении |
| Разработка технического задания | 15.11.2019 – 20.11.2019 | Оформленное ТЗ |  |
| Анализ предметной области, разработка иерархии предметной облсти | 24.11.2019 – 27.11.2019 | Понятия предметной области |  |
| Программирование приложения | 15.12.2019 – 30.12.2019 | Приложение для работы с видеомониторами |  |
| Оформление РПЗ | 30.12.2019– 10.01.2020 | Оформленная РПЗ |  |

1. ПОРЯДОК КОНТРОЛЯ

Порядок контроля и приемки представлен в таблице П1.2

Таблица П1.2

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование контрольного этапа выполнения курсовой работы | Сроки контроля | Результат выполнения | Отметка о приемке результата контрольного этапа |
| Техническое задание | 20.11.2019 | Утвержденное ТЗ |  |
| Первая версия приложения | 10.12.2019 | Версия ПО с указаниями недостатков |  |
| Финальная версия приложения | 19.01.2020 | Рабочая версия приложения |  |
| Оформленное РПЗ | 26.01.2020 | Защита |  |

# Приложение 2

1. Руководство пользователя
   1. Общие сведения

Данная программа предназначена для работы с энергетическими установками разных типов.

* 1. Описание установки

Данная программа не требует установки.

* 1. Описание запуска

Для открытия программы необходимо запустить файл kursOOP.exe, лежащий в папке kursOOP/Debug, находящийся на загрузочном диске. После запуска откроется главное окно программы (рис. П2.1):

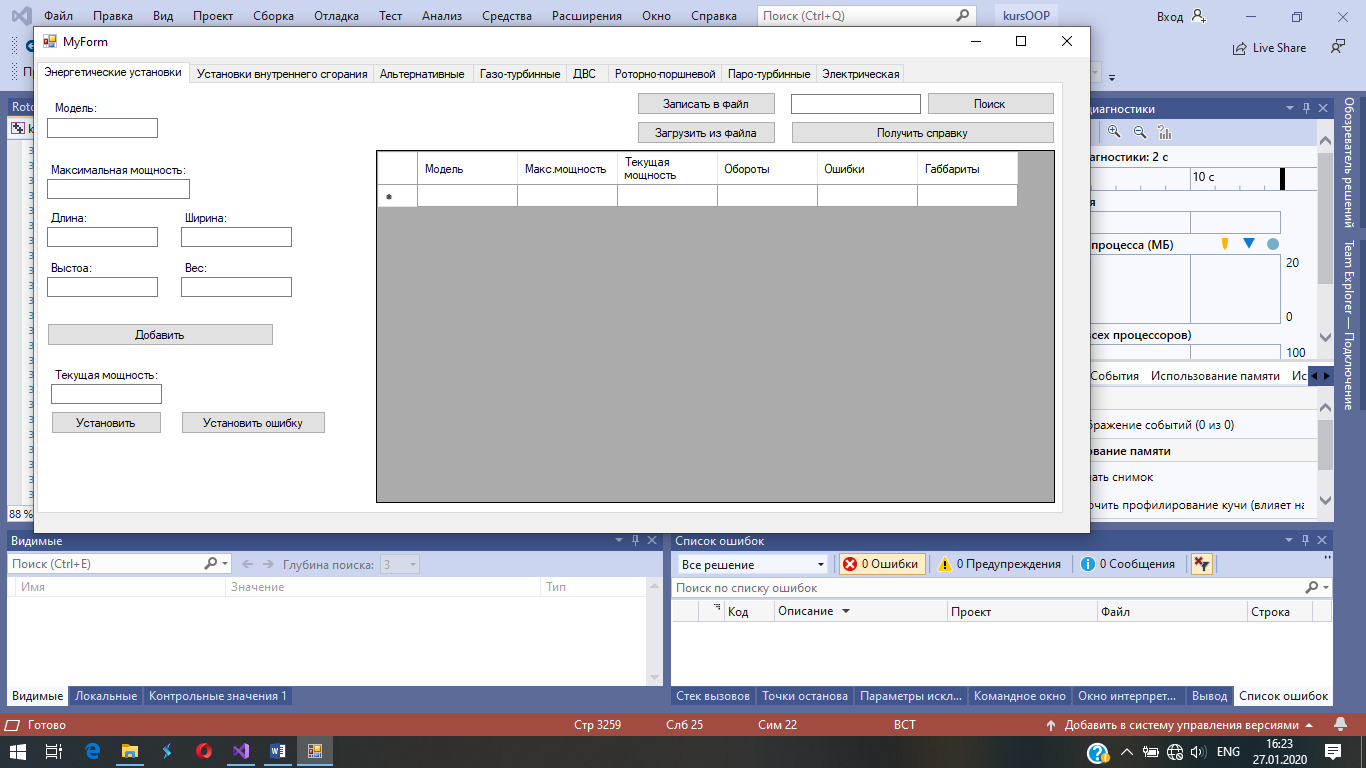


Рис. П2.1 Главное окно программы

Для начала работы необходимо выбрать вкладку с интересующим нас типом установки. Затем мы можем ввести параметры агрегата и нажать на кнопку «Добавить» или произвести загрузку из файла путем нажатия на соответствующий элемент управления. Помимо этого, существует возможность установки текущей мощности об объекте, установки ошибки и получения статуса о его работе, данные манипуляции осуществляются путем выбора элемента в таблице и нажатием соответсвующей кнопкт.

# Приложение 3

Текст программы

#pragma once

#include "CInterface.h"

#include "Stack.h"

#include"EnergyPlants.h"

#include "InternalCombustion.h"

#include "AlternativePlants.h"

#include "GasTurbine.h"

#include "DVS.h"

#include "ElectricPlants.h"

#include "Rotor.h"

#include "SteamTurbine.h"

#include "Vec.h"

namespace kursOOP {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyStack<EnergyPlants> sEP;

MyStack<InternalCombustion> sIC;

MyStack<AlternativePlants> sAP;

MyStack<DVS> sDVS;

MyStack<GasTurbine> sGT;

MyStack<SteamTurbine> sST;

MyStack<Rotor> sR;

Vec<CInterface^> PM;

private: System::Windows::Forms::TextBox^ textBox65;

public:

private: System::Windows::Forms::Label^ label57;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn3;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn4;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn5;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn6;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn43;

private: System::Windows::Forms::TextBox^ textBox66;

private: System::Windows::Forms::Label^ label58;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn7;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn8;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn9;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn10;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn11;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn12;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column7;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn13;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn14;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn15;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn16;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn17;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn18;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column8;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column9;

private: System::Windows::Forms::TextBox^ textBox68;

private: System::Windows::Forms::Label^ label60;

private: System::Windows::Forms::TextBox^ textBox67;

private: System::Windows::Forms::Label^ label59;

private: System::Windows::Forms::TextBox^ textBox69;

private: System::Windows::Forms::Label^ label61;

private: System::Windows::Forms::TextBox^ textBox70;

private: System::Windows::Forms::Label^ label62;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn19;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn20;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn21;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn22;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn23;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn24;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column12;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column13;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column3;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column4;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column5;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column6;

private: System::Windows::Forms::TextBox^ textBox71;

private: System::Windows::Forms::Label^ label63;

private: System::Windows::Forms::TextBox^ textBox72;

private: System::Windows::Forms::Label^ label64;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn25;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn26;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn27;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn28;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn29;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn30;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column10;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column11;

private: System::Windows::Forms::TextBox^ textBox75;

private: System::Windows::Forms::Label^ label67;

private: System::Windows::Forms::TextBox^ textBox74;

private: System::Windows::Forms::Label^ label66;

private: System::Windows::Forms::TextBox^ textBox73;

private: System::Windows::Forms::Label^ label65;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn31;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn32;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn33;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn34;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn35;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn36;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column14;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column15;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column16;

private: System::Windows::Forms::TextBox^ textBox77;

private: System::Windows::Forms::Label^ label69;

private: System::Windows::Forms::TextBox^ textBox76;

private: System::Windows::Forms::Label^ label68;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn37;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn38;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn39;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn40;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn41;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn42;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column17;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column18;

MyStack<ElectricPlants> sElP;

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

for (int i = 0; i < 8; i++)

{

PM.push\_back(nullptr);

}

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TabControl^ tabControl1;

protected:

private: System::Windows::Forms::TabPage^ tabPage1;

private: System::Windows::Forms::TabPage^ tabPage2;

private: System::Windows::Forms::TabPage^ tabPage3;

private: System::Windows::Forms::TabPage^ tabPage4;

private: System::Windows::Forms::TabPage^ tabPage5;

private: System::Windows::Forms::TabPage^ tabPage6;

private: System::Windows::Forms::TabPage^ tabPage7;

private: System::Windows::Forms::TabPage^ tabPage8;

private: System::Windows::Forms::DataGridView^ dataGridView1;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::TextBox^ textBox7;

private: System::Windows::Forms::Label^ label7;

private: System::Windows::Forms::TextBox^ textBox6;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::TextBox^ textBox5;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::TextBox^ textBox4;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::TextBox^ textBox8;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button11;

private: System::Windows::Forms::Button^ button12;

private: System::Windows::Forms::TextBox^ textBox9;

private: System::Windows::Forms::Button^ button13;

private: System::Windows::Forms::Button^ button14;

private: System::Windows::Forms::TextBox^ textBox10;

private: System::Windows::Forms::Label^ label8;

private: System::Windows::Forms::TextBox^ textBox11;

private: System::Windows::Forms::Label^ label9;

private: System::Windows::Forms::TextBox^ textBox12;

private: System::Windows::Forms::Label^ label10;

private: System::Windows::Forms::TextBox^ textBox13;

private: System::Windows::Forms::Label^ label11;

private: System::Windows::Forms::TextBox^ textBox14;

private: System::Windows::Forms::Label^ label12;

private: System::Windows::Forms::TextBox^ textBox15;

private: System::Windows::Forms::Label^ label13;

private: System::Windows::Forms::TextBox^ textBox16;

private: System::Windows::Forms::Label^ label14;

private: System::Windows::Forms::DataGridView^ dataGridView2;

private: System::Windows::Forms::Button^ button15;

private: System::Windows::Forms::Button^ button16;

private: System::Windows::Forms::Button^ button17;

private: System::Windows::Forms::Button^ button18;

private: System::Windows::Forms::Button^ button19;

private: System::Windows::Forms::TextBox^ textBox17;

private: System::Windows::Forms::Button^ button20;

private: System::Windows::Forms::Button^ button21;

private: System::Windows::Forms::TextBox^ textBox18;

private: System::Windows::Forms::Label^ label15;

private: System::Windows::Forms::TextBox^ textBox19;

private: System::Windows::Forms::Label^ label16;

private: System::Windows::Forms::TextBox^ textBox20;

private: System::Windows::Forms::Label^ label17;

private: System::Windows::Forms::TextBox^ textBox21;

private: System::Windows::Forms::Label^ label18;

private: System::Windows::Forms::TextBox^ textBox22;

private: System::Windows::Forms::Label^ label19;

private: System::Windows::Forms::TextBox^ textBox23;

private: System::Windows::Forms::Label^ label20;

private: System::Windows::Forms::TextBox^ textBox24;

private: System::Windows::Forms::Label^ label21;

private: System::Windows::Forms::DataGridView^ dataGridView3;

private: System::Windows::Forms::Button^ button22;

private: System::Windows::Forms::Button^ button23;

private: System::Windows::Forms::Button^ button24;

private: System::Windows::Forms::Button^ button25;

private: System::Windows::Forms::Button^ button26;

private: System::Windows::Forms::TextBox^ textBox25;

private: System::Windows::Forms::Button^ button27;

private: System::Windows::Forms::Button^ button28;

private: System::Windows::Forms::TextBox^ textBox26;

private: System::Windows::Forms::Label^ label22;

private: System::Windows::Forms::TextBox^ textBox27;

private: System::Windows::Forms::Label^ label23;

private: System::Windows::Forms::TextBox^ textBox28;

private: System::Windows::Forms::Label^ label24;

private: System::Windows::Forms::TextBox^ textBox29;

private: System::Windows::Forms::Label^ label25;

private: System::Windows::Forms::TextBox^ textBox30;

private: System::Windows::Forms::Label^ label26;

private: System::Windows::Forms::TextBox^ textBox31;

private: System::Windows::Forms::Label^ label27;

private: System::Windows::Forms::TextBox^ textBox32;

private: System::Windows::Forms::Label^ label28;

private: System::Windows::Forms::DataGridView^ dataGridView4;

private: System::Windows::Forms::Button^ button29;

private: System::Windows::Forms::Button^ button30;

private: System::Windows::Forms::Button^ button31;

private: System::Windows::Forms::Button^ button32;

private: System::Windows::Forms::Button^ button33;

private: System::Windows::Forms::TextBox^ textBox33;

private: System::Windows::Forms::Button^ button34;

private: System::Windows::Forms::Button^ button35;

private: System::Windows::Forms::TextBox^ textBox34;

private: System::Windows::Forms::Label^ label29;

private: System::Windows::Forms::TextBox^ textBox35;

private: System::Windows::Forms::Label^ label30;

private: System::Windows::Forms::TextBox^ textBox36;

private: System::Windows::Forms::Label^ label31;

private: System::Windows::Forms::TextBox^ textBox37;

private: System::Windows::Forms::Label^ label32;

private: System::Windows::Forms::TextBox^ textBox38;

private: System::Windows::Forms::Label^ label33;

private: System::Windows::Forms::TextBox^ textBox39;

private: System::Windows::Forms::Label^ label34;

private: System::Windows::Forms::TextBox^ textBox40;

private: System::Windows::Forms::Label^ label35;

private: System::Windows::Forms::DataGridView^ dataGridView5;

private: System::Windows::Forms::Button^ button36;

private: System::Windows::Forms::Button^ button37;

private: System::Windows::Forms::Button^ button38;

private: System::Windows::Forms::Button^ button39;

private: System::Windows::Forms::Button^ button40;

private: System::Windows::Forms::TextBox^ textBox41;

private: System::Windows::Forms::Button^ button41;

private: System::Windows::Forms::Button^ button42;

private: System::Windows::Forms::TextBox^ textBox42;

private: System::Windows::Forms::Label^ label36;

private: System::Windows::Forms::TextBox^ textBox43;

private: System::Windows::Forms::Label^ label37;

private: System::Windows::Forms::TextBox^ textBox44;

private: System::Windows::Forms::Label^ label38;

private: System::Windows::Forms::TextBox^ textBox45;

private: System::Windows::Forms::Label^ label39;

private: System::Windows::Forms::TextBox^ textBox46;

private: System::Windows::Forms::Label^ label40;

private: System::Windows::Forms::TextBox^ textBox47;

private: System::Windows::Forms::Label^ label41;

private: System::Windows::Forms::TextBox^ textBox48;

private: System::Windows::Forms::Label^ label42;

private: System::Windows::Forms::DataGridView^ dataGridView6;

private: System::Windows::Forms::Button^ button43;

private: System::Windows::Forms::Button^ button44;

private: System::Windows::Forms::Button^ button45;

private: System::Windows::Forms::Button^ button46;

private: System::Windows::Forms::Button^ button47;

private: System::Windows::Forms::TextBox^ textBox49;

private: System::Windows::Forms::Button^ button48;

private: System::Windows::Forms::Button^ button49;

private: System::Windows::Forms::TextBox^ textBox50;

private: System::Windows::Forms::Label^ label43;

private: System::Windows::Forms::TextBox^ textBox51;

private: System::Windows::Forms::Label^ label44;

private: System::Windows::Forms::TextBox^ textBox52;

private: System::Windows::Forms::Label^ label45;

private: System::Windows::Forms::TextBox^ textBox53;

private: System::Windows::Forms::Label^ label46;

private: System::Windows::Forms::TextBox^ textBox54;

private: System::Windows::Forms::Label^ label47;

private: System::Windows::Forms::TextBox^ textBox55;

private: System::Windows::Forms::Label^ label48;

private: System::Windows::Forms::TextBox^ textBox56;

private: System::Windows::Forms::Label^ label49;

private: System::Windows::Forms::DataGridView^ dataGridView7;

private: System::Windows::Forms::Button^ button50;

private: System::Windows::Forms::Button^ button51;

private: System::Windows::Forms::Button^ button52;

private: System::Windows::Forms::Button^ button53;

private: System::Windows::Forms::Button^ button54;

private: System::Windows::Forms::TextBox^ textBox57;

private: System::Windows::Forms::Button^ button55;

private: System::Windows::Forms::Button^ button56;

private: System::Windows::Forms::TextBox^ textBox58;

private: System::Windows::Forms::Label^ label50;

private: System::Windows::Forms::TextBox^ textBox59;

private: System::Windows::Forms::Label^ label51;

private: System::Windows::Forms::TextBox^ textBox60;

private: System::Windows::Forms::Label^ label52;

private: System::Windows::Forms::TextBox^ textBox61;

private: System::Windows::Forms::Label^ label53;

private: System::Windows::Forms::TextBox^ textBox62;

private: System::Windows::Forms::Label^ label54;

private: System::Windows::Forms::TextBox^ textBox63;

private: System::Windows::Forms::Label^ label55;

private: System::Windows::Forms::TextBox^ textBox64;

private: System::Windows::Forms::Label^ label56;

private: System::Windows::Forms::DataGridView^ dataGridView8;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->tabControl1 = (gcnew System::Windows::Forms::TabControl());

this->tabPage1 = (gcnew System::Windows::Forms::TabPage());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->textBox8 = (gcnew System::Windows::Forms::TextBox());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox7 = (gcnew System::Windows::Forms::TextBox());

this->label7 = (gcnew System::Windows::Forms::Label());

this->textBox6 = (gcnew System::Windows::Forms::TextBox());

this->label6 = (gcnew System::Windows::Forms::Label());

this->textBox5 = (gcnew System::Windows::Forms::TextBox());

this->label5 = (gcnew System::Windows::Forms::Label());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->label4 = (gcnew System::Windows::Forms::Label());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->label3 = (gcnew System::Windows::Forms::Label());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->Column1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column4 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column5 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column6 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage2 = (gcnew System::Windows::Forms::TabPage());

this->textBox65 = (gcnew System::Windows::Forms::TextBox());

this->label57 = (gcnew System::Windows::Forms::Label());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button11 = (gcnew System::Windows::Forms::Button());

this->button12 = (gcnew System::Windows::Forms::Button());

this->textBox9 = (gcnew System::Windows::Forms::TextBox());

this->button13 = (gcnew System::Windows::Forms::Button());

this->button14 = (gcnew System::Windows::Forms::Button());

this->textBox10 = (gcnew System::Windows::Forms::TextBox());

this->label8 = (gcnew System::Windows::Forms::Label());

this->textBox11 = (gcnew System::Windows::Forms::TextBox());

this->label9 = (gcnew System::Windows::Forms::Label());

this->textBox12 = (gcnew System::Windows::Forms::TextBox());

this->label10 = (gcnew System::Windows::Forms::Label());

this->textBox13 = (gcnew System::Windows::Forms::TextBox());

this->label11 = (gcnew System::Windows::Forms::Label());

this->textBox14 = (gcnew System::Windows::Forms::TextBox());

this->label12 = (gcnew System::Windows::Forms::Label());

this->textBox15 = (gcnew System::Windows::Forms::TextBox());

