Wiktor Komorowski

PROE

sem. 1

mgr inż. Marcin Kamil Bączyk

**Aplikacja/Moduł do obsługi dystrybutorów na stacji benzynowej**

Zmiany w specyfikacji:

Zmiany będą zaznaczone na tak, a w nawiasach () będą wypisane nowe rozwiązania.

**2. Interface aplikacji**

- Uruchamiany z poziomu powłoki z pobraniem danych podczas działania programu(dla tankowania samochodu)

Wybór rodzaju paliwa

Wybór ilości paliwa poprzez naciśnięcie Spacji („r”) aby rozpocząć tankowanie i ponowne jej kliknięcie aby zakończyć

- Uruchamiany z poziomu powłoki z pobraniem danych podczas działania programu(dla dostawy paliwa)

Wpisanie na klawiaturze „-d” („s”) (nie znanego normalnemu użytkownikowi) spowoduje możliwość doładowania paliwa o wybraną wartość

**3. Oczekiwany wynik działania programu**

-Sposób prezentowania wyników

Każdorazowo wypisywana jest do pliku ilość zatankowanego paliwa oraz należność do zapłaty

Po zakończeniu pracy program będzie wypisywał na ekran ilość paliwa które zostało jeszcze w zbiornikach stacji

**4. Analiza obiektów występujących w trakcie działania programu**

-Action

Obiekt odpowiedzialny za wykonywanie akcji

(Zamieniony na MainSequence->\_do\_() )

-Fuel

Reprezentuje paliwo i zawiera metody takie jak tankowanie samochodu czy dostawę paliwa

-Disel

Obiekt zawiera informacje o cenie na dm3 i ilość dostępnego tego rodzaju paliwa

-Petrol\_95(Zamieniono na Petrol)

Obiekt zawiera informacje o cenie na dm3 i ilość dostępnego tego rodzaju paliwa

-Petrol\_98

Obiekt zawiera informacje o cenie na dm3 i ilość dostępnego tego rodzaju paliwa

(Usunięty)

-Customer

Obiekt przetrzymuje aktualne informacje o stanie tankowania oraz należności za nie

-billGenerator

Obiekt gnerujący paragony

-raportGenerator

Obiekt generyjący raport z działania programu(spis obsłużonych osób)

-inputAnalizer

Obiekt analizujący czy został użyty przełącznik „-d” („s”)oraz wybrany typ paliwa

Dodano:

-MainMenu

Wyświetla Menu Gółwne

-Distibutor

Opowiedzialna za proces tankowania

-Supply

Odpowiedzialna za przebieg dostaw

-SupplyMenu

Wyświetla Menu dostaw

-Gas

Obiekt zawiera informacje o cenie na dm3 i ilość dostępnego tego rodzaju paliwa

**5. Przypadki użycia**

Pokazanie menu->Analiza danych wejściowych->Przekierowanie ich do odpowiednich obiektów->Wyświetlenie działania oraz aktualizacja raportu->Pokazanie menu…

**6. Statyczna analiza typów i interfejsów poszczególnych typów**

-Action (MainSequence)

Zaprzyjaźniona z inputAnalizer (agregacja InputAnalizer w MainSequence)

-Przechowuje instrukcję switch akcji programu

-Fuel

-Abstakcyjna

-Klasa bazowa

-zawiera metody wirtualne

-Disel

-Zawiera pola

float petrolStationLevel-ilość dostępnego paliwa

cost(price) – cena za dm3

string - nazwa

-Korzysta z metod klasy Fuel

-Petrol\_95 (Petrol)

-Zawiera pola

float petrolStationLevel-ilość dostępnego paliwa

cost(price) – cena za dm3

string - nazwa

-Korzysta z metod klasy Fuel

-Petrol\_98

-Zawiera pola

float petrolStationLevel-ilość dostępnego paliwa

float cost(price) – cena za dm3

string - nazwa

-Korzysta z metod klasy Fuel (Usunięto)

-Gas

-Zawiera pola

float petrolStationLevel-ilość dostępnego paliwa

float cost(price) – cena za dm3

string - nazwa

-Korzysta z metod klasy Fuel

-Customer

-Zwiera pola

refuelLevel-ilość zatankowanego paliwa

supplyLevel-ilość dostarczonego paliwa (Usunięto)

-Zawiera metodę

Refuel

Supply(Jest oddzielną klasą)

-billGenerator

Jest zaprzyjaźniona z Customer, Disel, Petrol\_95, Petrol\_98(agregacja)

