НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра Автоматизованих Систем Обробки Інформації та Управління

# КУРСОВА РОБОТА

з дисципліни " Програмування - 2. Структури даних та алгоритми"

На тему: Ломбард

Студента першого курсу групи ІС-92

Шульмана Дениса Олеговича

Спеціальності: Інформаційні системи та технології

Керівник: старший викладач Проскура Світлана Леонідівна

Кількість балів:\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка:\_\_\_\_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_ доцент Крамар Юлія Михайлівна

(підпис)

\_\_\_\_\_\_\_\_\_\_ старший викладач Проскура Світлана Леонідівна

(підпис)

Київ- 2020 рік

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРИТЕТ УКРАЇНИ «КПІ ІМ. СІКОРСЬКОГО»

Кафеда автоматизованих систем обробки інформації і управління

Дисципліна «Структури даних та алгоритми»

Спеціальність 126 «Інформаційні системи та технології»

Курс 1, Група ІС-92, Семестр 2

# ЗАВДАННЯ

**на курсову роботу студента**

|  |
| --- |
| Шульмана Дениса Олеговича |

1. Тема проекту(роботи):

Ломбард. Запозичення і повертання грошей та облік речей під заставу

2. Строк здачі студенткою закінченого проекту (роботи):

До 5 червня 2020 року.

3. Вихідні дані до проекту(роботи):

Операційна система, середовище IDE Visual Studio 2019, мова програмування C#.NET, опис предметної галузі, стартовий набір з двох клієнтів та одного працівника ломбарду.

4. Опис предметної галузі проекту:

Ломбард надає послуги запозичання грошей під заставу цінних

речей. Клієнти в обмін на гроші надають під заставу речі на

певний термін. За бажанням клієнта та категорією речі під заставу

річ може бути викупленою першим бажаючим речей певної

категорії, чи не пізніше певного строку її минулим власником, або

зі зростаючим процентом боргу для власника в залежності від

терміну зберігання. У клієнтів є кредитна історія, що також може

впливати на умови запозичання грошей. Ломбард підраховує

прибуток та витрати

5. Зміст розрахунково-пояснювальної записки:

Титульний аркуш, лист завдання, зміст, вступ, аналіз предметної області, кодування, опис програмного забезпечення, результати тестування, висновки, перелік посилань, додатки.

6. Дата видачі завдання: 1 квітня 2020 року.

# КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
|  | Назва етапів  курсової роботи | Термін виконання етапу | Підпис студента та викладача |
| 1 | Отримання теми КР | 02.04.2020 |  |
| 2 | Підготовка ПЗ | до 03.04.2020 |  |
| 3 | Аналіз предметної області | до 12.04 |  |
| 4 | Проектування архітектури програмної системи | до 19.04.2020 |  |
| 5 | Розробка сценарію роботи програми | до 25.04.2020 |  |
| 6 | Узгодження з керівником інтерфейсу користувача | до 02.05.2020 |  |
| 7 | Розробка програмного забезпечення | до 17.05.2020 |  |
| 8 | Тестування програми | до 20.05.2020 |  |
| 9 | Підготовка пояснювальної записки | до 25.05.2020 |  |
| 10 | Здача курсової роботи на перевірку | до 23.05.2020 |  |
| 11 | Захист курсової роботи | До 05.06.2020 |  |

Студент: Шульман Денис Олегович \_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Керівник: Проскура Світлана Леонідівна \_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

«\_\_»\_\_\_\_\_\_\_2020 року

# ЗМІСТ:

[КУРСОВА РОБОТА 1](#_Toc42076177)

[ЗАВДАННЯ 2](#_Toc42076178)

[КАЛЕНДАРНИЙ ПЛАН 3](#_Toc42076179)

[ЗМІСТ: 4](#_Toc42076180)

[Вступ 5](#_Toc42076181)

[1.1. Середовище розробки та мова програмування. 5](#_Toc42076182)

[1.2. Призначення курсової роботи 5](#_Toc42076183)

[1.3. Сутність вирішуваної задачі, ціль роботи 5](#_Toc42076184)

[АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7](#_Toc42076185)