this->label13 = (gcnew System::Windows::Forms::Label());

this->textBox16 = (gcnew System::Windows::Forms::TextBox());

this->label14 = (gcnew System::Windows::Forms::Label());

this->dataGridView2 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn4 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn5 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn6 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn43 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage3 = (gcnew System::Windows::Forms::TabPage());

this->textBox66 = (gcnew System::Windows::Forms::TextBox());

this->label58 = (gcnew System::Windows::Forms::Label());

this->button15 = (gcnew System::Windows::Forms::Button());

this->button16 = (gcnew System::Windows::Forms::Button());

this->button17 = (gcnew System::Windows::Forms::Button());

this->button18 = (gcnew System::Windows::Forms::Button());

this->button19 = (gcnew System::Windows::Forms::Button());

this->textBox17 = (gcnew System::Windows::Forms::TextBox());

this->button20 = (gcnew System::Windows::Forms::Button());

this->button21 = (gcnew System::Windows::Forms::Button());

this->textBox18 = (gcnew System::Windows::Forms::TextBox());

this->label15 = (gcnew System::Windows::Forms::Label());

this->textBox19 = (gcnew System::Windows::Forms::TextBox());

this->label16 = (gcnew System::Windows::Forms::Label());

this->textBox20 = (gcnew System::Windows::Forms::TextBox());

this->label17 = (gcnew System::Windows::Forms::Label());

this->textBox21 = (gcnew System::Windows::Forms::TextBox());

this->label18 = (gcnew System::Windows::Forms::Label());

this->textBox22 = (gcnew System::Windows::Forms::TextBox());

this->label19 = (gcnew System::Windows::Forms::Label());

this->textBox23 = (gcnew System::Windows::Forms::TextBox());

this->label20 = (gcnew System::Windows::Forms::Label());

this->textBox24 = (gcnew System::Windows::Forms::TextBox());

this->label21 = (gcnew System::Windows::Forms::Label());

this->dataGridView3 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn7 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn8 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn9 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn10 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn11 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn12 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column7 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage4 = (gcnew System::Windows::Forms::TabPage());

this->textBox68 = (gcnew System::Windows::Forms::TextBox());

this->label60 = (gcnew System::Windows::Forms::Label());

this->textBox67 = (gcnew System::Windows::Forms::TextBox());

this->label59 = (gcnew System::Windows::Forms::Label());

this->button22 = (gcnew System::Windows::Forms::Button());

this->button23 = (gcnew System::Windows::Forms::Button());

this->button24 = (gcnew System::Windows::Forms::Button());

this->button25 = (gcnew System::Windows::Forms::Button());

this->button26 = (gcnew System::Windows::Forms::Button());

this->textBox25 = (gcnew System::Windows::Forms::TextBox());

this->button27 = (gcnew System::Windows::Forms::Button());

this->button28 = (gcnew System::Windows::Forms::Button());

this->textBox26 = (gcnew System::Windows::Forms::TextBox());

this->label22 = (gcnew System::Windows::Forms::Label());

this->textBox27 = (gcnew System::Windows::Forms::TextBox());

this->label23 = (gcnew System::Windows::Forms::Label());

this->textBox28 = (gcnew System::Windows::Forms::TextBox());

this->label24 = (gcnew System::Windows::Forms::Label());

this->textBox29 = (gcnew System::Windows::Forms::TextBox());

this->label25 = (gcnew System::Windows::Forms::Label());

this->textBox30 = (gcnew System::Windows::Forms::TextBox());

this->label26 = (gcnew System::Windows::Forms::Label());

this->textBox31 = (gcnew System::Windows::Forms::TextBox());

this->label27 = (gcnew System::Windows::Forms::Label());

this->textBox32 = (gcnew System::Windows::Forms::TextBox());

this->label28 = (gcnew System::Windows::Forms::Label());

this->dataGridView4 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn13 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn14 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn15 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn16 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn17 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn18 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column8 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column9 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage5 = (gcnew System::Windows::Forms::TabPage());

this->textBox69 = (gcnew System::Windows::Forms::TextBox());

this->label61 = (gcnew System::Windows::Forms::Label());

this->textBox70 = (gcnew System::Windows::Forms::TextBox());

this->label62 = (gcnew System::Windows::Forms::Label());

this->button29 = (gcnew System::Windows::Forms::Button());

this->button30 = (gcnew System::Windows::Forms::Button());

this->button31 = (gcnew System::Windows::Forms::Button());

this->button32 = (gcnew System::Windows::Forms::Button());

this->button33 = (gcnew System::Windows::Forms::Button());

this->textBox33 = (gcnew System::Windows::Forms::TextBox());

this->button34 = (gcnew System::Windows::Forms::Button());

this->button35 = (gcnew System::Windows::Forms::Button());

this->textBox34 = (gcnew System::Windows::Forms::TextBox());

this->label29 = (gcnew System::Windows::Forms::Label());

this->textBox35 = (gcnew System::Windows::Forms::TextBox());

this->label30 = (gcnew System::Windows::Forms::Label());

this->textBox36 = (gcnew System::Windows::Forms::TextBox());

this->label31 = (gcnew System::Windows::Forms::Label());

this->textBox37 = (gcnew System::Windows::Forms::TextBox());

this->label32 = (gcnew System::Windows::Forms::Label());

this->textBox38 = (gcnew System::Windows::Forms::TextBox());

this->label33 = (gcnew System::Windows::Forms::Label());

this->textBox39 = (gcnew System::Windows::Forms::TextBox());

this->label34 = (gcnew System::Windows::Forms::Label());

this->textBox40 = (gcnew System::Windows::Forms::TextBox());

this->label35 = (gcnew System::Windows::Forms::Label());

this->dataGridView5 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn19 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn20 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn21 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn22 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn23 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn24 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column12 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column13 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage6 = (gcnew System::Windows::Forms::TabPage());

this->textBox71 = (gcnew System::Windows::Forms::TextBox());

this->label63 = (gcnew System::Windows::Forms::Label());

this->textBox72 = (gcnew System::Windows::Forms::TextBox());

this->label64 = (gcnew System::Windows::Forms::Label());

this->button36 = (gcnew System::Windows::Forms::Button());

this->button37 = (gcnew System::Windows::Forms::Button());

this->button38 = (gcnew System::Windows::Forms::Button());

this->button39 = (gcnew System::Windows::Forms::Button());

this->button40 = (gcnew System::Windows::Forms::Button());

this->textBox41 = (gcnew System::Windows::Forms::TextBox());

this->button41 = (gcnew System::Windows::Forms::Button());

this->button42 = (gcnew System::Windows::Forms::Button());

this->textBox42 = (gcnew System::Windows::Forms::TextBox());

this->label36 = (gcnew System::Windows::Forms::Label());

this->textBox43 = (gcnew System::Windows::Forms::TextBox());

this->label37 = (gcnew System::Windows::Forms::Label());

this->textBox44 = (gcnew System::Windows::Forms::TextBox());

this->label38 = (gcnew System::Windows::Forms::Label());

this->textBox45 = (gcnew System::Windows::Forms::TextBox());

this->label39 = (gcnew System::Windows::Forms::Label());

this->textBox46 = (gcnew System::Windows::Forms::TextBox());

this->label40 = (gcnew System::Windows::Forms::Label());

this->textBox47 = (gcnew System::Windows::Forms::TextBox());

this->label41 = (gcnew System::Windows::Forms::Label());

this->textBox48 = (gcnew System::Windows::Forms::TextBox());

this->label42 = (gcnew System::Windows::Forms::Label());

this->dataGridView6 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn25 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn26 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn27 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn28 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn29 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn30 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column10 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column11 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage7 = (gcnew System::Windows::Forms::TabPage());

this->textBox75 = (gcnew System::Windows::Forms::TextBox());

this->label67 = (gcnew System::Windows::Forms::Label());

this->textBox74 = (gcnew System::Windows::Forms::TextBox());

this->label66 = (gcnew System::Windows::Forms::Label());

this->textBox73 = (gcnew System::Windows::Forms::TextBox());

this->label65 = (gcnew System::Windows::Forms::Label());

this->button43 = (gcnew System::Windows::Forms::Button());

this->button44 = (gcnew System::Windows::Forms::Button());

this->button45 = (gcnew System::Windows::Forms::Button());

this->button46 = (gcnew System::Windows::Forms::Button());

this->button47 = (gcnew System::Windows::Forms::Button());

this->textBox49 = (gcnew System::Windows::Forms::TextBox());

this->button48 = (gcnew System::Windows::Forms::Button());

this->button49 = (gcnew System::Windows::Forms::Button());

this->textBox50 = (gcnew System::Windows::Forms::TextBox());

this->label43 = (gcnew System::Windows::Forms::Label());

this->textBox51 = (gcnew System::Windows::Forms::TextBox());

this->label44 = (gcnew System::Windows::Forms::Label());

this->textBox52 = (gcnew System::Windows::Forms::TextBox());

this->label45 = (gcnew System::Windows::Forms::Label());

this->textBox53 = (gcnew System::Windows::Forms::TextBox());

this->label46 = (gcnew System::Windows::Forms::Label());

this->textBox54 = (gcnew System::Windows::Forms::TextBox());

this->label47 = (gcnew System::Windows::Forms::Label());

this->textBox55 = (gcnew System::Windows::Forms::TextBox());

this->label48 = (gcnew System::Windows::Forms::Label());

this->textBox56 = (gcnew System::Windows::Forms::TextBox());

this->label49 = (gcnew System::Windows::Forms::Label());

this->dataGridView7 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn31 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn32 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn33 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn34 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn35 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn36 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column14 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column15 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column16 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabPage8 = (gcnew System::Windows::Forms::TabPage());

this->textBox77 = (gcnew System::Windows::Forms::TextBox());

this->label69 = (gcnew System::Windows::Forms::Label());

this->textBox76 = (gcnew System::Windows::Forms::TextBox());

this->label68 = (gcnew System::Windows::Forms::Label());

this->button50 = (gcnew System::Windows::Forms::Button());

this->button51 = (gcnew System::Windows::Forms::Button());

this->button52 = (gcnew System::Windows::Forms::Button());

this->button53 = (gcnew System::Windows::Forms::Button());

this->button54 = (gcnew System::Windows::Forms::Button());

this->textBox57 = (gcnew System::Windows::Forms::TextBox());

this->button55 = (gcnew System::Windows::Forms::Button());

this->button56 = (gcnew System::Windows::Forms::Button());

this->textBox58 = (gcnew System::Windows::Forms::TextBox());

this->label50 = (gcnew System::Windows::Forms::Label());

this->textBox59 = (gcnew System::Windows::Forms::TextBox());

this->label51 = (gcnew System::Windows::Forms::Label());

this->textBox60 = (gcnew System::Windows::Forms::TextBox());

this->label52 = (gcnew System::Windows::Forms::Label());

this->textBox61 = (gcnew System::Windows::Forms::TextBox());

this->label53 = (gcnew System::Windows::Forms::Label());

this->textBox62 = (gcnew System::Windows::Forms::TextBox());

this->label54 = (gcnew System::Windows::Forms::Label());

this->textBox63 = (gcnew System::Windows::Forms::TextBox());

this->label55 = (gcnew System::Windows::Forms::Label());

this->textBox64 = (gcnew System::Windows::Forms::TextBox());

this->label56 = (gcnew System::Windows::Forms::Label());

this->dataGridView8 = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn37 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn38 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn39 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn40 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn41 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn42 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column17 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column18 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->tabControl1->SuspendLayout();

this->tabPage1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

this->tabPage2->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView2))->BeginInit();

this->tabPage3->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView3))->BeginInit();

this->tabPage4->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView4))->BeginInit();

this->tabPage5->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView5))->BeginInit();

this->tabPage6->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView6))->BeginInit();

this->tabPage7->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView7))->BeginInit();

this->tabPage8->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView8))->BeginInit();

this->SuspendLayout();

//

// tabControl1

//

this->tabControl1->Controls->Add(this->tabPage1);

this->tabControl1->Controls->Add(this->tabPage2);

this->tabControl1->Controls->Add(this->tabPage3);

this->tabControl1->Controls->Add(this->tabPage4);

this->tabControl1->Controls->Add(this->tabPage5);

this->tabControl1->Controls->Add(this->tabPage6);

this->tabControl1->Controls->Add(this->tabPage7);

this->tabControl1->Controls->Add(this->tabPage8);

this->tabControl1->Location = System::Drawing::Point(3, 5);

this->tabControl1->Name = L"tabControl1";

this->tabControl1->SelectedIndex = 0;

this->tabControl1->Size = System::Drawing::Size(1028, 452);

this->tabControl1->TabIndex = 0;

//

// tabPage1

//

this->tabPage1->Controls->Add(this->button4);

this->tabPage1->Controls->Add(this->button7);

this->tabPage1->Controls->Add(this->button6);

this->tabPage1->Controls->Add(this->button5);

this->tabPage1->Controls->Add(this->button3);

this->tabPage1->Controls->Add(this->textBox8);

this->tabPage1->Controls->Add(this->button2);

this->tabPage1->Controls->Add(this->button1);

this->tabPage1->Controls->Add(this->textBox7);

this->tabPage1->Controls->Add(this->label7);

this->tabPage1->Controls->Add(this->textBox6);

this->tabPage1->Controls->Add(this->label6);

this->tabPage1->Controls->Add(this->textBox5);

this->tabPage1->Controls->Add(this->label5);

this->tabPage1->Controls->Add(this->textBox4);

this->tabPage1->Controls->Add(this->label4);

this->tabPage1->Controls->Add(this->textBox3);

this->tabPage1->Controls->Add(this->label3);

this->tabPage1->Controls->Add(this->textBox2);

this->tabPage1->Controls->Add(this->label2);

this->tabPage1->Controls->Add(this->textBox1);

this->tabPage1->Controls->Add(this->label1);

this->tabPage1->Controls->Add(this->dataGridView1);

this->tabPage1->Location = System::Drawing::Point(4, 22);

this->tabPage1->Name = L"tabPage1";

this->tabPage1->Padding = System::Windows::Forms::Padding(3);

this->tabPage1->Size = System::Drawing::Size(1020, 426);

this->tabPage1->TabIndex = 0;

this->tabPage1->Text = L"Энергетические установки";

this->tabPage1->UseVisualStyleBackColor = true;

this->tabPage1->Click += gcnew System::EventHandler(this, &MyForm::tabPage1\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(140, 327);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(145, 23);

this->button4->TabIndex = 23;

this->button4->Text = L"Установить ошибку";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4\_Click\_1);

//

// button7

//

this->button7->Location = System::Drawing::Point(596, 37);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(139, 23);

this->button7->TabIndex = 22;

this->button7->Text = L"Загрузить из файла";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &MyForm::button7\_Click);

//

// button6

//

this->button6->Location = System::Drawing::Point(596, 8);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(139, 23);

this->button6->TabIndex = 21;

this->button6->Text = L"Записать в файл";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &MyForm::button6\_Click);

//

// button5

//

this->button5->Location = System::Drawing::Point(750, 37);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(264, 23);

this->button5->TabIndex = 20;

this->button5->Text = L"Получить справку";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &MyForm::button5\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(886, 8);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(128, 23);

this->button3->TabIndex = 18;

this->button3->Text = L"Поиск";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);

//

// textBox8

//

this->textBox8->Location = System::Drawing::Point(750, 10);

this->textBox8->Name = L"textBox8";

this->textBox8->Size = System::Drawing::Size(130, 20);

this->textBox8->TabIndex = 17;

//

// button2

//

this->button2->Location = System::Drawing::Point(6, 239);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(227, 23);

this->button2->TabIndex = 16;

this->button2->Text = L"Добавить";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// button1

//

this->button1->Location = System::Drawing::Point(10, 327);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(111, 23);

this->button1->TabIndex = 15;

this->button1->Text = L"Установить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// textBox7

//

this->textBox7->Location = System::Drawing::Point(10, 300);

this->textBox7->Name = L"textBox7";

this->textBox7->Size = System::Drawing::Size(111, 20);

this->textBox7->TabIndex = 14;

//

// label7

//

this->label7->AutoSize = true;

this->label7->Location = System::Drawing::Point(11, 284);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(110, 13);

this->label7->TabIndex = 13;

this->label7->Text = L"Текущая мощность:";

//

// textBox6

//

this->textBox6->Location = System::Drawing::Point(140, 193);

this->textBox6->Name = L"textBox6";

this->textBox6->Size = System::Drawing::Size(111, 20);

this->textBox6->TabIndex = 12;

//

// label6

//

this->label6->AutoSize = true;

this->label6->Location = System::Drawing::Point(141, 177);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(29, 13);

this->label6->TabIndex = 11;

this->label6->Text = L"Вес:";

//

// textBox5

//

this->textBox5->Location = System::Drawing::Point(140, 143);

this->textBox5->Name = L"textBox5";

this->textBox5->Size = System::Drawing::Size(111, 20);

this->textBox5->TabIndex = 10;

//

// label5

//

this->label5->AutoSize = true;

this->label5->Location = System::Drawing::Point(141, 127);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(49, 13);

this->label5->TabIndex = 9;

this->label5->Text = L"Ширина:";

//

// textBox4

//

this->textBox4->Location = System::Drawing::Point(6, 193);

this->textBox4->Name = L"textBox4";

this->textBox4->Size = System::Drawing::Size(111, 20);

this->textBox4->TabIndex = 8;

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(7, 177);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(48, 13);

this->label4->TabIndex = 7;

this->label4->Text = L"Выстоа:";

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(6, 143);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(111, 20);

this->textBox3->TabIndex = 6;

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(7, 127);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(43, 13);

this->label3->TabIndex = 5;

this->label3->Text = L"Длина:";

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(6, 95);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(143, 20);

this->textBox2->TabIndex = 4;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(7, 79);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(142, 13);

this->label2->TabIndex = 3;

this->label2->Text = L"Максимальная мощность:";

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(6, 34);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(111, 20);

this->textBox1->TabIndex = 2;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(11, 17);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(49, 13);

this->label1->TabIndex = 1;

this->label1->Text = L"Модель:";

//

// dataGridView1

//

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(6) {

this->Column1,

this->Column2, this->Column3, this->Column4, this->Column5, this->Column6

});

this->dataGridView1->Location = System::Drawing::Point(335, 66);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->Size = System::Drawing::Size(679, 353);

this->dataGridView1->TabIndex = 0;

//

// Column1

//

this->Column1->HeaderText = L"Модель";

this->Column1->Name = L"Column1";

this->Column1->ReadOnly = true;

//

// Column2

//

this->Column2->HeaderText = L"Макс.мощность";

this->Column2->Name = L"Column2";

this->Column2->ReadOnly = true;

//

// Column3

//

this->Column3->HeaderText = L"Текущая мощность";

this->Column3->Name = L"Column3";

this->Column3->ReadOnly = true;

//

// Column4

//

this->Column4->HeaderText = L"Обороты";

this->Column4->Name = L"Column4";

this->Column4->ReadOnly = true;

//

// Column5

//

this->Column5->HeaderText = L"Ошибки";

this->Column5->Name = L"Column5";

this->Column5->ReadOnly = true;

//

// Column6

//

this->Column6->HeaderText = L"Габбариты";

this->Column6->Name = L"Column6";

this->Column6->ReadOnly = true;

