МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №2

з дисципліни «Стек технологій .NET»

Виконав:

Студент групи КН-36а

Рубан Ю. Д.

Перевірила:

Доц. Літвінова Ю. С.

Харків – 2018

**Тема:**

Аналіз об’єкта інформації та моделювання структури програмної системи на верхньому рівні: розробка діаграми класів.

**Ціль:**

Виконати аналіз об’єкта інформації та змоделювати структури майбутньої програмної системи на верхньому рівні у вигляді кінчених діаграм класів.

**Функції получені у результаті аналізу діаграми прецедентів:**

1) Замовлення товару

2) Обробити заявку

3) Визначитися з необхідними ресурсами та обладненням

4) Перевірити чи є місце на складі та чи є товар на складі

5) Купити матеріали

6) Відправити ресурси на склад

7) Підготовити тех. процес товару

8) Виготовити товар

9) Відправити товар на склад

10) Розрахувати кількість грузовиків для доставки

11) Погрузити товар

12) Доставити товар замовнику

13) Підтвердити получення товару

**Письмовий опис сущностей системи, получений у результаті аналізу діаграми прецедентів:**

1) Material: базовий клас для ресурсів та обладненя, має назву Name, та має ціну Price, та поле IsOneTimeUse яке показує чи одноразовий це матеріал

2) Resource: нащадок класу Material, займає площу на складі Area

3) Tool: нащадок класу Material, має дату придбання BuyDate

4) Product: нащадок класу Material, товар, який виробляє виробництво, має унікальний ідентифікатор Id, та поле яке показує чи виготовлений цей товар IsMade, Area показує скільки місця займе продукт на складі

5) ProductCollection: клас, який містить колекцію продуктів, та має функцію підрахунку вартості товарів TotalPrice()

6) PayBuilder: клас який на основі колекції продуктів ProductCollection приймає оплату замовлення, має функцію Pay() яка приймає оплачену суму, та повертає об'єкт заявки Order

7) Order: клас заявки на купівлю товару, має перелік замовлених товарів ProductCollection, поле яке показує чи оброблена ця заявка IsProcessed та статичну функцію MakeOrder() яка приймає колекцію товарів Product та повертає PayBuilder.

8) OrderProcessor: статичний клас, який має функцію ProcessOrder(), яка приймає заявку Order, та повертає об'єкт MaterialsCollection, та відмічає заявку як оброблену

9) MaterialsCollection: клас, який містить функцію NeededPlace() яка повертає значення необхідної площі, щоб розмістити матеріали на складі, та має поле Dictionary<Product,Material[]> яке асоціює певний продукт з потрібними матеріалами для його виробництва, конструктор приймає об'єкт ProductCollection.

10) IStorage: інтерфейс складу, має функцію Push() яка приймає колекцію матеріалів, або один матеріал, та фунцію Pop(), яка видає матеріал за його назвою, та функцію IsInStorage() яка приймає назву матеріалу, та кількість і повертає bool

11) Storage: реалізує інтерфейс IStorage, також має словник Dictionary<string, Material[]>, та поле Area - сумарна площа складу

12) StorageDecorator: реалізує інтерфейс IStorage, приймає у конструкторі об'єкт IStorage, та у методах Push(), Pop(), IsInStorage() та реалізує додаткову бізнес логіку

13) StorageFactory: статичний клас, який спрощує створення об'єктів IStorage, має метод Create(), який повертає IStorage

14) MaterialChecker: має метод GetNeededMaterials(), який приймає колекцію необхідних матеріалів та продуктів, та об'єкт IStorage і повертає коллекцію матеріалів яких не хватає на складі

14) IMaterialBuyer: інтерфейс, який має метод BuyMaterials(), який відповідає за закупівлю матеріалів повертає колекцію закуплених матеріалів.

15) MaterialBuyer: реалізує інтерфейс IMaterialBuyer, має поле MaterialChecker.

16) Guild: класс який відповідає за роботу цеху, має поле IStorage та IProductMaker і функцію MakeGoods(), яка робить товари, заносить їх до складу та повертає колекцію товарів з міткою IsMade=true, та приймає об'єкти MaterialCollection і ProductCollection

17) IProductMaker: інтерфейс який відповідає за виробництво одного конкретного товара, має метод MakeProduct() який приймає продукт, та колекцію матеріалів

18) ProductMaker: реалізує інтерфейс IProductMaker

19) Truck: клас який моделює грузовик що може доставляти продукти, має поле колекції продуктів Products та поле Area скільки площі в нього можна помістити

19) DeliveryService: клас який реалізує патерн сінглтона, функція GetTrucks() яка підраховує кількість необхідних грузовиків, та повертає їх колекцію, та метод LoadTrucks() який заповньює колекцію грузовиків продуктами. Останім методом викликається метод Deliver() який доставляє замовнику колекцію грузовиків для подальшого використання.

20) NotEnoughtSpaceException: клас виключення, який показує що у об'єкта не хватає місця. Нащадок класа Exception

21) UnpackingService: клас, який відповідає за те, щоб розгрузити колекцію продуктів Product[] з грузовиків Truck[], також він перевіряє товари згідно з замовленням Order в функції Unpack().

**Діаграма класів**

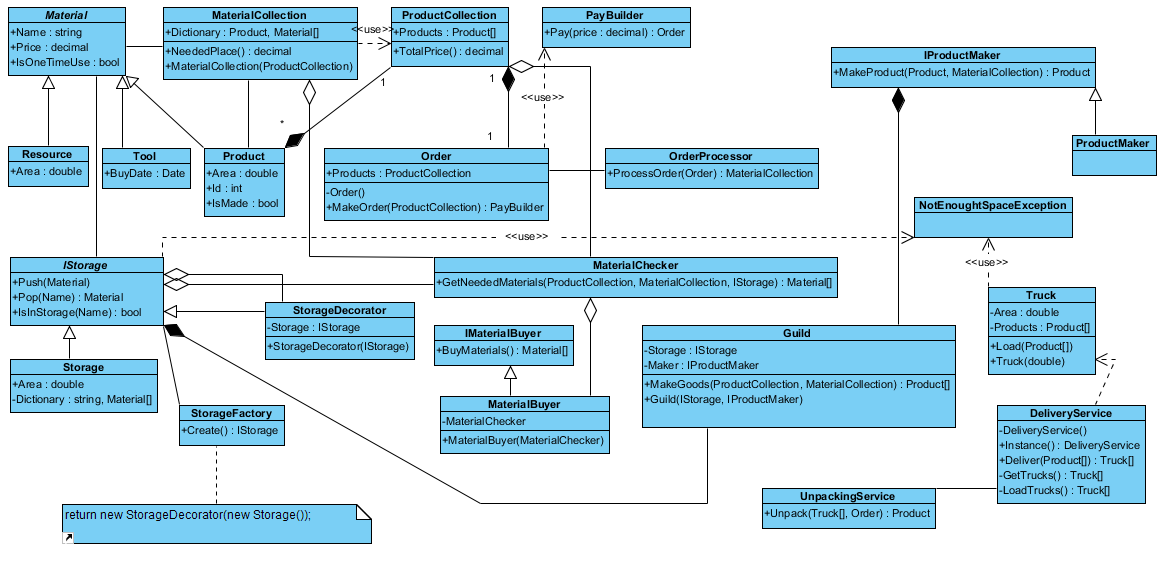
Діаграма класів показана на рисунку 1.

Рисунок 1 – Діаграма класів

**Висновки:**

У даній лабораторній роботі було виконано аналіз об’єкта інформації та змоделювано структури майбутньої програмної системи на верхньому рівні у вигляді кінчених діаграм класів.