[1. Аналіз стану ломбардів на даний момент 7](#_Toc42076186)

[2. Недоліки системи та способи їх подолання за допомогою програми. 7](#_Toc42076187)

[3. Конкретизація постановки завдання. 8](#_Toc42076188)

[4. Структура вхідних та вихідних даних 8](#_Toc42076189)

[ОПИС ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ 10](#_Toc42076190)

[1. Перелік класів «Ломбард»: 10](#_Toc42076191)

[2. Типи зв’язків 10](#_Toc42076192)

[3. Детальний опис класів, їх атрибутів, методів та інтерфейсів: 10](#_Toc42076193)

[4. Діаграма класів 12](#_Toc42076194)

[РЕЗУЛЬТАТИ ТЕСТУВАННЯ 13](#_Toc42076195)

[1. Інтерфейс користувача 13](#_Toc42076196)

[2. Результати тестування 13](#_Toc42076197)

[ВИСНОВКИ 17](#_Toc42076198)

[ДЖЕРЕЛА 18](#_Toc42076199)

# Вступ

## **Середовище розробки та мова програмування**.

Реалізація програми була здійснена на одній з запропонованих мов програмування, що ми вивчали протягом семестру: C# .NET Framework. Середовищем програмування була обрана Microsoft Visual Studio 2019, що дає можливість реалізувати програму на 100%.

Перевага була надана C# а не C++, бо в цій мові зручніше працювати з класами, наявний прибиральник сміття, події та делегати.

До того ж вона може запропонувати зручне підключення динамічних бібліотек, що робить процес розробки швидше.

## Призначення курсової роботи

Розроблення програми, консольного додатку з підключенням власної бібліотеки, з використанням об'єктно-орієнтованої технології програмування мовою С#.

Практично застосувати основні концепції ООП – поліморфізм, інкапсуляцію, статичні та екземплярні члени класів бібліотеки - один із найпотужніших інструментаріїв ООП; проектування, розробка, налагодження та супровід сучасного програмного забезпечення інформаційних та комп’ютерних систем.

А також поглибити та узагальнити набуті впродовж курсу знання, засточовуючи практичні навички в області створення сучасних програмних продуктів, викопистовуючи об’єктно – орієнтовний підхід.

## Сутність вирішуваної задачі, ціль роботи

Ломбард – бізнес, що надає клієнтам кошти в обмін на речі, які можна повернути назад впродовж певного терміну, чи після нього, але з доплатою. Речі можуть бути викуплені іншими клієнтами в будь який час перебування речей в ломбарді.

Програма служить для того, аби автоматизувати процес торгівлі між працівником ломбарду та клієнтами, а також для відслідковування історії обміну товарів між суб’єктами торгівлі(працівник ломбарду та клієнти).

Кожен суб’єкт має своє власне ім’я та залишок коштів на рахунку. Кошти, з якими суб’єкт починає відносини, вводяться на початку програми.

На відміну від клієнта, працівник ломбарду має також свій ID (дозвіл на ведення бізнесу) та значення прибутку, витрат, та прибутку нетто.

Консольний додаток зроблений у вигляді меню з вибором пунктів, що слугують для визначення суб’єкту відносин та дій, що він виконує.

Ця форма є зручною для розуміння як для фахівця, так і для недосвідченого користувача.

Інформаційні приписи є структурованими та зрозумілими.

Програма передбачає такі дії: додання товарів до будь-якого суб’єкта відносин, продаж товару ломбарду в обмін на вартість товару, викуп товару клієнтом а також інформаційні повідомлення про стан суб’єкта та історію операцій.

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## Аналіз стану ломбардів на даний момент

Зі зростанням рівня безробіття, стрибками курсу валют та зростаючим відсотком людей за межею бідності, ломбарди стали невід’ємною частиною кожної вулиці, адже розмір клієнтської бази цих закладів невмолимо збільшується з кожним днем.