//

// tabPage2

//

this->tabPage2->Controls->Add(this->textBox65);

this->tabPage2->Controls->Add(this->label57);

this->tabPage2->Controls->Add(this->button8);

this->tabPage2->Controls->Add(this->button9);

this->tabPage2->Controls->Add(this->button10);

this->tabPage2->Controls->Add(this->button11);

this->tabPage2->Controls->Add(this->button12);

this->tabPage2->Controls->Add(this->textBox9);

this->tabPage2->Controls->Add(this->button13);

this->tabPage2->Controls->Add(this->button14);

this->tabPage2->Controls->Add(this->textBox10);

this->tabPage2->Controls->Add(this->label8);

this->tabPage2->Controls->Add(this->textBox11);

this->tabPage2->Controls->Add(this->label9);

this->tabPage2->Controls->Add(this->textBox12);

this->tabPage2->Controls->Add(this->label10);

this->tabPage2->Controls->Add(this->textBox13);

this->tabPage2->Controls->Add(this->label11);

this->tabPage2->Controls->Add(this->textBox14);

this->tabPage2->Controls->Add(this->label12);

this->tabPage2->Controls->Add(this->textBox15);

this->tabPage2->Controls->Add(this->label13);

this->tabPage2->Controls->Add(this->textBox16);

this->tabPage2->Controls->Add(this->label14);

this->tabPage2->Controls->Add(this->dataGridView2);

this->tabPage2->Location = System::Drawing::Point(4, 22);

this->tabPage2->Name = L"tabPage2";

this->tabPage2->Padding = System::Windows::Forms::Padding(3);

this->tabPage2->Size = System::Drawing::Size(1020, 426);

this->tabPage2->TabIndex = 1;

this->tabPage2->Text = L" Установки внутреннего сгорания";

this->tabPage2->UseVisualStyleBackColor = true;

//

// textBox65

//

this->textBox65->Location = System::Drawing::Point(179, 96);

this->textBox65->Name = L"textBox65";

this->textBox65->Size = System::Drawing::Size(111, 20);

this->textBox65->TabIndex = 48;

//

// label57

//

this->label57->AutoSize = true;

this->label57->Location = System::Drawing::Point(184, 79);

this->label57->Name = L"label57";

this->label57->Size = System::Drawing::Size(53, 13);

this->label57->TabIndex = 47;

this->label57->Text = L"Топливо:";

//

// button8

//

this->button8->Location = System::Drawing::Point(140, 327);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(145, 23);

this->button8->TabIndex = 46;

this->button8->Text = L"Установить ошибку";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &MyForm::button8\_Click);

//

// button9

//

this->button9->Location = System::Drawing::Point(596, 37);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(139, 23);

this->button9->TabIndex = 45;

this->button9->Text = L"Загрузить из файла";

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &MyForm::button9\_Click);

//

// button10

//

this->button10->Location = System::Drawing::Point(596, 8);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(139, 23);

this->button10->TabIndex = 44;

this->button10->Text = L"Записать в файл";

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &MyForm::button10\_Click);

//

// button11

//

this->button11->Location = System::Drawing::Point(750, 37);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(264, 23);

this->button11->TabIndex = 43;

this->button11->Text = L"Получить справку";

this->button11->UseVisualStyleBackColor = true;

this->button11->Click += gcnew System::EventHandler(this, &MyForm::button11\_Click);

//

// button12

//

this->button12->Location = System::Drawing::Point(886, 8);

this->button12->Name = L"button12";

this->button12->Size = System::Drawing::Size(128, 23);

this->button12->TabIndex = 42;

this->button12->Text = L"Поиск";

this->button12->UseVisualStyleBackColor = true;

this->button12->Click += gcnew System::EventHandler(this, &MyForm::button12\_Click);

//

// textBox9

//

this->textBox9->Location = System::Drawing::Point(750, 10);

this->textBox9->Name = L"textBox9";

this->textBox9->Size = System::Drawing::Size(130, 20);

this->textBox9->TabIndex = 41;

//

// button13

//

this->button13->Location = System::Drawing::Point(6, 239);

this->button13->Name = L"button13";

this->button13->Size = System::Drawing::Size(227, 23);

this->button13->TabIndex = 40;

this->button13->Text = L"Добавить";

this->button13->UseVisualStyleBackColor = true;

this->button13->Click += gcnew System::EventHandler(this, &MyForm::button13\_Click);

//

// button14

//

this->button14->Location = System::Drawing::Point(10, 327);

this->button14->Name = L"button14";

this->button14->Size = System::Drawing::Size(111, 23);

this->button14->TabIndex = 39;

this->button14->Text = L"Установить";

this->button14->UseVisualStyleBackColor = true;

this->button14->Click += gcnew System::EventHandler(this, &MyForm::button14\_Click);

//

// textBox10

//

this->textBox10->Location = System::Drawing::Point(10, 300);

this->textBox10->Name = L"textBox10";

this->textBox10->Size = System::Drawing::Size(111, 20);

this->textBox10->TabIndex = 38;

//

// label8

//

this->label8->AutoSize = true;

this->label8->Location = System::Drawing::Point(11, 284);

this->label8->Name = L"label8";

this->label8->Size = System::Drawing::Size(110, 13);

this->label8->TabIndex = 37;

this->label8->Text = L"Текущая мощность:";

//

// textBox11

//

this->textBox11->Location = System::Drawing::Point(140, 193);

this->textBox11->Name = L"textBox11";

this->textBox11->Size = System::Drawing::Size(111, 20);

this->textBox11->TabIndex = 36;

//

// label9

//

this->label9->AutoSize = true;

this->label9->Location = System::Drawing::Point(141, 177);

this->label9->Name = L"label9";

this->label9->Size = System::Drawing::Size(29, 13);

this->label9->TabIndex = 35;

this->label9->Text = L"Вес:";

//

// textBox12

//

this->textBox12->Location = System::Drawing::Point(140, 143);

this->textBox12->Name = L"textBox12";

this->textBox12->Size = System::Drawing::Size(111, 20);

this->textBox12->TabIndex = 34;

//

// label10

//

this->label10->AutoSize = true;

this->label10->Location = System::Drawing::Point(141, 127);

this->label10->Name = L"label10";

this->label10->Size = System::Drawing::Size(49, 13);

this->label10->TabIndex = 33;

this->label10->Text = L"Ширина:";

//

// textBox13

//

this->textBox13->Location = System::Drawing::Point(6, 193);

this->textBox13->Name = L"textBox13";

this->textBox13->Size = System::Drawing::Size(111, 20);

this->textBox13->TabIndex = 32;

//

// label11

//

this->label11->AutoSize = true;

this->label11->Location = System::Drawing::Point(7, 177);

this->label11->Name = L"label11";

this->label11->Size = System::Drawing::Size(48, 13);

this->label11->TabIndex = 31;

this->label11->Text = L"Выстоа:";

//

// textBox14

//

this->textBox14->Location = System::Drawing::Point(6, 143);

this->textBox14->Name = L"textBox14";

this->textBox14->Size = System::Drawing::Size(111, 20);

this->textBox14->TabIndex = 30;

//

// label12

//

this->label12->AutoSize = true;

this->label12->Location = System::Drawing::Point(7, 127);

this->label12->Name = L"label12";

this->label12->Size = System::Drawing::Size(43, 13);

this->label12->TabIndex = 29;

this->label12->Text = L"Длина:";

//

// textBox15

//

this->textBox15->Location = System::Drawing::Point(6, 95);

this->textBox15->Name = L"textBox15";

this->textBox15->Size = System::Drawing::Size(143, 20);

this->textBox15->TabIndex = 28;

//

// label13

//

this->label13->AutoSize = true;

this->label13->Location = System::Drawing::Point(7, 79);

this->label13->Name = L"label13";

this->label13->Size = System::Drawing::Size(142, 13);

this->label13->TabIndex = 27;

this->label13->Text = L"Максимальная мощность:";

//

// textBox16

//

this->textBox16->Location = System::Drawing::Point(6, 34);

this->textBox16->Name = L"textBox16";

this->textBox16->Size = System::Drawing::Size(111, 20);

this->textBox16->TabIndex = 26;

//

// label14

//

this->label14->AutoSize = true;

this->label14->Location = System::Drawing::Point(11, 17);

this->label14->Name = L"label14";

this->label14->Size = System::Drawing::Size(49, 13);

this->label14->TabIndex = 25;

this->label14->Text = L"Модель:";

//

// dataGridView2

//

this->dataGridView2->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView2->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(7) {

this->dataGridViewTextBoxColumn1,

this->dataGridViewTextBoxColumn2, this->dataGridViewTextBoxColumn3, this->dataGridViewTextBoxColumn4, this->dataGridViewTextBoxColumn5,

this->dataGridViewTextBoxColumn6, this->dataGridViewTextBoxColumn43

});

this->dataGridView2->Location = System::Drawing::Point(370, 67);

this->dataGridView2->Name = L"dataGridView2";

this->dataGridView2->Size = System::Drawing::Size(644, 353);

this->dataGridView2->TabIndex = 24;

//

// dataGridViewTextBoxColumn1

//

this->dataGridViewTextBoxColumn1->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn1->Name = L"dataGridViewTextBoxColumn1";

this->dataGridViewTextBoxColumn1->ReadOnly = true;

//

// dataGridViewTextBoxColumn2

//

this->dataGridViewTextBoxColumn2->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn2->Name = L"dataGridViewTextBoxColumn2";

this->dataGridViewTextBoxColumn2->ReadOnly = true;

//

// dataGridViewTextBoxColumn3

//

this->dataGridViewTextBoxColumn3->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn3->Name = L"dataGridViewTextBoxColumn3";

this->dataGridViewTextBoxColumn3->ReadOnly = true;

//

// dataGridViewTextBoxColumn4

//

this->dataGridViewTextBoxColumn4->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn4->Name = L"dataGridViewTextBoxColumn4";

this->dataGridViewTextBoxColumn4->ReadOnly = true;

//

// dataGridViewTextBoxColumn5

//

this->dataGridViewTextBoxColumn5->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn5->Name = L"dataGridViewTextBoxColumn5";

this->dataGridViewTextBoxColumn5->ReadOnly = true;

//

// dataGridViewTextBoxColumn6

//

this->dataGridViewTextBoxColumn6->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn6->Name = L"dataGridViewTextBoxColumn6";

this->dataGridViewTextBoxColumn6->ReadOnly = true;

//

// dataGridViewTextBoxColumn43

//

this->dataGridViewTextBoxColumn43->HeaderText = L"Топливо";

this->dataGridViewTextBoxColumn43->Name = L"dataGridViewTextBoxColumn43";

//

// tabPage3

//

this->tabPage3->Controls->Add(this->textBox66);

this->tabPage3->Controls->Add(this->label58);

this->tabPage3->Controls->Add(this->button15);

this->tabPage3->Controls->Add(this->button16);

this->tabPage3->Controls->Add(this->button17);

this->tabPage3->Controls->Add(this->button18);

this->tabPage3->Controls->Add(this->button19);

this->tabPage3->Controls->Add(this->textBox17);

this->tabPage3->Controls->Add(this->button20);

this->tabPage3->Controls->Add(this->button21);

this->tabPage3->Controls->Add(this->textBox18);

this->tabPage3->Controls->Add(this->label15);

this->tabPage3->Controls->Add(this->textBox19);

this->tabPage3->Controls->Add(this->label16);

this->tabPage3->Controls->Add(this->textBox20);

this->tabPage3->Controls->Add(this->label17);

this->tabPage3->Controls->Add(this->textBox21);

this->tabPage3->Controls->Add(this->label18);

this->tabPage3->Controls->Add(this->textBox22);

this->tabPage3->Controls->Add(this->label19);

this->tabPage3->Controls->Add(this->textBox23);

this->tabPage3->Controls->Add(this->label20);

this->tabPage3->Controls->Add(this->textBox24);

this->tabPage3->Controls->Add(this->label21);

this->tabPage3->Controls->Add(this->dataGridView3);

this->tabPage3->Location = System::Drawing::Point(4, 22);

this->tabPage3->Name = L"tabPage3";

this->tabPage3->Size = System::Drawing::Size(1020, 426);

this->tabPage3->TabIndex = 2;

this->tabPage3->Text = L"Альтернативные ";

this->tabPage3->UseVisualStyleBackColor = true;

//

// textBox66

//

this->textBox66->Location = System::Drawing::Point(179, 96);

this->textBox66->Name = L"textBox66";

this->textBox66->Size = System::Drawing::Size(111, 20);

this->textBox66->TabIndex = 48;

//

// label58

//

this->label58->AutoSize = true;

this->label58->Location = System::Drawing::Point(184, 79);

this->label58->Name = L"label58";

this->label58->Size = System::Drawing::Size(58, 13);

this->label58->TabIndex = 47;

this->label58->Text = L"Источник:";

//

// button15

//

this->button15->Location = System::Drawing::Point(140, 327);

this->button15->Name = L"button15";

this->button15->Size = System::Drawing::Size(145, 23);

this->button15->TabIndex = 46;

this->button15->Text = L"Установить ошибку";

this->button15->UseVisualStyleBackColor = true;

this->button15->Click += gcnew System::EventHandler(this, &MyForm::button15\_Click);

//

// button16

//

this->button16->Location = System::Drawing::Point(596, 37);

this->button16->Name = L"button16";

this->button16->Size = System::Drawing::Size(139, 23);

this->button16->TabIndex = 45;

this->button16->Text = L"Загрузить из файла";

this->button16->UseVisualStyleBackColor = true;

this->button16->Click += gcnew System::EventHandler(this, &MyForm::button16\_Click);

//

// button17

//

this->button17->Location = System::Drawing::Point(596, 8);

this->button17->Name = L"button17";

this->button17->Size = System::Drawing::Size(139, 23);

this->button17->TabIndex = 44;

this->button17->Text = L"Записать в файл";

this->button17->UseVisualStyleBackColor = true;

this->button17->Click += gcnew System::EventHandler(this, &MyForm::button17\_Click);

//

// button18

//

this->button18->Location = System::Drawing::Point(750, 37);

this->button18->Name = L"button18";

this->button18->Size = System::Drawing::Size(264, 23);

this->button18->TabIndex = 43;

this->button18->Text = L"Получить справку";

this->button18->UseVisualStyleBackColor = true;

this->button18->Click += gcnew System::EventHandler(this, &MyForm::button18\_Click);

//

// button19

//

this->button19->Location = System::Drawing::Point(886, 8);

this->button19->Name = L"button19";

this->button19->Size = System::Drawing::Size(128, 23);

this->button19->TabIndex = 42;

this->button19->Text = L"Поиск";

this->button19->UseVisualStyleBackColor = true;

this->button19->Click += gcnew System::EventHandler(this, &MyForm::button19\_Click);

//

// textBox17

//

this->textBox17->Location = System::Drawing::Point(750, 10);

this->textBox17->Name = L"textBox17";

this->textBox17->Size = System::Drawing::Size(130, 20);

this->textBox17->TabIndex = 41;

//

// button20

//

this->button20->Location = System::Drawing::Point(6, 239);

this->button20->Name = L"button20";

this->button20->Size = System::Drawing::Size(227, 23);

this->button20->TabIndex = 40;

this->button20->Text = L"Добавить";

this->button20->UseVisualStyleBackColor = true;

this->button20->Click += gcnew System::EventHandler(this, &MyForm::button20\_Click);

//

// button21

//

this->button21->Location = System::Drawing::Point(10, 327);

this->button21->Name = L"button21";

this->button21->Size = System::Drawing::Size(111, 23);

this->button21->TabIndex = 39;

this->button21->Text = L"Установить";

this->button21->UseVisualStyleBackColor = true;

this->button21->Click += gcnew System::EventHandler(this, &MyForm::button21\_Click);

//

// textBox18

//

this->textBox18->Location = System::Drawing::Point(10, 300);

this->textBox18->Name = L"textBox18";

this->textBox18->Size = System::Drawing::Size(111, 20);

this->textBox18->TabIndex = 38;

//

// label15

//

this->label15->AutoSize = true;

this->label15->Location = System::Drawing::Point(11, 284);

this->label15->Name = L"label15";

this->label15->Size = System::Drawing::Size(110, 13);

this->label15->TabIndex = 37;

this->label15->Text = L"Текущая мощность:";

//

// textBox19

//

this->textBox19->Location = System::Drawing::Point(140, 193);

this->textBox19->Name = L"textBox19";

this->textBox19->Size = System::Drawing::Size(111, 20);

this->textBox19->TabIndex = 36;

//

// label16

//

this->label16->AutoSize = true;

this->label16->Location = System::Drawing::Point(141, 177);

this->label16->Name = L"label16";

this->label16->Size = System::Drawing::Size(29, 13);

this->label16->TabIndex = 35;

this->label16->Text = L"Вес:";

//

// textBox20

//

this->textBox20->Location = System::Drawing::Point(140, 143);

this->textBox20->Name = L"textBox20";

this->textBox20->Size = System::Drawing::Size(111, 20);

this->textBox20->TabIndex = 34;

//

// label17

//

this->label17->AutoSize = true;

this->label17->Location = System::Drawing::Point(141, 127);

this->label17->Name = L"label17";

this->label17->Size = System::Drawing::Size(49, 13);

this->label17->TabIndex = 33;

this->label17->Text = L"Ширина:";

//

// textBox21

//

this->textBox21->Location = System::Drawing::Point(6, 193);

this->textBox21->Name = L"textBox21";

this->textBox21->Size = System::Drawing::Size(111, 20);

this->textBox21->TabIndex = 32;

//

// label18

//

this->label18->AutoSize = true;

this->label18->Location = System::Drawing::Point(7, 177);

this->label18->Name = L"label18";

this->label18->Size = System::Drawing::Size(48, 13);

this->label18->TabIndex = 31;

this->label18->Text = L"Выстоа:";

//

// textBox22

//

this->textBox22->Location = System::Drawing::Point(6, 143);

this->textBox22->Name = L"textBox22";

this->textBox22->Size = System::Drawing::Size(111, 20);

this->textBox22->TabIndex = 30;

//

// label19

//

this->label19->AutoSize = true;

this->label19->Location = System::Drawing::Point(7, 127);

this->label19->Name = L"label19";

this->label19->Size = System::Drawing::Size(43, 13);

this->label19->TabIndex = 29;

this->label19->Text = L"Длина:";

//

// textBox23

//

this->textBox23->Location = System::Drawing::Point(6, 95);

this->textBox23->Name = L"textBox23";

this->textBox23->Size = System::Drawing::Size(143, 20);

this->textBox23->TabIndex = 28;

//

// label20

//

this->label20->AutoSize = true;

this->label20->Location = System::Drawing::Point(7, 79);

this->label20->Name = L"label20";

this->label20->Size = System::Drawing::Size(142, 13);

this->label20->TabIndex = 27;

this->label20->Text = L"Максимальная мощность:";

//

// textBox24

//

this->textBox24->Location = System::Drawing::Point(6, 34);

this->textBox24->Name = L"textBox24";

this->textBox24->Size = System::Drawing::Size(111, 20);

this->textBox24->TabIndex = 26;

