Задание – курсов проект по ООП

*Storage Engine App*

*Приложение за съхраняване на данни и частична симулация работата на различни видове ел. двигатели*

Настолното приложение ще предоставя модерен графичен интерфейс за работа (Windows Forms). Интерфейсът ще бъде разделен визуално на три блока – информация за двигателя, операции за работа с приложението и визуализация на съхранените данни. Приложението ще работи с безплатна онлайн база данни и ще има възможност за съхраняване (еxport) на данните от базата данни в текстов файл. Представям накратко работата на всеки един от трите блока:

**Първият блок** ще представлява информация за избрания вид двигател като потребителят ще има възможност ръчно да въведе необходимите данни в поставените осем текстови кутии (textboxes) – ID (сериен 8-цифрен номер) на двигателя, мощност, напрежение, обороти/мин, ток и честота, цена – червена карта, цена – зелена карта. Цената на червената карта означава, че на двигателя е извършен пълен механичен ремонт, докато цената на зелената карта означава, че е извършен частичен механичен ремонт. Също така в този блок ще има обособено падащо меню, от което потребителят трябва да избере вида двигател – постоянно-токов, асинхронен или генератор.

**Вторият блок** ще съдържа основните CRUD операции – бутони (събития), а именно “Добави”, “Ъпдейт”, “Изтрий”, “Прегледай”, “Търси”. Бутони “Завърти по посока на часовниковата стрелка” и “Завърти обратно на часовниковата стрелка”, тези бутони ще симулират завъртането на посочения от потребителя двигател като за целта ще се визуализира промяната на поляритета на трите фази – “R-S-T” на случаен принцип (class Random). Бутон “Произведи ток”, който ще бъде видим, само ако посоченият двигател е генератор. Симулативно чрез този бутон ще се произвежда някакво напрежение в диапазона 12-400V (Random число), като за да се симулира работата на този генератор умишлено ще има забавяне на програмата посредством System.Threading. В този блок ще има етикет (label) “Общ брой” показващ общия брой добавени двигатели в съответната таблица от базата данни. Ще има и бутон “Еxport” за извличане на информацията от базата данни в текстов файл. Самата база данни ще съдържа три таблици с осем колони за всички данни въведени от потребителя. Също така ще има бутон за обновяване на информацията, който ще действа, само ако има някаква промяна в данните.

**Третият блок** ще визуализира съответната таблица от базата данни за текущия избран двигател (dataGridView). Потребителят ще има възможност да сортира посредством стрелки всяка една от колоните в таблицата. За да може да ъпдейтне данните от нея ще бъде необходимо с двоен клик да избере запис – ред от таблицата и посредством бутона “Ъпдейт” да промени информацията.

*Информация за работата по етапи:*

**Етап 1:**

1.       **Наследяване** – йерархия от класове за двигателите – постоянно-токов, асинхронен и генератор;

2.       **Капсулация** – всеки двигател притежава характеристики – мощност (capacity) kW, напрежение V, обороти-мин (2900-3000), ток (200-300A), честота (херцове – до 100 Hz); също така всеки двигател има ID (8-цифрен номер), цена (зелена карта – частичен механичен ремонт и червена карта – пълен ел. механичен ремонт);

3.       **Полиморфизъм** – двигателите се съхраняват в една колекция в базовия клас Engine; ще бъдат изградени интерфейси IRotateable – два метода RotateClockwize и RotateCounterClockwize; IProduceable – с метод ProduceEnergy; изброените интерфейси ще наследяват класовете двигатели, за които отговарят;

4.       **Виртуални методи** – асинхронните и постоянно-токовите двигатели може да се завъртят по посока и обратно на часовниковата стрелка – смяната на честотата води до завъртане и промяна на оборотите; генераторът може да се завърти само по посока на часовниковата стрелка и да произвежда електричество

5.       **Употреба на различни модификатори за достъп** – за характеристиките на двигателите, техните методите, при извличането на данните (Export) и т.н.;

6.       **Свойства** – мощност kW, напрежение V, обороти-мин, ток, честота и т.н.

7.       **Делегати и събития** – бутоните могат да се избират с мишката, като това поражда събитие OnClick; при избор на запис от таблицата за ъпдейт се поражда събитие MouseDoubleClick; събитие KeyPress за постигане на рестрикция при въвеждане данните в текстовите кутии (допустими са само цифри)

**Етап 2:**

1.       **Интерфейс за работа с програмата** – Windows Forms интерфейс с три форми: **groupbox1**- информация за двигателя – осем текстови кутии за въвеждане на информацията и падащ списък за избор на двигател; **groupbox2** – форма, съдържаща бутоните за функционалност, бутон за презареждане на информацията и label за общия брой двигатели, бутох за извличане (export) на данните в текстов файл; **groupbox3** за визуализация на таблицата от базата данни;

2.       **Употреба на специализирана програмна функционалност взаимодействаща с изградената в етап едно йерархия** – Използване на функционалността на **System.Threading** за умишлено забавяне работата на програмата и симулиране на произвеждане на ток от генератора. **MySql.Data.MySqlClient** за работа и изпращане на заявки към базата данни. **System.Data** за работа със самата база данни. **System.Windows.Forms** за основната функционалност на бутоните и визуализацията на съобщения към потребителя (MessageBox).

**Етап 3:**

1.       **Съхранение на информация във файлове, посредством сериализация** – цялата информацията от базата данни за избрания вид двигател може да бъде записана в текстов файл посредством бутона “Export” и след това този файл да бъде използван от потребителя за необходимите цели

2.       **Използване на LINQ** – при работа с колекцията двигатели – всеки един добавен двигател в базата данни ще се съхранява също и в лист (операции с листовете – сортиране по възходящ или низходящ ред посредством System.Linq, филтриране с Where)

3.       **Реализация на библиотека** – йерархията от всички видове двигатели ще бъде реализирана в библиотека, като всеки двигател (обект) ще бъде независим от **Windows Forms** класовете.