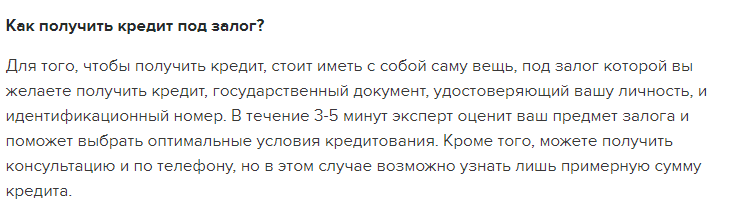
Через це, велика кількість прибуткових закладів стикнулася з проблемою перенасиченості клієнтською базою. Найкращим рішенням цієї проблеми є автоматизація процесів контролю за потоком речей та коштів

Автоматизовані програми здатні швидше за людей заносити дані про них та торгівлю до різноманітних баз, зберігати інформацію, обраховувати вартість викупу товару та забезпечувати конфіденційність даних.

## Недоліки системи та способи їх подолання за допомогою програми.

Основні проблеми, з якими стикнулися рано чи пізно усі представники даної діяльності :

1. Недостатня конфіденційність інформації: запис даних на паперових носіях легше викрасти чи зіпсувати, ніж захищений електронний варіант
2. Швидкодія: через те, що такий бізнес потребує заповнення більшої кількості паперів та перевірку документів, наприклад паспорт та ідентифікаційний номер
3. Збільшення кількості клієнтів, що автоматично приводить до більшої кількості помилок та витрат часу, якщо процес ведеться вручну.



<https://24tv.ua/ru/lombardy_ukrainy_chto_berut_v_zalog_lombardy_uslovija_dokumenty_n1194364>

Ідеальним вирішенням цих трьох пунктів є автоматизована програма, що автоматично перераховує кошти користувачу та зберігає дані про транзакції у сховищі.

## Конкретизація постановки завдання.

Отож, виходячи з потреб бізнесу та клієнтів, програма повинна виконувати такі функції:

* Кожен клієнт або працівник ломбарду повинен мати закріплений за ним перелік товарів, якими він володіє
* Кожен суб’єкт повинен мати залишок коштів на рахунку
* Кожен суб’єкт має своє ім’я(працівник до того ж має дозвіл на ведення бізнесу)
* Працівник має інформацію про прибутковість свого ломбарду або ломбарду де він працює
* Працівник ломбарду та клієнти мають змогу обмінюватись речами в обмін на гроші в залежності від умов обміну.
* За кожним суб’єктом ведеться запис його історії обміну речей
* В будь який момент часу можна дізнатися про статус кожного суб’єкта торгівлі
* При затримці з викупом товару, на клієнта накладається пеня в залежності від часу затримки

## Структура вхідних та вихідних даних

Дані, що задаються автоматично при запуску програми:

* 3 об’єкти: 2 клієнта та працівник ломбарду
* Імена суб’єктів взаємодії
* Дозвіл ведення бізнесу у працівника ломбарду

Дані, що вводяться при запуску програми:

* Кошти кожного з суб’єктів

Дані, що вводяться при роботі програми

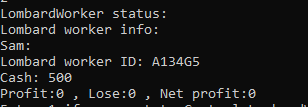
* Створюються товари для обміну
* Дані товару, обмін якого виконується(вартість, назва, категорія, час(на який товар закладається),час)
* Час, через який клієнт повернувся

Вихідні дані:

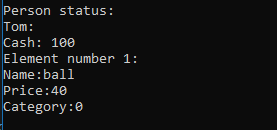
* Історія обміну товарів у клієнта та працівника
* Стан клієнта або працівника на даний момент(Клієнт: залишок на рахунку, наявні товари. Працівник: витрати, прибуток, чистий прибуток, дозвіл)
* Інформація про ломбард(назва, працівники)

Приклади:

Статус працівника:



Статус клієнта:



Інформація про ломбард:



Історія клієнта:



# ОПИС ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

## Перелік класів «Ломбард»:

* Good
* Goods1
* Goods2
* Lombard
* LombardWorker
* Person
* Sides

## Типи зв’язків

* Успадкування
  + Sides →Person
  + Sides →LombardWorker
* Реалізація
  + ISides→Sides
* Композиція
  + Goods2→Good
  + Goods1→Good
  + Lombard→LombardWorker
* Асоціація
  + Lombard→Good
  + Lombard→Person
* Агрегація
  + LombardWorker→Good
  + Person→Good
  + Lombard→Good

## Детальний опис класів, їх атрибутів, методів та інтерфейсів:

**Good:**

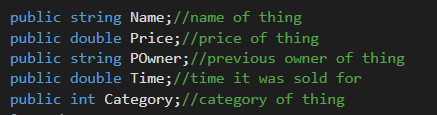
****

Рис. Поля класу Good

Від класу можна породжувати об’єкти, що слугують як речі. Має конструктор для заповнення полів.