//

// label21

//

this->label21->AutoSize = true;

this->label21->Location = System::Drawing::Point(11, 17);

this->label21->Name = L"label21";

this->label21->Size = System::Drawing::Size(49, 13);

this->label21->TabIndex = 25;

this->label21->Text = L"Модель:";

//

// dataGridView3

//

this->dataGridView3->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView3->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(7) {

this->dataGridViewTextBoxColumn7,

this->dataGridViewTextBoxColumn8, this->dataGridViewTextBoxColumn9, this->dataGridViewTextBoxColumn10, this->dataGridViewTextBoxColumn11,

this->dataGridViewTextBoxColumn12, this->Column7

});

this->dataGridView3->Location = System::Drawing::Point(370, 66);

this->dataGridView3->Name = L"dataGridView3";

this->dataGridView3->Size = System::Drawing::Size(644, 353);

this->dataGridView3->TabIndex = 24;

//

// dataGridViewTextBoxColumn7

//

this->dataGridViewTextBoxColumn7->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn7->Name = L"dataGridViewTextBoxColumn7";

this->dataGridViewTextBoxColumn7->ReadOnly = true;

//

// dataGridViewTextBoxColumn8

//

this->dataGridViewTextBoxColumn8->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn8->Name = L"dataGridViewTextBoxColumn8";

this->dataGridViewTextBoxColumn8->ReadOnly = true;

//

// dataGridViewTextBoxColumn9

//

this->dataGridViewTextBoxColumn9->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn9->Name = L"dataGridViewTextBoxColumn9";

this->dataGridViewTextBoxColumn9->ReadOnly = true;

//

// dataGridViewTextBoxColumn10

//

this->dataGridViewTextBoxColumn10->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn10->Name = L"dataGridViewTextBoxColumn10";

this->dataGridViewTextBoxColumn10->ReadOnly = true;

//

// dataGridViewTextBoxColumn11

//

this->dataGridViewTextBoxColumn11->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn11->Name = L"dataGridViewTextBoxColumn11";

this->dataGridViewTextBoxColumn11->ReadOnly = true;

//

// dataGridViewTextBoxColumn12

//

this->dataGridViewTextBoxColumn12->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn12->Name = L"dataGridViewTextBoxColumn12";

this->dataGridViewTextBoxColumn12->ReadOnly = true;

//

// Column7

//

this->Column7->HeaderText = L"Источник";

this->Column7->Name = L"Column7";

//

// tabPage4

//

this->tabPage4->Controls->Add(this->textBox68);

this->tabPage4->Controls->Add(this->label60);

this->tabPage4->Controls->Add(this->textBox67);

this->tabPage4->Controls->Add(this->label59);

this->tabPage4->Controls->Add(this->button22);

this->tabPage4->Controls->Add(this->button23);

this->tabPage4->Controls->Add(this->button24);

this->tabPage4->Controls->Add(this->button25);

this->tabPage4->Controls->Add(this->button26);

this->tabPage4->Controls->Add(this->textBox25);

this->tabPage4->Controls->Add(this->button27);

this->tabPage4->Controls->Add(this->button28);

this->tabPage4->Controls->Add(this->textBox26);

this->tabPage4->Controls->Add(this->label22);

this->tabPage4->Controls->Add(this->textBox27);

this->tabPage4->Controls->Add(this->label23);

this->tabPage4->Controls->Add(this->textBox28);

this->tabPage4->Controls->Add(this->label24);

this->tabPage4->Controls->Add(this->textBox29);

this->tabPage4->Controls->Add(this->label25);

this->tabPage4->Controls->Add(this->textBox30);

this->tabPage4->Controls->Add(this->label26);

this->tabPage4->Controls->Add(this->textBox31);

this->tabPage4->Controls->Add(this->label27);

this->tabPage4->Controls->Add(this->textBox32);

this->tabPage4->Controls->Add(this->label28);

this->tabPage4->Controls->Add(this->dataGridView4);

this->tabPage4->Location = System::Drawing::Point(4, 22);

this->tabPage4->Name = L"tabPage4";

this->tabPage4->Size = System::Drawing::Size(1020, 426);

this->tabPage4->TabIndex = 3;

this->tabPage4->Text = L"Газо-турбинные";

this->tabPage4->UseVisualStyleBackColor = true;

//

// textBox68

//

this->textBox68->Location = System::Drawing::Point(162, 34);

this->textBox68->Name = L"textBox68";

this->textBox68->Size = System::Drawing::Size(111, 20);

this->textBox68->TabIndex = 50;

//

// label60

//

this->label60->AutoSize = true;

this->label60->Location = System::Drawing::Point(126, 17);

this->label60->Name = L"label60";

this->label60->Size = System::Drawing::Size(147, 13);

this->label60->TabIndex = 49;

this->label60->Text = L"Содержание кислорода (%):";

//

// textBox67

//

this->textBox67->Location = System::Drawing::Point(162, 96);

this->textBox67->Name = L"textBox67";

this->textBox67->Size = System::Drawing::Size(111, 20);

this->textBox67->TabIndex = 48;

//

// label59

//

this->label59->AutoSize = true;

this->label59->Location = System::Drawing::Point(167, 79);

this->label59->Name = L"label59";

this->label59->Size = System::Drawing::Size(53, 13);

this->label59->TabIndex = 47;

this->label59->Text = L"Топливо:";

//

// button22

//

this->button22->Location = System::Drawing::Point(140, 327);

this->button22->Name = L"button22";

this->button22->Size = System::Drawing::Size(145, 23);

this->button22->TabIndex = 46;

this->button22->Text = L"Установить ошибку";

this->button22->UseVisualStyleBackColor = true;

this->button22->Click += gcnew System::EventHandler(this, &MyForm::button22\_Click);

//

// button23

//

this->button23->Location = System::Drawing::Point(596, 37);

this->button23->Name = L"button23";

this->button23->Size = System::Drawing::Size(139, 23);

this->button23->TabIndex = 45;

this->button23->Text = L"Загрузить из файла";

this->button23->UseVisualStyleBackColor = true;

this->button23->Click += gcnew System::EventHandler(this, &MyForm::button23\_Click);

//

// button24

//

this->button24->Location = System::Drawing::Point(596, 8);

this->button24->Name = L"button24";

this->button24->Size = System::Drawing::Size(139, 23);

this->button24->TabIndex = 44;

this->button24->Text = L"Записать в файл";

this->button24->UseVisualStyleBackColor = true;

this->button24->Click += gcnew System::EventHandler(this, &MyForm::button24\_Click);

//

// button25

//

this->button25->Location = System::Drawing::Point(750, 37);

this->button25->Name = L"button25";

this->button25->Size = System::Drawing::Size(264, 23);

this->button25->TabIndex = 43;

this->button25->Text = L"Получить справку";

this->button25->UseVisualStyleBackColor = true;

this->button25->Click += gcnew System::EventHandler(this, &MyForm::button25\_Click);

//

// button26

//

this->button26->Location = System::Drawing::Point(886, 8);

this->button26->Name = L"button26";

this->button26->Size = System::Drawing::Size(128, 23);

this->button26->TabIndex = 42;

this->button26->Text = L"Поиск";

this->button26->UseVisualStyleBackColor = true;

this->button26->Click += gcnew System::EventHandler(this, &MyForm::button26\_Click);

//

// textBox25

//

this->textBox25->Location = System::Drawing::Point(750, 10);

this->textBox25->Name = L"textBox25";

this->textBox25->Size = System::Drawing::Size(130, 20);

this->textBox25->TabIndex = 41;

//

// button27

//

this->button27->Location = System::Drawing::Point(6, 239);

this->button27->Name = L"button27";

this->button27->Size = System::Drawing::Size(227, 23);

this->button27->TabIndex = 40;

this->button27->Text = L"Добавить";

this->button27->UseVisualStyleBackColor = true;

this->button27->Click += gcnew System::EventHandler(this, &MyForm::button27\_Click);

//

// button28

//

this->button28->Location = System::Drawing::Point(10, 327);

this->button28->Name = L"button28";

this->button28->Size = System::Drawing::Size(111, 23);

this->button28->TabIndex = 39;

this->button28->Text = L"Установить";

this->button28->UseVisualStyleBackColor = true;

this->button28->Click += gcnew System::EventHandler(this, &MyForm::button28\_Click);

//

// textBox26

//

this->textBox26->Location = System::Drawing::Point(10, 300);

this->textBox26->Name = L"textBox26";

this->textBox26->Size = System::Drawing::Size(111, 20);

this->textBox26->TabIndex = 38;

//

// label22

//

this->label22->AutoSize = true;

this->label22->Location = System::Drawing::Point(11, 284);

this->label22->Name = L"label22";

this->label22->Size = System::Drawing::Size(110, 13);

this->label22->TabIndex = 37;

this->label22->Text = L"Текущая мощность:";

//

// textBox27

//

this->textBox27->Location = System::Drawing::Point(140, 193);

this->textBox27->Name = L"textBox27";

this->textBox27->Size = System::Drawing::Size(111, 20);

this->textBox27->TabIndex = 36;

//

// label23

//

this->label23->AutoSize = true;

this->label23->Location = System::Drawing::Point(141, 177);

this->label23->Name = L"label23";

this->label23->Size = System::Drawing::Size(29, 13);

this->label23->TabIndex = 35;

this->label23->Text = L"Вес:";

//

// textBox28

//

this->textBox28->Location = System::Drawing::Point(140, 143);

this->textBox28->Name = L"textBox28";

this->textBox28->Size = System::Drawing::Size(111, 20);

this->textBox28->TabIndex = 34;

//

// label24

//

this->label24->AutoSize = true;

this->label24->Location = System::Drawing::Point(141, 127);

this->label24->Name = L"label24";

this->label24->Size = System::Drawing::Size(49, 13);

this->label24->TabIndex = 33;

this->label24->Text = L"Ширина:";

//

// textBox29

//

this->textBox29->Location = System::Drawing::Point(6, 193);

this->textBox29->Name = L"textBox29";

this->textBox29->Size = System::Drawing::Size(111, 20);

this->textBox29->TabIndex = 32;

//

// label25

//

this->label25->AutoSize = true;

this->label25->Location = System::Drawing::Point(7, 177);

this->label25->Name = L"label25";

this->label25->Size = System::Drawing::Size(48, 13);

this->label25->TabIndex = 31;

this->label25->Text = L"Выстоа:";

//

// textBox30

//

this->textBox30->Location = System::Drawing::Point(6, 143);

this->textBox30->Name = L"textBox30";

this->textBox30->Size = System::Drawing::Size(111, 20);

this->textBox30->TabIndex = 30;

//

// label26

//

this->label26->AutoSize = true;

this->label26->Location = System::Drawing::Point(7, 127);

this->label26->Name = L"label26";

this->label26->Size = System::Drawing::Size(43, 13);

this->label26->TabIndex = 29;

this->label26->Text = L"Длина:";

//

// textBox31

//

this->textBox31->Location = System::Drawing::Point(6, 95);

this->textBox31->Name = L"textBox31";

this->textBox31->Size = System::Drawing::Size(143, 20);

this->textBox31->TabIndex = 28;

//

// label27

//

this->label27->AutoSize = true;

this->label27->Location = System::Drawing::Point(7, 79);

this->label27->Name = L"label27";

this->label27->Size = System::Drawing::Size(142, 13);

this->label27->TabIndex = 27;

this->label27->Text = L"Максимальная мощность:";

//

// textBox32

//

this->textBox32->Location = System::Drawing::Point(6, 34);

this->textBox32->Name = L"textBox32";

this->textBox32->Size = System::Drawing::Size(111, 20);

this->textBox32->TabIndex = 26;

//

// label28

//

this->label28->AutoSize = true;

this->label28->Location = System::Drawing::Point(11, 17);

this->label28->Name = L"label28";

this->label28->Size = System::Drawing::Size(49, 13);

this->label28->TabIndex = 25;

this->label28->Text = L"Модель:";

//

// dataGridView4

//

this->dataGridView4->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView4->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(8) {

this->dataGridViewTextBoxColumn13,

this->dataGridViewTextBoxColumn14, this->dataGridViewTextBoxColumn15, this->dataGridViewTextBoxColumn16, this->dataGridViewTextBoxColumn17,

this->dataGridViewTextBoxColumn18, this->Column8, this->Column9

});

this->dataGridView4->Location = System::Drawing::Point(370, 66);

this->dataGridView4->Name = L"dataGridView4";

this->dataGridView4->Size = System::Drawing::Size(644, 353);

this->dataGridView4->TabIndex = 24;

//

// dataGridViewTextBoxColumn13

//

this->dataGridViewTextBoxColumn13->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn13->Name = L"dataGridViewTextBoxColumn13";

this->dataGridViewTextBoxColumn13->ReadOnly = true;

//

// dataGridViewTextBoxColumn14

//

this->dataGridViewTextBoxColumn14->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn14->Name = L"dataGridViewTextBoxColumn14";

this->dataGridViewTextBoxColumn14->ReadOnly = true;

//

// dataGridViewTextBoxColumn15

//

this->dataGridViewTextBoxColumn15->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn15->Name = L"dataGridViewTextBoxColumn15";

this->dataGridViewTextBoxColumn15->ReadOnly = true;

//

// dataGridViewTextBoxColumn16

//

this->dataGridViewTextBoxColumn16->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn16->Name = L"dataGridViewTextBoxColumn16";

this->dataGridViewTextBoxColumn16->ReadOnly = true;

//

// dataGridViewTextBoxColumn17

//

this->dataGridViewTextBoxColumn17->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn17->Name = L"dataGridViewTextBoxColumn17";

this->dataGridViewTextBoxColumn17->ReadOnly = true;

//

// dataGridViewTextBoxColumn18

//

this->dataGridViewTextBoxColumn18->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn18->Name = L"dataGridViewTextBoxColumn18";

this->dataGridViewTextBoxColumn18->ReadOnly = true;

//

// Column8

//

this->Column8->HeaderText = L"Топливо";

this->Column8->Name = L"Column8";

//

// Column9

//

this->Column9->HeaderText = L"Содержание кислорода";

this->Column9->Name = L"Column9";

//

// tabPage5

//

this->tabPage5->Controls->Add(this->textBox69);

this->tabPage5->Controls->Add(this->label61);

this->tabPage5->Controls->Add(this->textBox70);

this->tabPage5->Controls->Add(this->label62);

this->tabPage5->Controls->Add(this->button29);

this->tabPage5->Controls->Add(this->button30);

this->tabPage5->Controls->Add(this->button31);

this->tabPage5->Controls->Add(this->button32);

this->tabPage5->Controls->Add(this->button33);

this->tabPage5->Controls->Add(this->textBox33);

this->tabPage5->Controls->Add(this->button34);

this->tabPage5->Controls->Add(this->button35);

this->tabPage5->Controls->Add(this->textBox34);

this->tabPage5->Controls->Add(this->label29);

this->tabPage5->Controls->Add(this->textBox35);

this->tabPage5->Controls->Add(this->label30);

this->tabPage5->Controls->Add(this->textBox36);

this->tabPage5->Controls->Add(this->label31);

this->tabPage5->Controls->Add(this->textBox37);

this->tabPage5->Controls->Add(this->label32);

this->tabPage5->Controls->Add(this->textBox38);

this->tabPage5->Controls->Add(this->label33);

this->tabPage5->Controls->Add(this->textBox39);

this->tabPage5->Controls->Add(this->label34);

this->tabPage5->Controls->Add(this->textBox40);

this->tabPage5->Controls->Add(this->label35);

this->tabPage5->Controls->Add(this->dataGridView5);

this->tabPage5->Location = System::Drawing::Point(4, 22);

this->tabPage5->Name = L"tabPage5";

this->tabPage5->Size = System::Drawing::Size(1020, 426);

this->tabPage5->TabIndex = 4;

this->tabPage5->Text = L"ДВС";

this->tabPage5->UseVisualStyleBackColor = true;

//

// textBox69

//

this->textBox69->Location = System::Drawing::Point(174, 34);

this->textBox69->Name = L"textBox69";

this->textBox69->Size = System::Drawing::Size(111, 20);

this->textBox69->TabIndex = 54;

//

// label61

//

this->label61->AutoSize = true;

this->label61->Location = System::Drawing::Point(179, 18);

this->label61->Name = L"label61";

this->label61->Size = System::Drawing::Size(73, 13);

this->label61->TabIndex = 53;

this->label61->Text = L"Тип топлива:";

//

// textBox70

//

this->textBox70->Location = System::Drawing::Point(174, 95);

this->textBox70->Name = L"textBox70";

this->textBox70->Size = System::Drawing::Size(111, 20);

this->textBox70->TabIndex = 52;

//

// label62

//

this->label62->AutoSize = true;

this->label62->Location = System::Drawing::Point(179, 79);

this->label62->Name = L"label62";

this->label62->Size = System::Drawing::Size(53, 13);

this->label62->TabIndex = 51;

this->label62->Text = L"Топливо:";

//

// button29

//

this->button29->Location = System::Drawing::Point(140, 327);

this->button29->Name = L"button29";

this->button29->Size = System::Drawing::Size(145, 23);

this->button29->TabIndex = 46;

this->button29->Text = L"Установить ошибку";

this->button29->UseVisualStyleBackColor = true;

this->button29->Click += gcnew System::EventHandler(this, &MyForm::button29\_Click);

//

// button30

//

this->button30->Location = System::Drawing::Point(596, 37);

this->button30->Name = L"button30";

this->button30->Size = System::Drawing::Size(139, 23);

this->button30->TabIndex = 45;

this->button30->Text = L"Загрузить из файла";

this->button30->UseVisualStyleBackColor = true;

this->button30->Click += gcnew System::EventHandler(this, &MyForm::button30\_Click);

//

// button31

//

this->button31->Location = System::Drawing::Point(596, 8);

this->button31->Name = L"button31";

this->button31->Size = System::Drawing::Size(139, 23);

this->button31->TabIndex = 44;

this->button31->Text = L"Записать в файл";

this->button31->UseVisualStyleBackColor = true;

this->button31->Click += gcnew System::EventHandler(this, &MyForm::button31\_Click);

//

// button32

//

this->button32->Location = System::Drawing::Point(750, 37);

this->button32->Name = L"button32";

this->button32->Size = System::Drawing::Size(264, 23);

this->button32->TabIndex = 43;

this->button32->Text = L"Получить справку";

this->button32->UseVisualStyleBackColor = true;

this->button32->Click += gcnew System::EventHandler(this, &MyForm::button32\_Click);

//

// button33

//

this->button33->Location = System::Drawing::Point(886, 8);

this->button33->Name = L"button33";

this->button33->Size = System::Drawing::Size(128, 23);

this->button33->TabIndex = 42;

this->button33->Text = L"Поиск";

this->button33->UseVisualStyleBackColor = true;

this->button33->Click += gcnew System::EventHandler(this, &MyForm::button33\_Click);

//

// textBox33

//

this->textBox33->Location = System::Drawing::Point(750, 10);

this->textBox33->Name = L"textBox33";

this->textBox33->Size = System::Drawing::Size(130, 20);

this->textBox33->TabIndex = 41;

//

// button34

//

this->button34->Location = System::Drawing::Point(6, 239);

this->button34->Name = L"button34";

this->button34->Size = System::Drawing::Size(227, 23);

this->button34->TabIndex = 40;

this->button34->Text = L"Добавить";

this->button34->UseVisualStyleBackColor = true;

this->button34->Click += gcnew System::EventHandler(this, &MyForm::button34\_Click);

