1. Temat

„Program/Moduł do obsługi sklepu na stacji benzynowej”

1. Interfejs aplikacji

- Uruchamianie z poziomu powłoki

- zastosowanie menu użytkowania typu naciśnij przycisk

1. Oczekiwany wynik działania programu

- Wyświetlanie dostępnych artykułów

- Prowadzenie koszyka

- Prowadzenie rejestrów

- pokazanie stanu, ile do zapłaty

- kupienie przedmiotu

1. Analiza obiektów występujących w trakcie działania aplikacji

Do

Obiekt wykonujący działanie kup, zapłać, wyświetl stan

InputArgumentAnalyzer

Obiekt analizujący argument wywołania czynności w sklepie

MarketStan

Obiekt przetrzymujący aktualne informacje o stanie sklepu

Basket

Obiekt przetrzymujący listę rzeczy w koszyku

Virtualgrocery

Obiekt przetrzymujący metody do włożenia produktu(Grocery) do koszyka, podliczenia itp.

Drinks

Obiekt zawierający informacje o napojach i ich cenie

Food

Obiekt zawierający informacje o jedzeniu i jego cenie

Others

Obiekt zawierający informacje o innych przedmiotach i ich cenie

CashRegister

Obiekt podliczający koszyk

RegisterMaker

Obiekt tworzący rejestr sklepu

- Opcjonalnie

MarketFileCreator

Obiekt tworzący zaopatrzenie sklepu z pliku

1. Przypadki użycia (tworzenie historii)

Przykład wywoływania: nazwa programu

Pokazanie menu -> Analizujemy argumenty wejściowe -> przekazanie dalszego działania do odpowiednich obiektów -> wyświetlenie działania -> pokazanie menu…

1. Statyczna analiza typów i interfejsów poszczególnych modułów

Do

- abstrakcyjna, agregacja z InputArgumentAnalyzer

- Przechowuje metodę Switch działań sklepu

InputArgumentAnalyzer

- klasa bazowa

- pobiera i przechowuje wybór użytkownika

char input;

MarketStan

- agregacja z klasą Food, Drinks, Others

- przechowuje stan zawartości sklepu

Dictionary stan; // użyję std::map

Basket

- klasa bazowa

- przechowuje elementy dodane do koszyka oraz ile do zapłaty

Dictionary basket; // użyję std::map

Float doZapłaty;

Virtualgrocery

- klas bazowa, typ abstrakcyjny

- klasa przechowuje metody wirtualne

Driknks

- klasa dzicząca po Vitualgrocery

- przechowuje struktury woda, sok, energetyk

Każda z struktur ma pola

char nazwa;

int ilość = 60;

float cena;

Food

- klasa dzicząca po Vitualgrocery

- przechowuje struktury hotdog, zapiekanka

Każda z struktur ma pola

char nazwa;

int ilość = 60;

float cena;

Others

- klasa dzicząca po Vitualgrocery

- przechowuje struktury Olej, ledy

Każda z struktur ma pola

char nazwa;

int ilość = 60;

CashRegister

- klasa agregacja z Basket

- przechowuje stan kasy

RegisterMaker

-klasa tworząc rejestr

- przechowuje plik