**Lombard:**

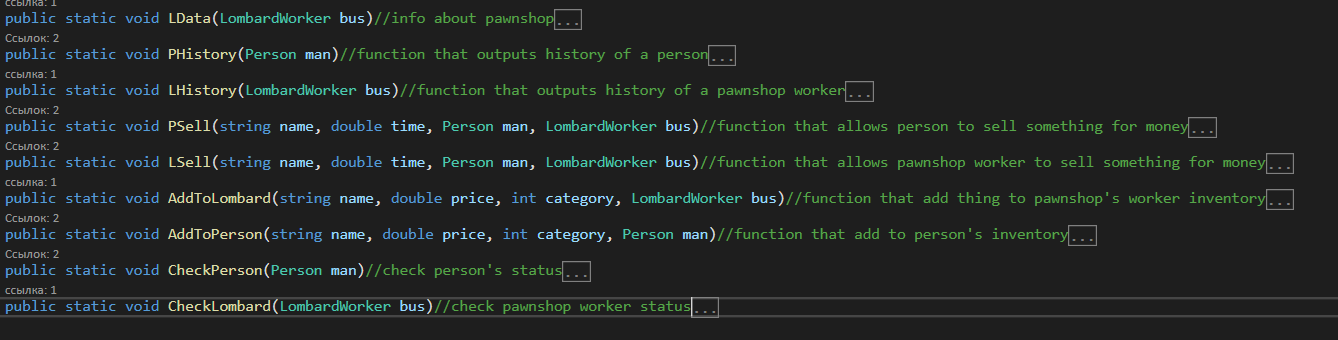
****

Рис. Поля класу Lombard

Клас з статичними функціями, що забезпечує виконання основних функцій програми.

**LombardWorker:**

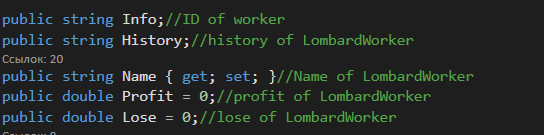
****

Рис. Поля класу LombardWorker

Клас унаслідується від класу Sides, на основі цього класу створюється об’єкт-працівник ломбарду

Має конструктор для заповнення полів

**Person:**

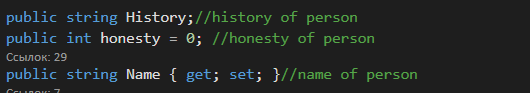
****

Рис. Поля класу Person

Клас унаслідується від класу Sides, на основі цього класу створюється об’єкт-клієнт

Має конструктор для заповнення полів

**Sides:**

Батьківський абстрактний клас для LombardWorker & Person з функціями, що будуть визначені в наступних класах.