//

// button35

//

this->button35->Location = System::Drawing::Point(10, 327);

this->button35->Name = L"button35";

this->button35->Size = System::Drawing::Size(111, 23);

this->button35->TabIndex = 39;

this->button35->Text = L"Установить";

this->button35->UseVisualStyleBackColor = true;

this->button35->Click += gcnew System::EventHandler(this, &MyForm::button35\_Click);

//

// textBox34

//

this->textBox34->Location = System::Drawing::Point(10, 300);

this->textBox34->Name = L"textBox34";

this->textBox34->Size = System::Drawing::Size(111, 20);

this->textBox34->TabIndex = 38;

//

// label29

//

this->label29->AutoSize = true;

this->label29->Location = System::Drawing::Point(11, 284);

this->label29->Name = L"label29";

this->label29->Size = System::Drawing::Size(110, 13);

this->label29->TabIndex = 37;

this->label29->Text = L"Текущая мощность:";

//

// textBox35

//

this->textBox35->Location = System::Drawing::Point(140, 193);

this->textBox35->Name = L"textBox35";

this->textBox35->Size = System::Drawing::Size(111, 20);

this->textBox35->TabIndex = 36;

//

// label30

//

this->label30->AutoSize = true;

this->label30->Location = System::Drawing::Point(141, 177);

this->label30->Name = L"label30";

this->label30->Size = System::Drawing::Size(29, 13);

this->label30->TabIndex = 35;

this->label30->Text = L"Вес:";

//

// textBox36

//

this->textBox36->Location = System::Drawing::Point(140, 143);

this->textBox36->Name = L"textBox36";

this->textBox36->Size = System::Drawing::Size(111, 20);

this->textBox36->TabIndex = 34;

//

// label31

//

this->label31->AutoSize = true;

this->label31->Location = System::Drawing::Point(141, 127);

this->label31->Name = L"label31";

this->label31->Size = System::Drawing::Size(49, 13);

this->label31->TabIndex = 33;

this->label31->Text = L"Ширина:";

//

// textBox37

//

this->textBox37->Location = System::Drawing::Point(6, 193);

this->textBox37->Name = L"textBox37";

this->textBox37->Size = System::Drawing::Size(111, 20);

this->textBox37->TabIndex = 32;

//

// label32

//

this->label32->AutoSize = true;

this->label32->Location = System::Drawing::Point(7, 177);

this->label32->Name = L"label32";

this->label32->Size = System::Drawing::Size(48, 13);

this->label32->TabIndex = 31;

this->label32->Text = L"Выстоа:";

//

// textBox38

//

this->textBox38->Location = System::Drawing::Point(6, 143);

this->textBox38->Name = L"textBox38";

this->textBox38->Size = System::Drawing::Size(111, 20);

this->textBox38->TabIndex = 30;

//

// label33

//

this->label33->AutoSize = true;

this->label33->Location = System::Drawing::Point(7, 127);

this->label33->Name = L"label33";

this->label33->Size = System::Drawing::Size(43, 13);

this->label33->TabIndex = 29;

this->label33->Text = L"Длина:";

//

// textBox39

//

this->textBox39->Location = System::Drawing::Point(6, 95);

this->textBox39->Name = L"textBox39";

this->textBox39->Size = System::Drawing::Size(143, 20);

this->textBox39->TabIndex = 28;

//

// label34

//

this->label34->AutoSize = true;

this->label34->Location = System::Drawing::Point(7, 79);

this->label34->Name = L"label34";

this->label34->Size = System::Drawing::Size(142, 13);

this->label34->TabIndex = 27;

this->label34->Text = L"Максимальная мощность:";

//

// textBox40

//

this->textBox40->Location = System::Drawing::Point(6, 34);

this->textBox40->Name = L"textBox40";

this->textBox40->Size = System::Drawing::Size(111, 20);

this->textBox40->TabIndex = 26;

//

// label35

//

this->label35->AutoSize = true;

this->label35->Location = System::Drawing::Point(11, 17);

this->label35->Name = L"label35";

this->label35->Size = System::Drawing::Size(49, 13);

this->label35->TabIndex = 25;

this->label35->Text = L"Модель:";

//

// dataGridView5

//

this->dataGridView5->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView5->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(8) {

this->dataGridViewTextBoxColumn19,

this->dataGridViewTextBoxColumn20, this->dataGridViewTextBoxColumn21, this->dataGridViewTextBoxColumn22, this->dataGridViewTextBoxColumn23,

this->dataGridViewTextBoxColumn24, this->Column12, this->Column13

});

this->dataGridView5->Location = System::Drawing::Point(370, 66);

this->dataGridView5->Name = L"dataGridView5";

this->dataGridView5->Size = System::Drawing::Size(644, 353);

this->dataGridView5->TabIndex = 24;

//

// dataGridViewTextBoxColumn19

//

this->dataGridViewTextBoxColumn19->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn19->Name = L"dataGridViewTextBoxColumn19";

this->dataGridViewTextBoxColumn19->ReadOnly = true;

//

// dataGridViewTextBoxColumn20

//

this->dataGridViewTextBoxColumn20->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn20->Name = L"dataGridViewTextBoxColumn20";

this->dataGridViewTextBoxColumn20->ReadOnly = true;

//

// dataGridViewTextBoxColumn21

//

this->dataGridViewTextBoxColumn21->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn21->Name = L"dataGridViewTextBoxColumn21";

this->dataGridViewTextBoxColumn21->ReadOnly = true;

//

// dataGridViewTextBoxColumn22

//

this->dataGridViewTextBoxColumn22->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn22->Name = L"dataGridViewTextBoxColumn22";

this->dataGridViewTextBoxColumn22->ReadOnly = true;

//

// dataGridViewTextBoxColumn23

//

this->dataGridViewTextBoxColumn23->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn23->Name = L"dataGridViewTextBoxColumn23";

this->dataGridViewTextBoxColumn23->ReadOnly = true;

//

// dataGridViewTextBoxColumn24

//

this->dataGridViewTextBoxColumn24->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn24->Name = L"dataGridViewTextBoxColumn24";

this->dataGridViewTextBoxColumn24->ReadOnly = true;

//

// Column12

//

this->Column12->HeaderText = L"Топливо";

this->Column12->Name = L"Column12";

//

// Column13

//

this->Column13->HeaderText = L"Тип топлива";

this->Column13->Name = L"Column13";

//

// tabPage6

//

this->tabPage6->Controls->Add(this->textBox71);

this->tabPage6->Controls->Add(this->label63);

this->tabPage6->Controls->Add(this->textBox72);

this->tabPage6->Controls->Add(this->label64);

this->tabPage6->Controls->Add(this->button36);

this->tabPage6->Controls->Add(this->button37);

this->tabPage6->Controls->Add(this->button38);

this->tabPage6->Controls->Add(this->button39);

this->tabPage6->Controls->Add(this->button40);

this->tabPage6->Controls->Add(this->textBox41);

this->tabPage6->Controls->Add(this->button41);

this->tabPage6->Controls->Add(this->button42);

this->tabPage6->Controls->Add(this->textBox42);

this->tabPage6->Controls->Add(this->label36);

this->tabPage6->Controls->Add(this->textBox43);

this->tabPage6->Controls->Add(this->label37);

this->tabPage6->Controls->Add(this->textBox44);

this->tabPage6->Controls->Add(this->label38);

this->tabPage6->Controls->Add(this->textBox45);

this->tabPage6->Controls->Add(this->label39);

this->tabPage6->Controls->Add(this->textBox46);

this->tabPage6->Controls->Add(this->label40);

this->tabPage6->Controls->Add(this->textBox47);

this->tabPage6->Controls->Add(this->label41);

this->tabPage6->Controls->Add(this->textBox48);

this->tabPage6->Controls->Add(this->label42);

this->tabPage6->Controls->Add(this->dataGridView6);

this->tabPage6->Location = System::Drawing::Point(4, 22);

this->tabPage6->Name = L"tabPage6";

this->tabPage6->Size = System::Drawing::Size(1020, 426);

this->tabPage6->TabIndex = 5;

this->tabPage6->Text = L"Роторно-поршневой";

this->tabPage6->UseVisualStyleBackColor = true;

//

// textBox71

//

this->textBox71->Location = System::Drawing::Point(173, 34);

this->textBox71->Name = L"textBox71";

this->textBox71->Size = System::Drawing::Size(111, 20);

this->textBox71->TabIndex = 58;

//

// label63

//

this->label63->AutoSize = true;

this->label63->Location = System::Drawing::Point(178, 18);

this->label63->Name = L"label63";

this->label63->Size = System::Drawing::Size(115, 13);

this->label63->TabIndex = 57;

this->label63->Text = L"Удельная мощность:";

//

// textBox72

//

this->textBox72->Location = System::Drawing::Point(173, 95);

this->textBox72->Name = L"textBox72";

this->textBox72->Size = System::Drawing::Size(111, 20);

this->textBox72->TabIndex = 56;

//

// label64

//

this->label64->AutoSize = true;

this->label64->Location = System::Drawing::Point(178, 79);

this->label64->Name = L"label64";

this->label64->Size = System::Drawing::Size(53, 13);

this->label64->TabIndex = 55;

this->label64->Text = L"Топливо:";

//

// button36

//

this->button36->Location = System::Drawing::Point(140, 327);

this->button36->Name = L"button36";

this->button36->Size = System::Drawing::Size(145, 23);

this->button36->TabIndex = 46;

this->button36->Text = L"Установить ошибку";

this->button36->UseVisualStyleBackColor = true;

this->button36->Click += gcnew System::EventHandler(this, &MyForm::button36\_Click);

//

// button37

//

this->button37->Location = System::Drawing::Point(596, 37);

this->button37->Name = L"button37";

this->button37->Size = System::Drawing::Size(139, 23);

this->button37->TabIndex = 45;

this->button37->Text = L"Загрузить из файла";

this->button37->UseVisualStyleBackColor = true;

this->button37->Click += gcnew System::EventHandler(this, &MyForm::button37\_Click);

//

// button38

//

this->button38->Location = System::Drawing::Point(596, 8);

this->button38->Name = L"button38";

this->button38->Size = System::Drawing::Size(139, 23);

this->button38->TabIndex = 44;

this->button38->Text = L"Записать в файл";

this->button38->UseVisualStyleBackColor = true;

this->button38->Click += gcnew System::EventHandler(this, &MyForm::button38\_Click);

//

// button39

//

this->button39->Location = System::Drawing::Point(750, 37);

this->button39->Name = L"button39";

this->button39->Size = System::Drawing::Size(264, 23);

this->button39->TabIndex = 43;

this->button39->Text = L"Получить справку";

this->button39->UseVisualStyleBackColor = true;

this->button39->Click += gcnew System::EventHandler(this, &MyForm::button39\_Click);

//

// button40

//

this->button40->Location = System::Drawing::Point(886, 8);

this->button40->Name = L"button40";

this->button40->Size = System::Drawing::Size(128, 23);

this->button40->TabIndex = 42;

this->button40->Text = L"Поиск";

this->button40->UseVisualStyleBackColor = true;

this->button40->Click += gcnew System::EventHandler(this, &MyForm::button40\_Click);

//

// textBox41

//

this->textBox41->Location = System::Drawing::Point(750, 10);

this->textBox41->Name = L"textBox41";

this->textBox41->Size = System::Drawing::Size(130, 20);

this->textBox41->TabIndex = 41;

//

// button41

//

this->button41->Location = System::Drawing::Point(6, 239);

this->button41->Name = L"button41";

this->button41->Size = System::Drawing::Size(227, 23);

this->button41->TabIndex = 40;

this->button41->Text = L"Добавить";

this->button41->UseVisualStyleBackColor = true;

this->button41->Click += gcnew System::EventHandler(this, &MyForm::button41\_Click);

//

// button42

//

this->button42->Location = System::Drawing::Point(10, 327);

this->button42->Name = L"button42";

this->button42->Size = System::Drawing::Size(111, 23);

this->button42->TabIndex = 39;

this->button42->Text = L"Установить";

this->button42->UseVisualStyleBackColor = true;

this->button42->Click += gcnew System::EventHandler(this, &MyForm::button42\_Click);

//

// textBox42

//

this->textBox42->Location = System::Drawing::Point(10, 300);

this->textBox42->Name = L"textBox42";

this->textBox42->Size = System::Drawing::Size(111, 20);

this->textBox42->TabIndex = 38;

//

// label36

//

this->label36->AutoSize = true;

this->label36->Location = System::Drawing::Point(11, 284);

this->label36->Name = L"label36";

this->label36->Size = System::Drawing::Size(110, 13);

this->label36->TabIndex = 37;

this->label36->Text = L"Текущая мощность:";

//

// textBox43

//

this->textBox43->Location = System::Drawing::Point(140, 193);

this->textBox43->Name = L"textBox43";

this->textBox43->Size = System::Drawing::Size(111, 20);

this->textBox43->TabIndex = 36;

//

// label37

//

this->label37->AutoSize = true;

this->label37->Location = System::Drawing::Point(141, 177);

this->label37->Name = L"label37";

this->label37->Size = System::Drawing::Size(29, 13);

this->label37->TabIndex = 35;

this->label37->Text = L"Вес:";

//

// textBox44

//

this->textBox44->Location = System::Drawing::Point(140, 143);

this->textBox44->Name = L"textBox44";

this->textBox44->Size = System::Drawing::Size(111, 20);

this->textBox44->TabIndex = 34;

//

// label38

//

this->label38->AutoSize = true;

this->label38->Location = System::Drawing::Point(141, 127);

this->label38->Name = L"label38";

this->label38->Size = System::Drawing::Size(49, 13);

this->label38->TabIndex = 33;

this->label38->Text = L"Ширина:";

//

// textBox45

//

this->textBox45->Location = System::Drawing::Point(6, 193);

this->textBox45->Name = L"textBox45";

this->textBox45->Size = System::Drawing::Size(111, 20);

this->textBox45->TabIndex = 32;

//

// label39

//

this->label39->AutoSize = true;

this->label39->Location = System::Drawing::Point(7, 177);

this->label39->Name = L"label39";

this->label39->Size = System::Drawing::Size(48, 13);

this->label39->TabIndex = 31;

this->label39->Text = L"Выстоа:";

//

// textBox46

//

this->textBox46->Location = System::Drawing::Point(6, 143);

this->textBox46->Name = L"textBox46";

this->textBox46->Size = System::Drawing::Size(111, 20);

this->textBox46->TabIndex = 30;

//

// label40

//

this->label40->AutoSize = true;

this->label40->Location = System::Drawing::Point(7, 127);

this->label40->Name = L"label40";

this->label40->Size = System::Drawing::Size(43, 13);

this->label40->TabIndex = 29;

this->label40->Text = L"Длина:";

//

// textBox47

//

this->textBox47->Location = System::Drawing::Point(6, 95);

this->textBox47->Name = L"textBox47";

this->textBox47->Size = System::Drawing::Size(143, 20);

this->textBox47->TabIndex = 28;

//

// label41

//

this->label41->AutoSize = true;

this->label41->Location = System::Drawing::Point(7, 79);

this->label41->Name = L"label41";

this->label41->Size = System::Drawing::Size(142, 13);

this->label41->TabIndex = 27;

this->label41->Text = L"Максимальная мощность:";

//

// textBox48

//

this->textBox48->Location = System::Drawing::Point(6, 34);

this->textBox48->Name = L"textBox48";

this->textBox48->Size = System::Drawing::Size(111, 20);

this->textBox48->TabIndex = 26;

//

// label42

//

this->label42->AutoSize = true;

this->label42->Location = System::Drawing::Point(11, 17);

this->label42->Name = L"label42";

this->label42->Size = System::Drawing::Size(49, 13);

this->label42->TabIndex = 25;

this->label42->Text = L"Модель:";

//

// dataGridView6

//

this->dataGridView6->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView6->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(8) {

this->dataGridViewTextBoxColumn25,

this->dataGridViewTextBoxColumn26, this->dataGridViewTextBoxColumn27, this->dataGridViewTextBoxColumn28, this->dataGridViewTextBoxColumn29,

this->dataGridViewTextBoxColumn30, this->Column10, this->Column11

});

this->dataGridView6->Location = System::Drawing::Point(370, 66);

this->dataGridView6->Name = L"dataGridView6";

this->dataGridView6->Size = System::Drawing::Size(644, 353);

this->dataGridView6->TabIndex = 24;

//

// dataGridViewTextBoxColumn25

//

this->dataGridViewTextBoxColumn25->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn25->Name = L"dataGridViewTextBoxColumn25";

this->dataGridViewTextBoxColumn25->ReadOnly = true;

//

// dataGridViewTextBoxColumn26

//

this->dataGridViewTextBoxColumn26->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn26->Name = L"dataGridViewTextBoxColumn26";

this->dataGridViewTextBoxColumn26->ReadOnly = true;

//

// dataGridViewTextBoxColumn27

//

this->dataGridViewTextBoxColumn27->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn27->Name = L"dataGridViewTextBoxColumn27";

this->dataGridViewTextBoxColumn27->ReadOnly = true;

//

// dataGridViewTextBoxColumn28

//

this->dataGridViewTextBoxColumn28->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn28->Name = L"dataGridViewTextBoxColumn28";

this->dataGridViewTextBoxColumn28->ReadOnly = true;

//

// dataGridViewTextBoxColumn29

//

this->dataGridViewTextBoxColumn29->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn29->Name = L"dataGridViewTextBoxColumn29";

this->dataGridViewTextBoxColumn29->ReadOnly = true;

//

// dataGridViewTextBoxColumn30

//

this->dataGridViewTextBoxColumn30->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn30->Name = L"dataGridViewTextBoxColumn30";

this->dataGridViewTextBoxColumn30->ReadOnly = true;

//

// Column10

//

this->Column10->HeaderText = L"Топливо";

this->Column10->Name = L"Column10";

//

// Column11

//

this->Column11->HeaderText = L"Удельная мощность";

this->Column11->Name = L"Column11";

//

// tabPage7

//

this->tabPage7->Controls->Add(this->textBox75);

this->tabPage7->Controls->Add(this->label67);

this->tabPage7->Controls->Add(this->textBox74);

this->tabPage7->Controls->Add(this->label66);

this->tabPage7->Controls->Add(this->textBox73);

this->tabPage7->Controls->Add(this->label65);

this->tabPage7->Controls->Add(this->button43);

this->tabPage7->Controls->Add(this->button44);

this->tabPage7->Controls->Add(this->button45);

this->tabPage7->Controls->Add(this->button46);

this->tabPage7->Controls->Add(this->button47);

this->tabPage7->Controls->Add(this->textBox49);

this->tabPage7->Controls->Add(this->button48);

this->tabPage7->Controls->Add(this->button49);

this->tabPage7->Controls->Add(this->textBox50);

this->tabPage7->Controls->Add(this->label43);

this->tabPage7->Controls->Add(this->textBox51);

this->tabPage7->Controls->Add(this->label44);

this->tabPage7->Controls->Add(this->textBox52);

this->tabPage7->Controls->Add(this->label45);

this->tabPage7->Controls->Add(this->textBox53);

this->tabPage7->Controls->Add(this->label46);

this->tabPage7->Controls->Add(this->textBox54);

this->tabPage7->Controls->Add(this->label47);

this->tabPage7->Controls->Add(this->textBox55);

this->tabPage7->Controls->Add(this->label48);

this->tabPage7->Controls->Add(this->textBox56);

this->tabPage7->Controls->Add(this->label49);

this->tabPage7->Controls->Add(this->dataGridView7);

this->tabPage7->Location = System::Drawing::Point(4, 22);

this->tabPage7->Name = L"tabPage7";

this->tabPage7->Size = System::Drawing::Size(1020, 426);

this->tabPage7->TabIndex = 6;

this->tabPage7->Text = L"Паро-турбинные";

this->tabPage7->UseVisualStyleBackColor = true;