-Zawiera metody

calculateBill – liczy należność

printBill – drukuje paragon(usunięto)

makeBill – przekazuje paragon do obiektu klasy Customer

-raportGenerator

Zprzyjźniona z billGenerator(agregacja z Customer)

-Zawiera metody

updateRaport (makeReFuelingRaport)– dodaje do pliku kolejne paragony

makeRaport – tworzy plik z raportem i dodaje tam pierwszy paragon(usunięto)

openFileStream- otwiera strumie do pliku z raportem i ewentualnie go tworzy

closeFileStream- zamyka strumień do pliku

-inputAnalizer

-Zawiera pola

key(userChioce) – typ boolowski z informacją czy został użyty przełącznik „-d” („s”)

fuelType- rodzaj wybranego paliwa (usunięto)

-Zawiera metody

getKey(getUserChoice)

getFuelType(usunięto)

Spis klas:

class Distributor

}

private:

    double reFuelingTime, reFuelingSpeed, stationFuelLevel, fuelTaken;

    char startButton, endButton;

    std::shared\_ptr<Fuel> fuelType;

public:

    Distributor();

    void setFuelType();

    void getStationFuelLevel();

    float giveStationFuelLevel();

    void getStartButton();

    void getEndButton();

    void reFuel();

    float giveFuelTaken();

class Customer

{

private:

    float bill, fuelTaken;

public:

    Customer();

    void saveBill();

    float giveBill();

    void setFuelTaken();

    float giveFuelTaken();

};

class BillGenerator

{

private:

    float calculatedFee;

    std::shared\_ptr<Fuel> fuelType;

    std::shared\_ptr<Distributor> calculatingParameter;

    std::shared\_ptr<Customer> customer;

public:

    BillGenerator();

    void setFuelType();

    void calculateFee();

    void makeBill();

};

class Fuel

{

public:

    virtual void giveFuel()=0;

    virtual void loadFuel()=0;

    virtual float giveFuelStatus()=0;

    virtual float giveFuelPrice()=0;

    virtual std::string giveFuelName()=0;

    virtual ~Fuel(){};

};

class Gas : public Fuel

{

private:

    float stationFuleLevel, price;

    std::string fuelName;

public:

    Gas();

    virtual void giveFuel();

    virtual void loadFuel();

    virtual float giveFuelStatus();

    virtual float giveFuelPrice();

    virtual std::string giveFuelName();

};

class Disel : public Fuel

{

private:

    float stationFuleLevel, price;

    std::string fuelName;

public:

    Disel();

    virtual void giveFuel();

    virtual void loadFuel();

    virtual float giveFuelStatus();

    virtual float giveFuelPrice();

    virtual std::string giveFuelName();

};

class Petrol : public Fuel

{

private:

    float stationFuleLevel, price;

    std::string fuelName;

public:

    Petrol();

    virtual void giveFuel();

    virtual void loadFuel();

    virtual float giveFuelStatus();

    virtual float giveFuelPrice();

    virtual std::string giveFuelName();

};

class InputAnalizer

{

private:

    char userChoice;

public:

    InputAnalizer();

    void getUserChoice();

    void interpretUserChoice();

    char giveUserChioce();

};

class MainMenu

{

public:

    MainMenu();

    void printMainMenu();

};

class MainSequence

{

private:

    int isRunning;

    std::shared\_ptr<Fuel> disel;

    std::shared\_ptr<Fuel> petrol;

    std::shared\_ptr<Fuel> gas;

    std::shared\_ptr<MainMenu> mainMenu;

    std::shared\_ptr<SupplyMenu> supplyMenu;

    std::shared\_ptr<InputAnalizer> inputAnalizer;

    std::shared\_ptr<Distributor> distributor;

    std::shared\_ptr<Supply> supply;

    std::shared\_ptr<Customer> customer;

    std::shared\_ptr<BillGenerator> billGenerator;

    std::shared\_ptr<RaportGenerator> raportGenerator;

public:

    MainSequence();

    void resorceAlocator();

    void reFuelingSequence();

    void supplySequence();

    void supplyMeneger();

    void printFinalFuelSatus();

    void \_do\_();

};

class RaportGenerator

{

private:

    std::shared\_ptr<Fuel> fuelType;

    std::shared\_ptr<Customer> customer;

    std::shared\_ptr<Supply> supply;

    std::fstream file;

    std::string fuelName;

    std::string fileName;

    float customerBill, customerFuel;

    int i;

public:

    RaportGenerator();

    void setFuelType();

    void getCustomerParameters();

    void getFuelName();

    void openFileStream();

    void closeFileStream();

    void makeReFuelingRaport();

    void makeSypplyRaport();

};

class Supply

{

private:

    float fuelGiven;

    std::shared\_ptr<Fuel> fuelType;

public:

    Supply();

    void getFuelSupply();

    void reSupply();

    void setFuelType();

   float giveSuppliedFuel();

};

class SupplyMenu

{

public:

    SupplyMenu();

    void printSupplyMenu();

};

Relacje klas:

Petrol, Disel i Gas dziedziczą po Fuel

BillGenerator wykorzystuje metody klasy Customer i Fuel

Distributor wykorzystuje metody klasy Fuel

MainSequence wykorzustuje metody wszystkich klas

RaportGenerator wykorzystuje metody klas Customer, Fuel i Supply

Supply wykorzystuje metody klasy Fuel

Opis ważnych elementów:

Metoda distributor->reFuel();

Metoda wykorzystuje zawartość biblioteki „std::chrono”. Wykorzystuje ją do wyliczenia ilości zatankowanego paliwa. Robi to na podstawie czasu, który upłyną od pierwszego wciśnięcia klawisz „r” przez użytkownika i potwierdzenia klawiszem Enter do ponownego wykonania tej czynności. Potem mnoży uzyskany czas przez szybkość z jaką dystrybutor podaj paliwo(zadanym w konstruktorze) otrzymuje wynik, który następnie jest podawany do kolejnych klas.

Instrukcja:

Tankowanie:

Po uruchomieniu należy wybrać rodzaj paliwa, który chce się zatankować. Następnie program poprosi o wprowadzenie litery „r” i kliknięcie Enter. Potem program zaczyna liczyć czas, należy odczekać tyle ile chcemy i ponownie wykonać w.w. czynności(jest to wypisywane na ekran podczas działania programu). Na końcu program doda nasz rachunek do raportu i ponownie wyświetli Menu Główne.

Dostawa:

Gdy wyświetli się Menu Główne należy kliknąć klawisz „s” i zatwierdzić Enterem. Pojawi się prośba o wybór rodzaju paliwa, którego dostawę chcemy przeprowadzić, a następnie prośba o wprowadzenie ilości paliwa w jaka jest w dostawie. Na końcu program doda wpis o dostawie do raportu i ponownie wyświetli Menu Główne.

Wyniki działania programu:

Zawartość pliku raport.txt:

Klient 1

Rodzaj paliwa: Disel

Zatankowano: 14.01 L

Za kwotę: 54.51 zł

Stan paliwa: 985.99 L

Klient 2

Rodzaj paliwa: Gaz

Zatankowano: 12.11 L

Za kwotę: 30.04 zł

Stan paliwa: 987.89 L

Klient 3

Rodzaj paliwa: Benzyna

Zatankowano: 21.61 L

Za kwotę: 94.87 zł

Stan paliwa: 978.39 L

Klient 4

Rodzaj paliwa: Gaz

Zatankowano: 20.04 L

Za kwotę: 49.69 zł

Stan paliwa: 967.85 L

Dostawa

{

Typ paliwa: Gaz

Dostarczono: 55.00 L

Stan paliwa: 1022.85 L

}

Klient 5

Rodzaj paliwa: Benzyna

Zatankowano: 17.82 L

Za kwotę: 78.24 zł

Stan paliwa: 960.57 L

Klient 6

Rodzaj paliwa: Disel

Zatankowano: 34.82 L

Za kwotę: 135.45 zł

Stan paliwa: 951.17 L

Dostawa

{

Typ paliwa: Disel

Dostarczono: 100.00 L

Stan paliwa: 1051.17 L

}

Dostawa

{

Typ paliwa: Gaz

Dostarczono: 10.00 L

Stan paliwa: 1032.85 L

}

Klient 7

Rodzaj paliwa: Disel

Zatankowano: 0.71 L

Za kwotę: 2.75 zł

Stan paliwa: 1050.46 L

Dostawa

{

Typ paliwa: Benzyna

Dostarczono: 115.00 L

Stan paliwa: 1075.57 L

}

Klient 8

Rodzaj paliwa: Benzyna

Zatankowano: 19.90 L

Za kwotę: 87.38 zł

Stan paliwa: 1055.66 L

Klient 9

Rodzaj paliwa: Disel

Zatankowano: 19.75 L

Za kwotę: 76.83 zł

Stan paliwa: 1030.71 L