**Goods1 & Goods2:**

Невеличкі класи для здійснення сховищ в класах LombardWorker & Person

**ISides:**

Інтерфейс, що є основою для класу Sides

## Методи:

**Add(Good thing)- метод для додавання речей до сховища суб’єкту**

**Delete(Good thing)- метьод для видалення речей з сховища суб’єкту**

**LData(LombardWorker bus)-метод для виведення даних про ломбард**

**PHistory(Person man)-виведення історії транзакцій людини**

**LHistory(LombardWorker bus)- виведення історії транзакцій працівника ломбарду**

**PSell(string name , double time, Person man, LombardWorker bus)-продажа предмету до ломбарду**

**LSell(string name, double time, Person man, LombardWorker bus)-**

**Покупка предмету з ломбарду(викуп)**

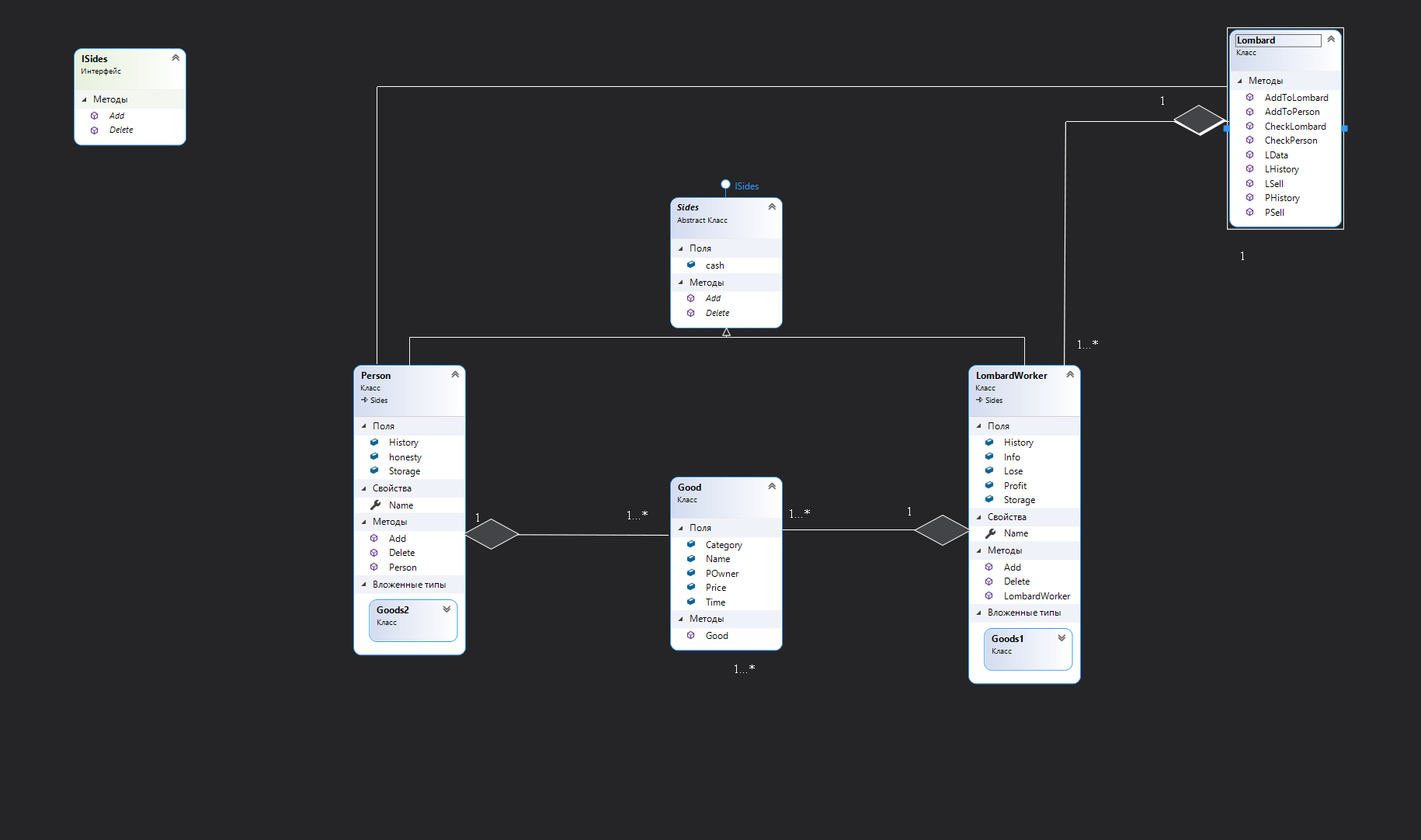
**AddToLombard(string name, double price, int category, LombardWorker bus)-додавання речей до працівника ломбарду**

**AddToPerson(string name, double price, int category, Person man)-додавання речей до клієнта**

**CheckPerson(Person man)-перевірка статусу клієнта**

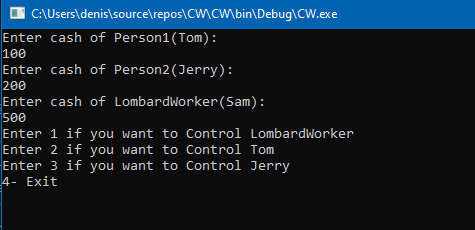
**CheckLombard(LombardWorker bus)- перевірка статусу працівника ломбарду**