//

// textBox75

//

this->textBox75->Location = System::Drawing::Point(170, 96);

this->textBox75->Name = L"textBox75";

this->textBox75->Size = System::Drawing::Size(63, 20);

this->textBox75->TabIndex = 54;

//

// label67

//

this->label67->AutoSize = true;

this->label67->Location = System::Drawing::Point(145, 79);

this->label67->Name = L"label67";

this->label67->Size = System::Drawing::Size(121, 13);

this->label67->TabIndex = 53;

this->label67->Text = L"Температура пара вх.:";

//

// textBox74

//

this->textBox74->Location = System::Drawing::Point(270, 96);

this->textBox74->Name = L"textBox74";

this->textBox74->Size = System::Drawing::Size(60, 20);

this->textBox74->TabIndex = 52;

//

// label66

//

this->label66->AutoSize = true;

this->label66->Location = System::Drawing::Point(272, 79);

this->label66->Name = L"label66";

this->label66->Size = System::Drawing::Size(35, 13);

this->label66->TabIndex = 51;

this->label66->Text = L"вых.::";

//

// textBox73

//

this->textBox73->Location = System::Drawing::Point(148, 35);

this->textBox73->Name = L"textBox73";

this->textBox73->Size = System::Drawing::Size(111, 20);

this->textBox73->TabIndex = 50;

//

// label65

//

this->label65->AutoSize = true;

this->label65->Location = System::Drawing::Point(153, 18);

this->label65->Name = L"label65";

this->label65->Size = System::Drawing::Size(58, 13);

this->label65->TabIndex = 49;

this->label65->Text = L"Источник:";

//

// button43

//

this->button43->Location = System::Drawing::Point(140, 327);

this->button43->Name = L"button43";

this->button43->Size = System::Drawing::Size(145, 23);

this->button43->TabIndex = 46;

this->button43->Text = L"Установить ошибку";

this->button43->UseVisualStyleBackColor = true;

this->button43->Click += gcnew System::EventHandler(this, &MyForm::button43\_Click);

//

// button44

//

this->button44->Location = System::Drawing::Point(596, 37);

this->button44->Name = L"button44";

this->button44->Size = System::Drawing::Size(139, 23);

this->button44->TabIndex = 45;

this->button44->Text = L"Загрузить из файла";

this->button44->UseVisualStyleBackColor = true;

this->button44->Click += gcnew System::EventHandler(this, &MyForm::button44\_Click);

//

// button45

//

this->button45->Location = System::Drawing::Point(596, 8);

this->button45->Name = L"button45";

this->button45->Size = System::Drawing::Size(139, 23);

this->button45->TabIndex = 44;

this->button45->Text = L"Записать в файл";

this->button45->UseVisualStyleBackColor = true;

this->button45->Click += gcnew System::EventHandler(this, &MyForm::button45\_Click);

//

// button46

//

this->button46->Location = System::Drawing::Point(750, 37);

this->button46->Name = L"button46";

this->button46->Size = System::Drawing::Size(264, 23);

this->button46->TabIndex = 43;

this->button46->Text = L"Получить справку";

this->button46->UseVisualStyleBackColor = true;

this->button46->Click += gcnew System::EventHandler(this, &MyForm::button46\_Click);

//

// button47

//

this->button47->Location = System::Drawing::Point(886, 8);

this->button47->Name = L"button47";

this->button47->Size = System::Drawing::Size(128, 23);

this->button47->TabIndex = 42;

this->button47->Text = L"Поиск";

this->button47->UseVisualStyleBackColor = true;

this->button47->Click += gcnew System::EventHandler(this, &MyForm::button47\_Click);

//

// textBox49

//

this->textBox49->Location = System::Drawing::Point(750, 10);

this->textBox49->Name = L"textBox49";

this->textBox49->Size = System::Drawing::Size(130, 20);

this->textBox49->TabIndex = 41;

//

// button48

//

this->button48->Location = System::Drawing::Point(6, 239);

this->button48->Name = L"button48";

this->button48->Size = System::Drawing::Size(227, 23);

this->button48->TabIndex = 40;

this->button48->Text = L"Добавить";

this->button48->UseVisualStyleBackColor = true;

this->button48->Click += gcnew System::EventHandler(this, &MyForm::button48\_Click);

//

// button49

//

this->button49->Location = System::Drawing::Point(10, 327);

this->button49->Name = L"button49";

this->button49->Size = System::Drawing::Size(111, 23);

this->button49->TabIndex = 39;

this->button49->Text = L"Установить";

this->button49->UseVisualStyleBackColor = true;

this->button49->Click += gcnew System::EventHandler(this, &MyForm::button49\_Click);

//

// textBox50

//

this->textBox50->Location = System::Drawing::Point(10, 300);

this->textBox50->Name = L"textBox50";

this->textBox50->Size = System::Drawing::Size(111, 20);

this->textBox50->TabIndex = 38;

//

// label43

//

this->label43->AutoSize = true;

this->label43->Location = System::Drawing::Point(11, 284);

this->label43->Name = L"label43";

this->label43->Size = System::Drawing::Size(110, 13);

this->label43->TabIndex = 37;

this->label43->Text = L"Текущая мощность:";

//

// textBox51

//

this->textBox51->Location = System::Drawing::Point(140, 193);

this->textBox51->Name = L"textBox51";

this->textBox51->Size = System::Drawing::Size(111, 20);

this->textBox51->TabIndex = 36;

//

// label44

//

this->label44->AutoSize = true;

this->label44->Location = System::Drawing::Point(141, 177);

this->label44->Name = L"label44";

this->label44->Size = System::Drawing::Size(29, 13);

this->label44->TabIndex = 35;

this->label44->Text = L"Вес:";

//

// textBox52

//

this->textBox52->Location = System::Drawing::Point(140, 143);

this->textBox52->Name = L"textBox52";

this->textBox52->Size = System::Drawing::Size(111, 20);

this->textBox52->TabIndex = 34;

//

// label45

//

this->label45->AutoSize = true;

this->label45->Location = System::Drawing::Point(141, 127);

this->label45->Name = L"label45";

this->label45->Size = System::Drawing::Size(49, 13);

this->label45->TabIndex = 33;

this->label45->Text = L"Ширина:";

//

// textBox53

//

this->textBox53->Location = System::Drawing::Point(6, 193);

this->textBox53->Name = L"textBox53";

this->textBox53->Size = System::Drawing::Size(111, 20);

this->textBox53->TabIndex = 32;

//

// label46

//

this->label46->AutoSize = true;

this->label46->Location = System::Drawing::Point(7, 177);

this->label46->Name = L"label46";

this->label46->Size = System::Drawing::Size(48, 13);

this->label46->TabIndex = 31;

this->label46->Text = L"Выстоа:";

//

// textBox54

//

this->textBox54->Location = System::Drawing::Point(6, 143);

this->textBox54->Name = L"textBox54";

this->textBox54->Size = System::Drawing::Size(111, 20);

this->textBox54->TabIndex = 30;

//

// label47

//

this->label47->AutoSize = true;

this->label47->Location = System::Drawing::Point(7, 127);

this->label47->Name = L"label47";

this->label47->Size = System::Drawing::Size(43, 13);

this->label47->TabIndex = 29;

this->label47->Text = L"Длина:";

//

// textBox55

//

this->textBox55->Location = System::Drawing::Point(6, 95);

this->textBox55->Name = L"textBox55";

this->textBox55->Size = System::Drawing::Size(143, 20);

this->textBox55->TabIndex = 28;

//

// label48

//

this->label48->AutoSize = true;

this->label48->Location = System::Drawing::Point(7, 79);

this->label48->Name = L"label48";

this->label48->Size = System::Drawing::Size(142, 13);

this->label48->TabIndex = 27;

this->label48->Text = L"Максимальная мощность:";

//

// textBox56

//

this->textBox56->Location = System::Drawing::Point(6, 34);

this->textBox56->Name = L"textBox56";

this->textBox56->Size = System::Drawing::Size(111, 20);

this->textBox56->TabIndex = 26;

//

// label49

//

this->label49->AutoSize = true;

this->label49->Location = System::Drawing::Point(11, 17);

this->label49->Name = L"label49";

this->label49->Size = System::Drawing::Size(49, 13);

this->label49->TabIndex = 25;

this->label49->Text = L"Модель:";

//

// dataGridView7

//

this->dataGridView7->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView7->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(9) {

this->dataGridViewTextBoxColumn31,

this->dataGridViewTextBoxColumn32, this->dataGridViewTextBoxColumn33, this->dataGridViewTextBoxColumn34, this->dataGridViewTextBoxColumn35,

this->dataGridViewTextBoxColumn36, this->Column14, this->Column15, this->Column16

});

this->dataGridView7->Location = System::Drawing::Point(370, 66);

this->dataGridView7->Name = L"dataGridView7";

this->dataGridView7->Size = System::Drawing::Size(644, 353);

this->dataGridView7->TabIndex = 24;

//

// dataGridViewTextBoxColumn31

//

this->dataGridViewTextBoxColumn31->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn31->Name = L"dataGridViewTextBoxColumn31";

this->dataGridViewTextBoxColumn31->ReadOnly = true;

//

// dataGridViewTextBoxColumn32

//

this->dataGridViewTextBoxColumn32->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn32->Name = L"dataGridViewTextBoxColumn32";

this->dataGridViewTextBoxColumn32->ReadOnly = true;

//

// dataGridViewTextBoxColumn33

//

this->dataGridViewTextBoxColumn33->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn33->Name = L"dataGridViewTextBoxColumn33";

this->dataGridViewTextBoxColumn33->ReadOnly = true;

//

// dataGridViewTextBoxColumn34

//

this->dataGridViewTextBoxColumn34->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn34->Name = L"dataGridViewTextBoxColumn34";

this->dataGridViewTextBoxColumn34->ReadOnly = true;

//

// dataGridViewTextBoxColumn35

//

this->dataGridViewTextBoxColumn35->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn35->Name = L"dataGridViewTextBoxColumn35";

this->dataGridViewTextBoxColumn35->ReadOnly = true;

//

// dataGridViewTextBoxColumn36

//

this->dataGridViewTextBoxColumn36->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn36->Name = L"dataGridViewTextBoxColumn36";

this->dataGridViewTextBoxColumn36->ReadOnly = true;

//

// Column14

//

this->Column14->HeaderText = L"Источник";

this->Column14->Name = L"Column14";

//

// Column15

//

this->Column15->HeaderText = L"Темп. пара. вх.";

this->Column15->Name = L"Column15";

//

// Column16

//

this->Column16->HeaderText = L"Темп. пара. вых.";

this->Column16->Name = L"Column16";

//

// tabPage8

//

this->tabPage8->Controls->Add(this->textBox77);

this->tabPage8->Controls->Add(this->label69);

this->tabPage8->Controls->Add(this->textBox76);

this->tabPage8->Controls->Add(this->label68);

this->tabPage8->Controls->Add(this->button50);

this->tabPage8->Controls->Add(this->button51);

this->tabPage8->Controls->Add(this->button52);

this->tabPage8->Controls->Add(this->button53);

this->tabPage8->Controls->Add(this->button54);

this->tabPage8->Controls->Add(this->textBox57);

this->tabPage8->Controls->Add(this->button55);

this->tabPage8->Controls->Add(this->button56);

this->tabPage8->Controls->Add(this->textBox58);

this->tabPage8->Controls->Add(this->label50);

this->tabPage8->Controls->Add(this->textBox59);

this->tabPage8->Controls->Add(this->label51);

this->tabPage8->Controls->Add(this->textBox60);

this->tabPage8->Controls->Add(this->label52);

this->tabPage8->Controls->Add(this->textBox61);

this->tabPage8->Controls->Add(this->label53);

this->tabPage8->Controls->Add(this->textBox62);

this->tabPage8->Controls->Add(this->label54);

this->tabPage8->Controls->Add(this->textBox63);

this->tabPage8->Controls->Add(this->label55);

this->tabPage8->Controls->Add(this->textBox64);

this->tabPage8->Controls->Add(this->label56);

this->tabPage8->Controls->Add(this->dataGridView8);

this->tabPage8->Location = System::Drawing::Point(4, 22);

this->tabPage8->Name = L"tabPage8";

this->tabPage8->Size = System::Drawing::Size(1020, 426);

this->tabPage8->TabIndex = 7;

this->tabPage8->Text = L"Электрическая";

this->tabPage8->UseVisualStyleBackColor = true;

//

// textBox77

//

this->textBox77->Location = System::Drawing::Point(191, 95);

this->textBox77->Name = L"textBox77";

this->textBox77->Size = System::Drawing::Size(111, 20);

this->textBox77->TabIndex = 52;

//

// label69

//

this->label69->AutoSize = true;

this->label69->Location = System::Drawing::Point(155, 79);

this->label69->Name = L"label69";

this->label69->Size = System::Drawing::Size(134, 13);

this->label69->TabIndex = 51;

this->label69->Text = L"Использование энергии:";

//

// textBox76

//

this->textBox76->Location = System::Drawing::Point(136, 35);

this->textBox76->Name = L"textBox76";

this->textBox76->Size = System::Drawing::Size(111, 20);

this->textBox76->TabIndex = 50;

//

// label68

//

this->label68->AutoSize = true;

this->label68->Location = System::Drawing::Point(141, 18);

this->label68->Name = L"label68";

this->label68->Size = System::Drawing::Size(58, 13);

this->label68->TabIndex = 49;

this->label68->Text = L"Источник:";

//

// button50

//

this->button50->Location = System::Drawing::Point(140, 327);

this->button50->Name = L"button50";

this->button50->Size = System::Drawing::Size(145, 23);

this->button50->TabIndex = 46;

this->button50->Text = L"Установить ошибку";

this->button50->UseVisualStyleBackColor = true;

this->button50->Click += gcnew System::EventHandler(this, &MyForm::button50\_Click);

//

// button51

//

this->button51->Location = System::Drawing::Point(596, 37);

this->button51->Name = L"button51";

this->button51->Size = System::Drawing::Size(139, 23);

this->button51->TabIndex = 45;

this->button51->Text = L"Загрузить из файла";

this->button51->UseVisualStyleBackColor = true;

this->button51->Click += gcnew System::EventHandler(this, &MyForm::button51\_Click);

//

// button52

//

this->button52->Location = System::Drawing::Point(596, 8);

this->button52->Name = L"button52";

this->button52->Size = System::Drawing::Size(139, 23);

this->button52->TabIndex = 44;

this->button52->Text = L"Записать в файл";

this->button52->UseVisualStyleBackColor = true;

this->button52->Click += gcnew System::EventHandler(this, &MyForm::button52\_Click);

//

// button53

//

this->button53->Location = System::Drawing::Point(750, 37);

this->button53->Name = L"button53";

this->button53->Size = System::Drawing::Size(264, 23);

this->button53->TabIndex = 43;

this->button53->Text = L"Получить справку";

this->button53->UseVisualStyleBackColor = true;

this->button53->Click += gcnew System::EventHandler(this, &MyForm::button53\_Click);

//

// button54

//

this->button54->Location = System::Drawing::Point(886, 8);

this->button54->Name = L"button54";

this->button54->Size = System::Drawing::Size(128, 23);

this->button54->TabIndex = 42;

this->button54->Text = L"Поиск";

this->button54->UseVisualStyleBackColor = true;

this->button54->Click += gcnew System::EventHandler(this, &MyForm::button54\_Click);

//

// textBox57

//

this->textBox57->Location = System::Drawing::Point(750, 10);

this->textBox57->Name = L"textBox57";

this->textBox57->Size = System::Drawing::Size(130, 20);

this->textBox57->TabIndex = 41;

//

// button55

//

this->button55->Location = System::Drawing::Point(6, 239);

this->button55->Name = L"button55";

this->button55->Size = System::Drawing::Size(227, 23);

this->button55->TabIndex = 40;

this->button55->Text = L"Добавить";

this->button55->UseVisualStyleBackColor = true;

this->button55->Click += gcnew System::EventHandler(this, &MyForm::button55\_Click);

//

// button56

//

this->button56->Location = System::Drawing::Point(10, 327);

this->button56->Name = L"button56";

this->button56->Size = System::Drawing::Size(111, 23);

this->button56->TabIndex = 39;

this->button56->Text = L"Установить";

this->button56->UseVisualStyleBackColor = true;

this->button56->Click += gcnew System::EventHandler(this, &MyForm::button56\_Click);

//

// textBox58

//

this->textBox58->Location = System::Drawing::Point(10, 300);

this->textBox58->Name = L"textBox58";

this->textBox58->Size = System::Drawing::Size(111, 20);

this->textBox58->TabIndex = 38;

//

// label50

//

this->label50->AutoSize = true;

this->label50->Location = System::Drawing::Point(11, 284);

this->label50->Name = L"label50";

this->label50->Size = System::Drawing::Size(110, 13);

this->label50->TabIndex = 37;

this->label50->Text = L"Текущая мощность:";

//

// textBox59

//

this->textBox59->Location = System::Drawing::Point(140, 193);

this->textBox59->Name = L"textBox59";

this->textBox59->Size = System::Drawing::Size(111, 20);

this->textBox59->TabIndex = 36;

//

// label51

//

this->label51->AutoSize = true;

this->label51->Location = System::Drawing::Point(141, 177);

this->label51->Name = L"label51";

this->label51->Size = System::Drawing::Size(29, 13);

this->label51->TabIndex = 35;

this->label51->Text = L"Вес:";

//

// textBox60

//

this->textBox60->Location = System::Drawing::Point(140, 143);

this->textBox60->Name = L"textBox60";

this->textBox60->Size = System::Drawing::Size(111, 20);

this->textBox60->TabIndex = 34;

//

// label52

//

this->label52->AutoSize = true;

this->label52->Location = System::Drawing::Point(141, 127);

this->label52->Name = L"label52";

this->label52->Size = System::Drawing::Size(49, 13);

this->label52->TabIndex = 33;

this->label52->Text = L"Ширина:";

//

// textBox61

//

this->textBox61->Location = System::Drawing::Point(6, 193);

this->textBox61->Name = L"textBox61";

this->textBox61->Size = System::Drawing::Size(111, 20);

this->textBox61->TabIndex = 32;

//

// label53

//

this->label53->AutoSize = true;

this->label53->Location = System::Drawing::Point(7, 177);

this->label53->Name = L"label53";

this->label53->Size = System::Drawing::Size(48, 13);

this->label53->TabIndex = 31;

this->label53->Text = L"Выстоа:";