## Діаграма класів

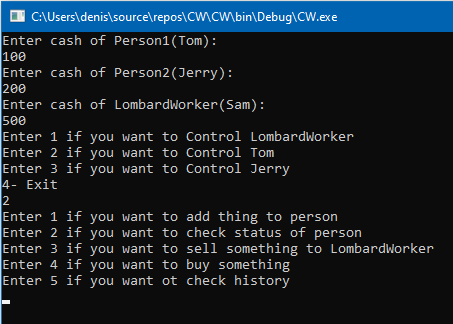


# РЕЗУЛЬТАТИ ТЕСТУВАННЯ

## Інтерфейс користувача

****

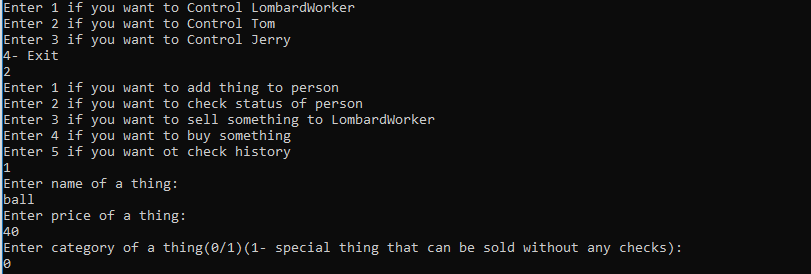
Реалізовано у вигляді меню з підпунктів.



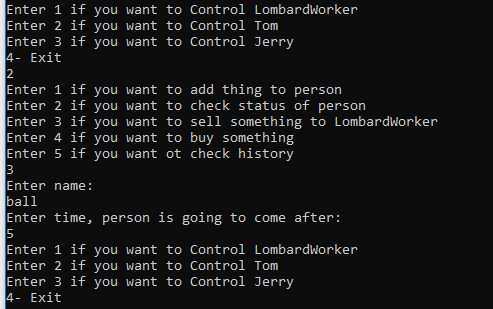
При виборі «гравця» наступним є обирання дії над ним, що також є

різними підпунктами меню.

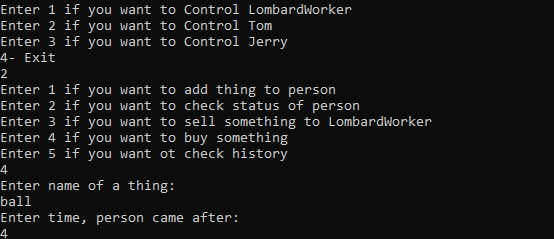
## Результати тестування

****

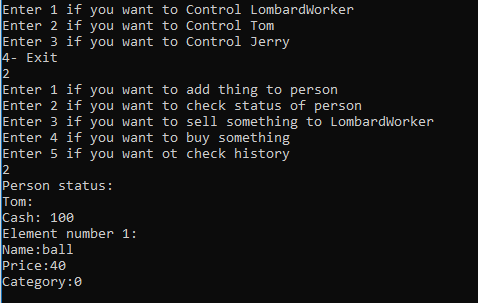
Нехай Том тепер володіє ball(надалі - м’яч) що коштує 40 грн.

****

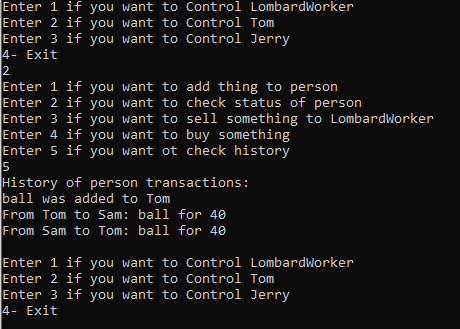
Тепер Том вирішив продати м’яча ломбарду і за контрактом повинен повернутися та викупити м’яч за 5 діб

****

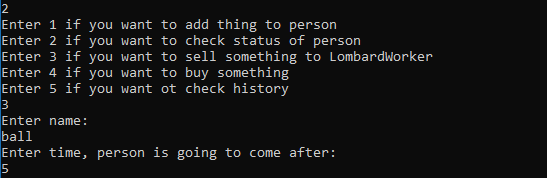
Том повернувся за 4 доби(вклався в термін), має можливість викупити за ті самі кошти, що і продав

****

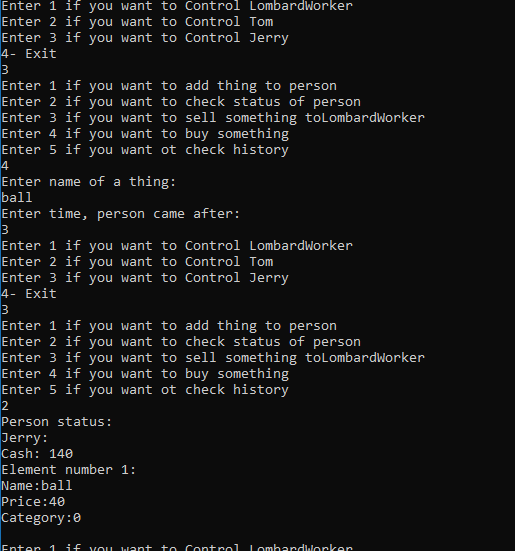
Перевіривши Тома, ми бачимо, що м’яч знову у нього і баланс коштів співпадає з очікуваним

****

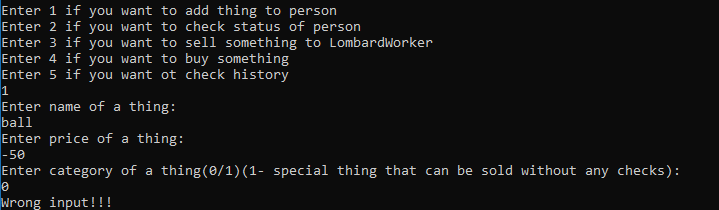
Перевіривши історію Тома, ми можемо побачити записи про усі події, що з ним відбувалися.

****

Том знову продає м’яча

****

Цього разу Джері викуповує м’яча раніше за Тома, але, оскільки це не його річ, він сплачує на 50% більше(60 грн.). Після перевірки статусу видно, що тепер Джері має м’яча, але, при вартості м’яча в 40 грн., сплачує 60 грн. і в залишу лишається 140 грн. при колишніх 200.



Програма реагує на не коректні вхідні дані, що відображається у вигляді відповідного запису

# ВИСНОВКИ

Написана мною програма симулює роботу ломбарду, а саме відносини клієнт-працівник, функціонально програма дозволяє здійснювати обмін речей з певними умовами та відслідковування стану та історії кожного з учасників.

Програма працює коректно та відслідковує введені дані на предмет їх адекватності. Процес викупу товару, що є найважливішим в роботі ломбарду, передбачає різні версії та вхідні дані взаємодії

При розробці програми були використані основні принципи ООП, а саме застосовані різні відношення між класами, взаємодія різних об’єктів.

Динамічна бібліотека реалізує предметну галузь програми, а інтерфейс користувача знаходиться в окремому проекті

Для себе, виконуючи курсову роботу та досліджуючи сферу діяльності, я виніс, що грамотне проектування та аналіз предметної області є настільки ж, а може і навіть більш важливими, ніж написання програмного коду.

При розробці належне дослідити предметну область та виділити основні «механіки» взаємодії у реальному світі та перенести їх на програмний код.

Розробка коду без підготовки та досліджень неминуче веде до «безладу» в коді аж до повної його переробки.

Отже, основним етапом створення програмного забезпечення є написання сценарію програми з відповідними класами та об’єктами, взаємодія між ними. Це вагомо полегшує та прискорює розробку

# ДЖЕРЕЛА

1. <https://24tv.ua/>
2. Презентації та відеометеріали курсу «Програмування - 2. Структури даних та алгоритми»
3. Презентації та відеометеріали курсу «Програмування - 2. Структури даних та алгоритми»;