//

// textBox62

//

this->textBox62->Location = System::Drawing::Point(6, 143);

this->textBox62->Name = L"textBox62";

this->textBox62->Size = System::Drawing::Size(111, 20);

this->textBox62->TabIndex = 30;

//

// label54

//

this->label54->AutoSize = true;

this->label54->Location = System::Drawing::Point(7, 127);

this->label54->Name = L"label54";

this->label54->Size = System::Drawing::Size(43, 13);

this->label54->TabIndex = 29;

this->label54->Text = L"Длина:";

//

// textBox63

//

this->textBox63->Location = System::Drawing::Point(6, 95);

this->textBox63->Name = L"textBox63";

this->textBox63->Size = System::Drawing::Size(143, 20);

this->textBox63->TabIndex = 28;

//

// label55

//

this->label55->AutoSize = true;

this->label55->Location = System::Drawing::Point(7, 79);

this->label55->Name = L"label55";

this->label55->Size = System::Drawing::Size(142, 13);

this->label55->TabIndex = 27;

this->label55->Text = L"Максимальная мощность:";

//

// textBox64

//

this->textBox64->Location = System::Drawing::Point(6, 34);

this->textBox64->Name = L"textBox64";

this->textBox64->Size = System::Drawing::Size(111, 20);

this->textBox64->TabIndex = 26;

//

// label56

//

this->label56->AutoSize = true;

this->label56->Location = System::Drawing::Point(11, 17);

this->label56->Name = L"label56";

this->label56->Size = System::Drawing::Size(49, 13);

this->label56->TabIndex = 25;

this->label56->Text = L"Модель:";

//

// dataGridView8

//

this->dataGridView8->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView8->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(8) {

this->dataGridViewTextBoxColumn37,

this->dataGridViewTextBoxColumn38, this->dataGridViewTextBoxColumn39, this->dataGridViewTextBoxColumn40, this->dataGridViewTextBoxColumn41,

this->dataGridViewTextBoxColumn42, this->Column17, this->Column18

});

this->dataGridView8->Location = System::Drawing::Point(370, 66);

this->dataGridView8->Name = L"dataGridView8";

this->dataGridView8->Size = System::Drawing::Size(644, 353);

this->dataGridView8->TabIndex = 24;

//

// dataGridViewTextBoxColumn37

//

this->dataGridViewTextBoxColumn37->HeaderText = L"Модель";

this->dataGridViewTextBoxColumn37->Name = L"dataGridViewTextBoxColumn37";

this->dataGridViewTextBoxColumn37->ReadOnly = true;

//

// dataGridViewTextBoxColumn38

//

this->dataGridViewTextBoxColumn38->HeaderText = L"Макс.мощность";

this->dataGridViewTextBoxColumn38->Name = L"dataGridViewTextBoxColumn38";

this->dataGridViewTextBoxColumn38->ReadOnly = true;

//

// dataGridViewTextBoxColumn39

//

this->dataGridViewTextBoxColumn39->HeaderText = L"Текущая мощность";

this->dataGridViewTextBoxColumn39->Name = L"dataGridViewTextBoxColumn39";

this->dataGridViewTextBoxColumn39->ReadOnly = true;

//

// dataGridViewTextBoxColumn40

//

this->dataGridViewTextBoxColumn40->HeaderText = L"Обороты";

this->dataGridViewTextBoxColumn40->Name = L"dataGridViewTextBoxColumn40";

this->dataGridViewTextBoxColumn40->ReadOnly = true;

//

// dataGridViewTextBoxColumn41

//

this->dataGridViewTextBoxColumn41->HeaderText = L"Ошибки";

this->dataGridViewTextBoxColumn41->Name = L"dataGridViewTextBoxColumn41";

this->dataGridViewTextBoxColumn41->ReadOnly = true;

//

// dataGridViewTextBoxColumn42

//

this->dataGridViewTextBoxColumn42->HeaderText = L"Габбариты";

this->dataGridViewTextBoxColumn42->Name = L"dataGridViewTextBoxColumn42";

this->dataGridViewTextBoxColumn42->ReadOnly = true;

//

// Column17

//

this->Column17->HeaderText = L"Источник";

this->Column17->Name = L"Column17";

//

// Column18

//

this->Column18->HeaderText = L"Использовано энергии";

this->Column18->Name = L"Column18";

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1056, 476);

this->Controls->Add(this->tabControl1);

this->Name = L"MyForm";

this->Text = L"MyForm";

this->tabControl1->ResumeLayout(false);

this->tabPage1->ResumeLayout(false);

this->tabPage1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

this->tabPage2->ResumeLayout(false);

this->tabPage2->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView2))->EndInit();

this->tabPage3->ResumeLayout(false);

this->tabPage3->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView3))->EndInit();

this->tabPage4->ResumeLayout(false);

this->tabPage4->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView4))->EndInit();

this->tabPage5->ResumeLayout(false);

this->tabPage5->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView5))->EndInit();

this->tabPage6->ResumeLayout(false);

this->tabPage6->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView6))->EndInit();

this->tabPage7->ResumeLayout(false);

this->tabPage7->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView7))->EndInit();

this->tabPage8->ResumeLayout(false);

this->tabPage8->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView8))->EndInit();

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text == "" || textBox2->Text == "" || textBox3->Text == "" || textBox4->Text == "" || textBox5->Text == "" || textBox6->Text == "")

{

EnergyPlants oEP;

sEP.push(oEP);

PM.insert(% sEP[0], 0);

PM[0]->Print(dataGridView1, dataGridView1->Rows->Count - 1);

}

else

{

try {

EnergyPlants oEP(textBox1->Text, Convert::ToDouble(textBox2->Text), Convert::ToDouble(textBox3->Text), Convert::ToDouble(textBox5->Text), Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox6->Text));

sEP.push(oEP);

PM.insert(% sEP[0], 0);

PM[0]->Print(dataGridView1, dataGridView1->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

EnergyPlants::PowerMoment(sEP,Convert::ToDouble(textBox7->Text), dataGridView1);

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

EnergyPlants::WriteToFile(sEP);

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e) {

EnergyPlants::ExtractFromFile(sEP, dataGridView1);

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[0]->Search(dataGridView1, textBox8->Text);

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

EnergyPlants::Status(sEP, dataGridView1);

}

private: System::Void tabPage1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button4\_Click\_1(System::Object^ sender, System::EventArgs^ e) {

EnergyPlants::Errors(sEP, dataGridView1);

}

private: System::Void button13\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox11->Text == "" || textBox12->Text == "" || textBox13->Text == "" || textBox14->Text == "" || textBox15->Text == "" || textBox65->Text == "" || textBox16->Text == "")

{

InternalCombustion oIC;

sIC.push(oIC);

PM.insert(% sIC[0], 1);

PM[1]->Print(dataGridView2, dataGridView2->Rows->Count - 1);

}

else

{

try {

InternalCombustion oIC(textBox16->Text, Convert::ToDouble(textBox15->Text), Convert::ToDouble(textBox14->Text), Convert::ToDouble(textBox12->Text), Convert::ToDouble(textBox13->Text), Convert::ToDouble(textBox11->Text), textBox65->Text);

sIC.push(oIC);

PM.insert(% sIC[0], 1);

PM[1]->Print(dataGridView2, dataGridView2->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button12\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[1]->Search(dataGridView2, textBox9->Text);

}

private: System::Void button11\_Click(System::Object^ sender, System::EventArgs^ e) {

InternalCombustion::Status(sIC, dataGridView2);

}

private: System::Void button14\_Click(System::Object^ sender, System::EventArgs^ e) {

InternalCombustion::PowerMoment(sIC, Convert::ToDouble(textBox10->Text), dataGridView2);

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e) {

InternalCombustion::Errors(sIC, dataGridView2);

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e) {

InternalCombustion::WriteToFile(sIC);

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e) {

InternalCombustion::ExtractFromFile(sIC, dataGridView2);

}

private: System::Void button20\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox19->Text == "" || textBox20->Text == "" || textBox21->Text == "" || textBox22->Text == "" || textBox23->Text == "" || textBox24->Text == "" || textBox66->Text == "")

{

AlternativePlants oAP;

sAP.push(oAP);

PM.insert(% sAP[0], 2);

PM[2]->Print(dataGridView3, dataGridView3->Rows->Count - 1);

}

else

{try{

AlternativePlants oAP(textBox24->Text, Convert::ToDouble(textBox23->Text), Convert::ToDouble(textBox22->Text), Convert::ToDouble(textBox20->Text), Convert::ToDouble(textBox21->Text), Convert::ToDouble(textBox19->Text), textBox66->Text);

sAP.push(oAP);

PM.insert(% sAP[0], 2);

PM[2]->Print(dataGridView3, dataGridView3->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button19\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[2]->Search(dataGridView3, textBox17->Text);

}

private: System::Void button18\_Click(System::Object^ sender, System::EventArgs^ e) {

AlternativePlants::Status(sAP, dataGridView3);

}

private: System::Void button17\_Click(System::Object^ sender, System::EventArgs^ e) {

AlternativePlants::WriteToFile(sAP);

}

private: System::Void button16\_Click(System::Object^ sender, System::EventArgs^ e) {

AlternativePlants::ExtractFromFile(sAP, dataGridView3);

}

private: System::Void button21\_Click(System::Object^ sender, System::EventArgs^ e) {

AlternativePlants::PowerMoment(sAP, Convert::ToDouble(textBox18->Text), dataGridView3);

}

private: System::Void button15\_Click(System::Object^ sender, System::EventArgs^ e) {

AlternativePlants::Errors(sAP, dataGridView3);

}

private: System::Void button27\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox32->Text == "" || textBox27->Text == "" || textBox28->Text == "" || textBox29->Text == "" || textBox30->Text == "" || textBox31->Text == "" || textBox67->Text == "" || textBox68->Text == "")

{

GasTurbine oGT;

sGT.push(oGT);

PM.insert(% sGT[0], 3);

PM[3]->Print(dataGridView4, dataGridView4->Rows->Count - 1);

}

else

{try{

GasTurbine oGT(textBox32->Text, Convert::ToDouble(textBox31->Text), Convert::ToDouble(textBox30->Text), Convert::ToDouble(textBox28->Text), Convert::ToDouble(textBox29->Text), Convert::ToDouble(textBox27->Text), textBox67->Text, Convert::ToDouble(textBox68->Text));

sGT.push(oGT);

PM.insert(% sGT[0], 3);

PM[3]->Print(dataGridView4, dataGridView4->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button28\_Click(System::Object^ sender, System::EventArgs^ e) {

GasTurbine::PowerMoment(sGT, Convert::ToDouble(textBox26->Text), dataGridView4);

}

private: System::Void button26\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[3]->Search(dataGridView4, textBox25->Text);

}

private: System::Void button25\_Click(System::Object^ sender, System::EventArgs^ e) {

GasTurbine::Status(sGT, dataGridView4);

}

private: System::Void button24\_Click(System::Object^ sender, System::EventArgs^ e) {

GasTurbine::WriteToFile(sGT);

}

private: System::Void button23\_Click(System::Object^ sender, System::EventArgs^ e) {

GasTurbine::ExtractFromFile(sGT, dataGridView4);

}

private: System::Void button22\_Click(System::Object^ sender, System::EventArgs^ e) {

GasTurbine::Errors(sGT, dataGridView4);

}

private: System::Void button34\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox40->Text == "" || textBox39->Text == "" || textBox38->Text == "" || textBox37->Text == "" || textBox36->Text == "" || textBox35->Text == "" || textBox69->Text == "" || textBox70->Text == "")

{

DVS oD;

sDVS.push(oD);

PM.insert(% sDVS[0], 4);

PM[4]->Print(dataGridView5, dataGridView5->Rows->Count - 1);

}

else

{try{

DVS oD(textBox40->Text, Convert::ToDouble(textBox39->Text), Convert::ToDouble(textBox38->Text), Convert::ToDouble(textBox36->Text), Convert::ToDouble(textBox37->Text), Convert::ToDouble(textBox35->Text), textBox70->Text,textBox69->Text);

sDVS.push(oD);

PM.insert(% sDVS[0], 4);

PM[4]->Print(dataGridView5, dataGridView5->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button33\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[4]->Search(dataGridView5, textBox33->Text);

}

private: System::Void button31\_Click(System::Object^ sender, System::EventArgs^ e) {

DVS::WriteToFile(sDVS);

}

private: System::Void button30\_Click(System::Object^ sender, System::EventArgs^ e) {

DVS::ExtractFromFile(sDVS, dataGridView5);

}

private: System::Void button32\_Click(System::Object^ sender, System::EventArgs^ e) {

DVS::Status(sDVS, dataGridView5);

}

private: System::Void button35\_Click(System::Object^ sender, System::EventArgs^ e) {

DVS::PowerMoment(sDVS, Convert::ToDouble(textBox34->Text), dataGridView5);

}

private: System::Void button29\_Click(System::Object^ sender, System::EventArgs^ e) {

DVS::Errors(sDVS, dataGridView5);

}

private: System::Void button41\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox48->Text == "" || textBox47->Text == "" || textBox46->Text == "" || textBox45->Text == "" || textBox44->Text == "" || textBox43->Text == "" || textBox72->Text == "" || textBox71->Text == "")

{

Rotor oR;

sR.push(oR);

PM.insert(% sR[0], 5);

PM[5]->Print(dataGridView6, dataGridView6->Rows->Count - 1);

}

else

{

try {

Rotor oR(textBox48->Text, Convert::ToDouble(textBox47->Text), Convert::ToDouble(textBox46->Text), Convert::ToDouble(textBox44->Text), Convert::ToDouble(textBox45->Text), Convert::ToDouble(textBox43->Text), textBox72->Text, Convert::ToDouble(textBox71->Text));

sR.push(oR);

PM.insert(% sR[0], 5);

PM[5]->Print(dataGridView6, dataGridView6->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button40\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[5]->Search(dataGridView6, textBox41->Text);

}

private: System::Void button38\_Click(System::Object^ sender, System::EventArgs^ e) {

Rotor::WriteToFile(sR);

}

private: System::Void button37\_Click(System::Object^ sender, System::EventArgs^ e) {

Rotor::ExtractFromFile(sR, dataGridView6);

}

private: System::Void button39\_Click(System::Object^ sender, System::EventArgs^ e) {

Rotor::Status(sR, dataGridView6);

}

private: System::Void button42\_Click(System::Object^ sender, System::EventArgs^ e) {

Rotor::PowerMoment(sR, Convert::ToDouble(textBox42->Text), dataGridView6);

}

private: System::Void button36\_Click(System::Object^ sender, System::EventArgs^ e) {

Rotor::Errors(sR, dataGridView6);

}

private: System::Void button48\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox51->Text == "" || textBox52->Text == "" || textBox53->Text == "" || textBox54->Text == "" || textBox55->Text == "" || textBox56->Text == "" || textBox73->Text == "" || textBox74->Text == "" || textBox75->Text == "")

{

SteamTurbine oS;

sST.push(oS);

PM.insert(% sST[0], 6);

PM[6]->Print(dataGridView7, dataGridView7->Rows->Count - 1);

}

else

{ try{

SteamTurbine oS(textBox56->Text, Convert::ToDouble(textBox55->Text), Convert::ToDouble(textBox54->Text), Convert::ToDouble(textBox52->Text), Convert::ToDouble(textBox53->Text), Convert::ToDouble(textBox51->Text), textBox73->Text, Convert::ToDouble(textBox75->Text), Convert::ToDouble(textBox74->Text));

sST.push(oS);

PM.insert(% sST[0], 6);

PM[6]->Print(dataGridView7, dataGridView7->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button47\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[6]->Search(dataGridView7, textBox49->Text);

}

private: System::Void button45\_Click(System::Object^ sender, System::EventArgs^ e) {

SteamTurbine::WriteToFile(sST);

}

private: System::Void button44\_Click(System::Object^ sender, System::EventArgs^ e) {

SteamTurbine::ExtractFromFile(sST, dataGridView7);

}

private: System::Void button46\_Click(System::Object^ sender, System::EventArgs^ e) {

SteamTurbine::Status(sST, dataGridView7);

}

private: System::Void button49\_Click(System::Object^ sender, System::EventArgs^ e) {

SteamTurbine::PowerMoment(sST, Convert::ToDouble(textBox50->Text), dataGridView7);

}

private: System::Void button43\_Click(System::Object^ sender, System::EventArgs^ e) {

SteamTurbine::Errors(sST, dataGridView7);

}

private: System::Void button55\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox64->Text == "" || textBox63->Text == "" || textBox62->Text == "" || textBox61->Text == "" || textBox60->Text == "" || textBox59->Text == "" || textBox76->Text == "" || textBox77->Text == "")

{

ElectricPlants oE;

sElP.push(oE);

PM.insert(% sElP[0], 7);

PM[7]->Print(dataGridView8, dataGridView8->Rows->Count - 1);

}

else

{try{

ElectricPlants oE(textBox64->Text, Convert::ToDouble(textBox63->Text), Convert::ToDouble(textBox62->Text), Convert::ToDouble(textBox60->Text), Convert::ToDouble(textBox61->Text), Convert::ToDouble(textBox59->Text), textBox76->Text, Convert::ToDouble(textBox77->Text));

sElP.push(oE);

PM.insert(% sElP[0], 7);

PM[7]->Print(dataGridView8, dataGridView8->Rows->Count - 1);

}

catch (...) { MessageBox::Show("Введены неккоректные данные", "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information); }

}

}

private: System::Void button54\_Click(System::Object^ sender, System::EventArgs^ e) {

PM[7]->Search(dataGridView8, textBox57->Text);

}

private: System::Void button53\_Click(System::Object^ sender, System::EventArgs^ e) {

ElectricPlants::Status(sElP, dataGridView8);

}

private: System::Void button52\_Click(System::Object^ sender, System::EventArgs^ e) {

ElectricPlants::WriteToFile(sElP);

}

private: System::Void button51\_Click(System::Object^ sender, System::EventArgs^ e) {

ElectricPlants::ExtractFromFile(sElP, dataGridView8);

}

private: System::Void button56\_Click(System::Object^ sender, System::EventArgs^ e) {

ElectricPlants::PowerMoment(sElP, Convert::ToDouble(textBox58->Text), dataGridView8);

}

private: System::Void button50\_Click(System::Object^ sender, System::EventArgs^ e) {

ElectricPlants::Errors(sElP, dataGridView8);

}

};

}

#pragma once

ref class AlternativePlants : public EnergyPlants

{

public:

AlternativePlants();

AlternativePlants(String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy);

AlternativePlants(AlternativePlants% value);

~AlternativePlants();

AlternativePlants% operator =(const AlternativePlants% right);

static void WriteToFile(MyStack<AlternativePlants>% mySt); // запись в файл

static void ExtractFromFile(MyStack<AlternativePlants>% mySt, DataGridView^ dgv); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ SeasonEffective();

static void PowerMoment(MyStack<AlternativePlants>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<AlternativePlants>% mySt, DataGridView^ dgv);

static void Status(MyStack<AlternativePlants>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

protected:

String^ typeSourceEnergy;

};

AlternativePlants::AlternativePlants() :EnergyPlants()

{

typeSourceEnergy = "no Data";

}

AlternativePlants::AlternativePlants(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ typeSourceEnergyR) :EnergyPlants(modelR, maxPowerR, lengthR, widthR, heightR, weightR)

{

this->typeSourceEnergy = typeSourceEnergyR;

}

AlternativePlants::AlternativePlants(AlternativePlants% value) :EnergyPlants(value)

{

typeSourceEnergy = value.typeSourceEnergy;

}

AlternativePlants% AlternativePlants::operator =(const AlternativePlants% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

typeSourceEnergy = right.typeSourceEnergy;

return \*this;

}

AlternativePlants::~AlternativePlants()

{

delete[] typeSourceEnergy;

}

void AlternativePlants::PowerMoment(MyStack<AlternativePlants>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void AlternativePlants::Errors(MyStack<AlternativePlants>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void AlternativePlants::Status(MyStack<AlternativePlants>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности"+"\n"+ mySt[i].SeasonEffective(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ AlternativePlants::SeasonEffective() {

if(typeSourceEnergy=="вода")return "Наиболее эффективна в период половодья и теплые месяцы";

else if (typeSourceEnergy == "пар" || typeSourceEnergy == "электричество")return "Наиболее эффективна круглыйгод";

else if (typeSourceEnergy == "солнце")return "Наиболее эффективна в дневное время весене-летний период";

else return "Нет данных о сезонной эффективности (данные существуют для ветра, воды, пара, солнца, электричества";

}

void AlternativePlants::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = typeSourceEnergy;

}

void AlternativePlants::WriteToFile(MyStack<AlternativePlants>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

// model

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].typeSourceEnergy);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void AlternativePlants::ExtractFromFile(MyStack<AlternativePlants>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

AlternativePlants obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine());

obj.Print(dgv, dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void AlternativePlants::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class InternalCombustion : public EnergyPlants

{

public:

InternalCombustion();

InternalCombustion(String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel);

InternalCombustion(InternalCombustion% value);

~InternalCombustion();

InternalCombustion% operator =(const InternalCombustion% right);

static void WriteToFile(MyStack<InternalCombustion>% mySt); // запись в файл

static void ExtractFromFile(MyStack<InternalCombustion>% mySt, DataGridView^ a); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ CalculatedFuel();

static void PowerMoment(MyStack<InternalCombustion>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<InternalCombustion>% mySt, DataGridView^ dgv);

static void Status(MyStack<InternalCombustion>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

protected:

String^ fuel;

};

InternalCombustion::InternalCombustion():EnergyPlants()

{

fuel = "no Data";

}

InternalCombustion::InternalCombustion(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR,String^ fuelR):EnergyPlants(modelR,maxPowerR, lengthR,widthR, heightR,weightR)

{

this->fuel = fuelR;

}

InternalCombustion::InternalCombustion(InternalCombustion% value):EnergyPlants(value)

{

fuel = value.fuel;

}

InternalCombustion% InternalCombustion::operator =(const InternalCombustion% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

fuel = right.fuel;

return \*this;

}

InternalCombustion::~InternalCombustion()

{

delete[] fuel;

}

void InternalCombustion::PowerMoment(MyStack<InternalCombustion>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void InternalCombustion::Errors(MyStack<InternalCombustion>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void InternalCombustion::Status(MyStack<InternalCombustion>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности" + "\n" + mySt[i].CalculatedFuel(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ InternalCombustion::CalculatedFuel()

{

double rf = rotation \* maxPower / 5;

return "Расход топлива: " + rf + " л/ч";

}

void InternalCombustion::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = fuel;

}

void InternalCombustion::WriteToFile(MyStack<InternalCombustion>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].fuel);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void InternalCombustion::ExtractFromFile(MyStack<InternalCombustion>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

InternalCombustion obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine());

obj.Print(dgv, dgv->Rows->Count);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void InternalCombustion::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class GasTurbine : public InternalCombustion

{

public:

GasTurbine();

GasTurbine(String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, double oxyConcentration);

GasTurbine(GasTurbine% value);

~GasTurbine();

GasTurbine% operator =(const GasTurbine% right);

static void WriteToFile(MyStack<GasTurbine>% mySt); // запись в файл

static void ExtractFromFile(MyStack<GasTurbine>% mySt, DataGridView^ a); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ AnaliticsBurning();

static void PowerMoment(MyStack<GasTurbine>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<GasTurbine>% mySt, DataGridView^ dgv);

static void Status(MyStack<GasTurbine>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

private:

double oxyConcentration;

};

GasTurbine::GasTurbine() :InternalCombustion()

{

oxyConcentration = 0;

}

GasTurbine::GasTurbine(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ fuelR,double oxyConcentrationR):InternalCombustion(modelR,maxPowerR,lengthR,widthR,heightR,weightR,fuelR)

{

this->oxyConcentration = oxyConcentrationR;

}

GasTurbine::GasTurbine(GasTurbine% value) :InternalCombustion(value)

{

oxyConcentration = value.oxyConcentration;

}

GasTurbine% GasTurbine::operator =(const GasTurbine% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

fuel = right.fuel;

oxyConcentration = right.oxyConcentration;

return \*this;

}

GasTurbine::~GasTurbine()

{

delete[] fuel;

}

void GasTurbine::PowerMoment(MyStack<GasTurbine>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void GasTurbine::Errors(MyStack<GasTurbine>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void GasTurbine::Status(MyStack<GasTurbine>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности" + "\n" + mySt[i].AnaliticsBurning(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ GasTurbine::AnaliticsBurning()

{

if (oxyConcentration <= 35) return "Анализ горения: Бедная смесь";

if (oxyConcentration > 35 && oxyConcentration < 65) return "Анализ горения: Нормальная смесь";

if (oxyConcentration >= 65 ) return "Анализ горения: Богатая смесь";

}

void GasTurbine::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = fuel;

dgv->Rows[rows]->Cells[7]->Value = oxyConcentration;

}

void GasTurbine::WriteToFile(MyStack<GasTurbine>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].fuel);

sw->WriteLine(mySt[i].oxyConcentration);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void GasTurbine::ExtractFromFile(MyStack<GasTurbine>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

GasTurbine obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine(), Convert::ToDouble(sr->ReadLine()));

obj.Print(dgv, dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void GasTurbine::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class DVS : public InternalCombustion

{

public:

DVS();

DVS(String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, String^ typeFuel);

DVS(DVS% value);

~DVS();

DVS% operator =(const DVS% right);

static void WriteToFile(MyStack<DVS>% mySt); // запись в файл

static void ExtractFromFile(MyStack<DVS>% mySt, DataGridView^ dgv); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ TypeDVS();

static void PowerMoment(MyStack<DVS>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<DVS>% mySt, DataGridView^ dgv);

static void Status(MyStack<DVS>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

private:

String^ typeFuel;

};

DVS::DVS() :InternalCombustion()

{

typeFuel = "no Data";

}

DVS::DVS(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ fuelR, String^ typeFuelR) :InternalCombustion(modelR, maxPowerR, lengthR, widthR, heightR, weightR, fuelR)

{

this->typeFuel = typeFuelR;

}

DVS::DVS(DVS% value) :InternalCombustion(value)

{

typeFuel = value.typeFuel;

}

DVS% DVS::operator =(const DVS% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

fuel = right.fuel;

typeFuel = right.typeFuel;

return \*this;

}

DVS::~DVS()

{

delete[] typeFuel;

}

void DVS::PowerMoment(MyStack<DVS>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void DVS::Errors(MyStack<DVS>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void DVS::Status(MyStack<DVS>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности" "\n" + mySt[i].TypeDVS(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ DVS::TypeDVS()

{

if (fuel == "бензин") return "Бензиновый двигатель";

else if (fuel == "газ") return "Бензиновый двигатель с установленым ГБО";

else if (fuel == "ДТ") return "Дизельный двигатель";

else return "Тип неопределен";

}

void DVS::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = fuel;

dgv->Rows[rows]->Cells[7]->Value = typeFuel;

}

void DVS::WriteToFile(MyStack<DVS>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].fuel);

sw->WriteLine(mySt[i].typeFuel);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void DVS::ExtractFromFile(MyStack<DVS>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

DVS obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine(), sr->ReadLine());

obj.Print(dgv,dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void DVS::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class Rotor : public InternalCombustion

{

public:

Rotor();

Rotor(String^ model, double maxPower, double length, double width, double height, double weight, String^ fuel, double udPower);

Rotor(Rotor% value);

~Rotor();

Rotor% operator =(const Rotor% right);

static void WriteToFile(MyStack<Rotor>% mySt); // запись в файл

static void ExtractFromFile(MyStack<Rotor>% mySt, DataGridView^ dgv); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ SravnSDVS();

static void PowerMoment(MyStack<Rotor>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<Rotor>% mySt, DataGridView^ dgv);

static void Status(MyStack<Rotor>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv,int rows) override; // добавление объекта

private:

double udPower;

};

Rotor::Rotor() :InternalCombustion()

{

udPower = 0;

}

Rotor::Rotor(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ fuelR, double oxyConcentrationR) :InternalCombustion(modelR, maxPowerR, lengthR, widthR, heightR, weightR, fuelR)

{

this->udPower = oxyConcentrationR;

}

Rotor::Rotor(Rotor% value) :InternalCombustion(value)

{

udPower = value.udPower;

}

Rotor% Rotor::operator =(const Rotor% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

fuel = right.fuel;

udPower = right.udPower;

return \*this;

}

Rotor::~Rotor()

{

delete[] fuel;

}

void Rotor::PowerMoment(MyStack<Rotor>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void Rotor::Errors(MyStack<Rotor>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void Rotor::Status(MyStack<Rotor>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности"+"\n"+ mySt[i].SravnSDVS(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ Rotor::SravnSDVS() {

double r = udPower / maxPower \* 0.15 + maxPower;

return "Для достижения заданной мощности на ДВС потребуется мощность: " + r;

}

void Rotor::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = fuel;

dgv->Rows[rows]->Cells[7]->Value = udPower;

}

void Rotor::WriteToFile(MyStack<Rotor>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].fuel);

sw->WriteLine(mySt[i].udPower);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void Rotor::ExtractFromFile(MyStack<Rotor>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

Rotor obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine(), Convert::ToDouble(sr->ReadLine()));

obj.Print(dgv, dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void Rotor::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class SteamTurbine : public AlternativePlants

{

public:

SteamTurbine();

SteamTurbine(String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy,double tempStreamIn, double tempStreamOut);

SteamTurbine(SteamTurbine% value);

~SteamTurbine();

SteamTurbine% operator =(const SteamTurbine% right);

static void WriteToFile(MyStack<SteamTurbine>% mySt); // запись в файл

static void ExtractFromFile(MyStack<SteamTurbine>% mySt, DataGridView^ dgv); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override;// поиск объекта

String^ TempRash();

static void PowerMoment(MyStack<SteamTurbine>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<SteamTurbine>% mySt, DataGridView^ dgv);

static void Status(MyStack<SteamTurbine>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

private:

double tempStreamIn;

double tempStreamOut;

};

SteamTurbine::SteamTurbine() :AlternativePlants()

{

tempStreamIn = 0;

tempStreamOut = 0;

}

SteamTurbine::SteamTurbine(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ typeSourceEnergyR, double tempStreamInR, double tempStreamOutR):AlternativePlants(modelR, maxPowerR, lengthR, widthR, heightR, weightR, typeSourceEnergyR)

{

this->tempStreamIn = tempStreamInR;

this->tempStreamOut = tempStreamOutR;

}

SteamTurbine::SteamTurbine(SteamTurbine% value) :AlternativePlants(value)

{

tempStreamIn = value.tempStreamIn;

tempStreamOut = value.tempStreamOut;

}

SteamTurbine% SteamTurbine::operator =(const SteamTurbine% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

typeSourceEnergy = right.typeSourceEnergy;

tempStreamIn = right.tempStreamIn;

tempStreamOut = right.tempStreamOut;

return \*this;

}

SteamTurbine::~SteamTurbine()

{

delete[] typeSourceEnergy;

}

void SteamTurbine::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = typeSourceEnergy;

dgv->Rows[rows]->Cells[7]->Value = tempStreamIn;

dgv->Rows[rows]->Cells[8]->Value = tempStreamOut;

}

void SteamTurbine::PowerMoment(MyStack<SteamTurbine>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void SteamTurbine::Errors(MyStack<SteamTurbine>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void SteamTurbine::Status(MyStack<SteamTurbine>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности \n" + mySt[i].TempRash(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ SteamTurbine::TempRash() {

double d = tempStreamIn - tempStreamOut;

return "Разница температур: " + d;

}

void SteamTurbine::WriteToFile(MyStack<SteamTurbine>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

// model

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].typeSourceEnergy);

sw->WriteLine(mySt[i].tempStreamIn);

sw->WriteLine(mySt[i].tempStreamOut);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void SteamTurbine::ExtractFromFile(MyStack<SteamTurbine>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

SteamTurbine obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()));

obj.Print(dgv, dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void SteamTurbine::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}

#pragma once

ref class ElectricPlants : public AlternativePlants

{

public:

ElectricPlants();

ElectricPlants(String^ model, double maxPower, double length, double width, double height, double weight, String^ typeSourceEnergy, double usingEnergy);

ElectricPlants(ElectricPlants% value);

~ElectricPlants();

ElectricPlants% operator =(const ElectricPlants% right);

static void WriteToFile(MyStack<ElectricPlants>% mySt); // запись в файл

static void ExtractFromFile(MyStack<ElectricPlants>% mySt, DataGridView^ dgv); // чтение из файла

void Search(DataGridView^ dgv, String^ str) override; // поиск объекта

String^ ClassEnergySave();

static void PowerMoment(MyStack<ElectricPlants>% mySt, double pw, DataGridView^ dgv);

static void Errors(MyStack<ElectricPlants>% mySt, DataGridView^ dgv);

static void Status(MyStack<ElectricPlants>% mySt, DataGridView^ dgv);

void Print(DataGridView^ dgv, int rows) override; // добавление объекта

private:

double usingEnergy;

};

ElectricPlants::ElectricPlants() :AlternativePlants()

{

usingEnergy = 0;

}

ElectricPlants::ElectricPlants(String^ modelR, double maxPowerR, double lengthR, double widthR, double heightR, double weightR, String^ typeSourceEnergyR, double usingEnergyR) :AlternativePlants(modelR, maxPowerR, lengthR, widthR, heightR, weightR, typeSourceEnergyR)

{

this->usingEnergy = usingEnergyR;

}

ElectricPlants::ElectricPlants(ElectricPlants% value) :AlternativePlants(value)

{

usingEnergy = value.usingEnergy;

}

ElectricPlants% ElectricPlants::operator =(const ElectricPlants% right)

{

if (this == % right)

{

return \*this;

}

model = right.model;

length = right.length;

width = right.width;

height = right.height;

weight = right.weight;

maxPower = right.maxPower;

rotation = right.rotation;

er = right.er;

typeSourceEnergy = right.typeSourceEnergy;

usingEnergy = right.usingEnergy;

return \*this;

}

ElectricPlants::~ElectricPlants()

{

delete[] typeSourceEnergy;

}

void ElectricPlants::Print(DataGridView^ dgv, int rows)

{

dgv->RowCount += 1;

dgv->Rows[rows]->Cells[0]->Value = model;

dgv->Rows[rows]->Cells[1]->Value = maxPower;

dgv->Rows[rows]->Cells[4]->Value = er;

dgv->Rows[rows]->Cells[5]->Value = length + "\*" + width + "\*" + height + "," + weight;

dgv->Rows[rows]->Cells[6]->Value = typeSourceEnergy;

dgv->Rows[rows]->Cells[7]->Value = usingEnergy;

}

void ElectricPlants::PowerMoment(MyStack<ElectricPlants>% mySt, double pw, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

dgv->Rows[dgv->CurrentRow->Index]->Cells[2]->Value = Convert::ToString(pw);

mySt[i].powerMoment = pw;

dgv->Rows[dgv->CurrentRow->Index]->Cells[3]->Value = (pw \* 79) - 50;

mySt[i].rotation = (pw \* 79) - 50;

}

void ElectricPlants::Errors(MyStack<ElectricPlants>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

mySt[i].er = true;

if (mySt[i].er == true) {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Ошибка";

}

else {

dgv->Rows[dgv->CurrentRow->Index]->Cells[4]->Value = "Отсуствуют";

}

}

void ElectricPlants::Status(MyStack<ElectricPlants>% mySt, DataGridView^ dgv)

{

int i = (mySt.GetSize() - 1) - dgv->CurrentRow->Index;

String^ per = Convert::ToString(mySt[i].powerMoment / mySt[i].maxPower \* 100);

String^ rab = "Штатно";

if (mySt[i].er == true)rab = "Аварийно";

MessageBox::Show("Установка работает: " + rab + " на " + per + " % от максимальной мощности \n" + mySt[i].ClassEnergySave(), "Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

String^ ElectricPlants::ClassEnergySave() {

double d = usingEnergy / maxPower;

if (d <= 1)return "Высокая энергоэффективность";

else if (d >= 0.5 && d<1)return "Средняя энергоэффективность";

else if (d > 1)return "Низкая энергоэффективность";

else return "Нет данных об энергоэффективности";

}

void ElectricPlants::WriteToFile(MyStack<ElectricPlants>% mySt)

{

try

{

Stream^ save;

SaveFileDialog^ dialog = gcnew SaveFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

dialog->CreatePrompt = true;

dialog->OverwritePrompt = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

save = dialog->OpenFile();

if (save != nullptr)

{

StreamWriter^ sw = gcnew StreamWriter(save);

for (int i = 0; i < mySt.GetSize(); i++)

{

// model

sw->WriteLine(mySt[i].model);

sw->WriteLine(mySt[i].maxPower);

sw->WriteLine(mySt[i].length);

sw->WriteLine(mySt[i].width);

sw->WriteLine(mySt[i].height);

sw->WriteLine(mySt[i].weight);

sw->WriteLine(mySt[i].typeSourceEnergy);

sw->WriteLine(mySt[i].usingEnergy);

}

sw->Close();

}

save->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при сохранении данных в тектовый файл!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void ElectricPlants::ExtractFromFile(MyStack<ElectricPlants>% mySt, DataGridView^ dgv)

{

try

{

Stream^ Open;

OpenFileDialog^ dialog = gcnew OpenFileDialog();

dialog->Filter = "Text files | \*.txt";

dialog->FilterIndex = 1;

dialog->RestoreDirectory = true;

if (dialog->ShowDialog() == DialogResult::OK)

{

Open = dialog->OpenFile();

if (Open != nullptr)

{

StreamReader^ sr = gcnew StreamReader(Open);

while (!sr->EndOfStream)

{

ElectricPlants obj(sr->ReadLine(), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), Convert::ToDouble(sr->ReadLine()), sr->ReadLine(), Convert::ToDouble(sr->ReadLine()));

obj.Print(dgv, dgv->Rows->Count-1);

mySt.push(obj);

}

sr->Close();

}

Open->Close();

}

}

catch (Exception ^ e)

{

MessageBox::Show("Ошибка при чтении данных из текстового файла!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

void ElectricPlants::Search(DataGridView^ dgv, String^ str)

{

dgv->ClearSelection();

for (int i = 0; i < dgv->Rows->Count; i++)

{

if (str == Convert::ToString(dgv->Rows[i]->Cells[0]->Value))

{

dgv->Rows[i]->Selected = true;

}

}

}