

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Кафедра ЕОМ



Лабораторна робота №1
з дисципліни “ Інженерія програмного забезпечення ”
на тему: “ UML діаграми варіантів використання та UML діаграми активності ”

Виконав: ст. гр. КІ-303
Гедз М.Р.
Прийняв:
Цигилик Л.О.

Мета: Освоїти принципи створення UML діаграм варіантів використання, що описують сценарій роботи системи (Use case diagram) та діаграм активності для клієнтської та серверної частин.

Завдання: Розробити UML діаграми сценаріїв роботи клієнтської та серверної частин, а також розробити діаграми активності для кожного сценарію. Кількість UML діаграм повинна бути не менше – 8.

ВАРІАНТ: 5 – Система наповнення та отримання готівки у банкоматах.

Виконання

UML діаграма сценарію роботи клієнтської частини



Рис. 1. Діаграма варіантів використання клієнтської частини

Код UML-діаграми:

```
@startuml
skinparam actorStyle awesome
left to right direction
actor "Користувач" as USER

together {
usecase "Авторизація користувача" as BA1
usecase "Зміна ПІН-коду картки" as BA2
usecase "Зняття готівки" as BA3
usecase "Наповнення готівки" as BA4
}

USER -down-> BA1
USER -down-> BA2
USER -down-> BA3
USER -down-> BA4

actor "Сервер" as SERVER

BA1 -down-> SERVER
BA2 -down-> SERVER
BA3 -down-> SERVER
BA4 -down-> SERVER
@endu
```

Опис клієнтської частини

Дана діаграма зображує можливі сценарії роботи клієнтської частини.

Оскільки система взаємодіє з користувачем і сервером, на ній присутні два актори: Користувач і Сервер. Після авторизації користувача у нас може бути три варіанта подій: перегляд балансу, зняття готівки та зміна ПІН-коду. Для спеціальних користувачів (інкасатори), доступна функція наповнення готівкою банкомат. Кожний подальший варіант використання має прямий зв'язок з входом у систему та сервером.

Передача даних можлива як від клієнта до сервера, так і від сервера до клієнта.

При спробі увійти до системи в базі даних ідентифікується користувач. Після цього клієнт має пройти підтвердження. На стороні сервера обробляється дані об'єкта і в разі небезпеки повідомляється клієнту.

Кожен пакет, переданий від клієнта до сервера по протоколам TCP/IP сервер перевіряє на цілісність помилки і декодує.

Серверна частина

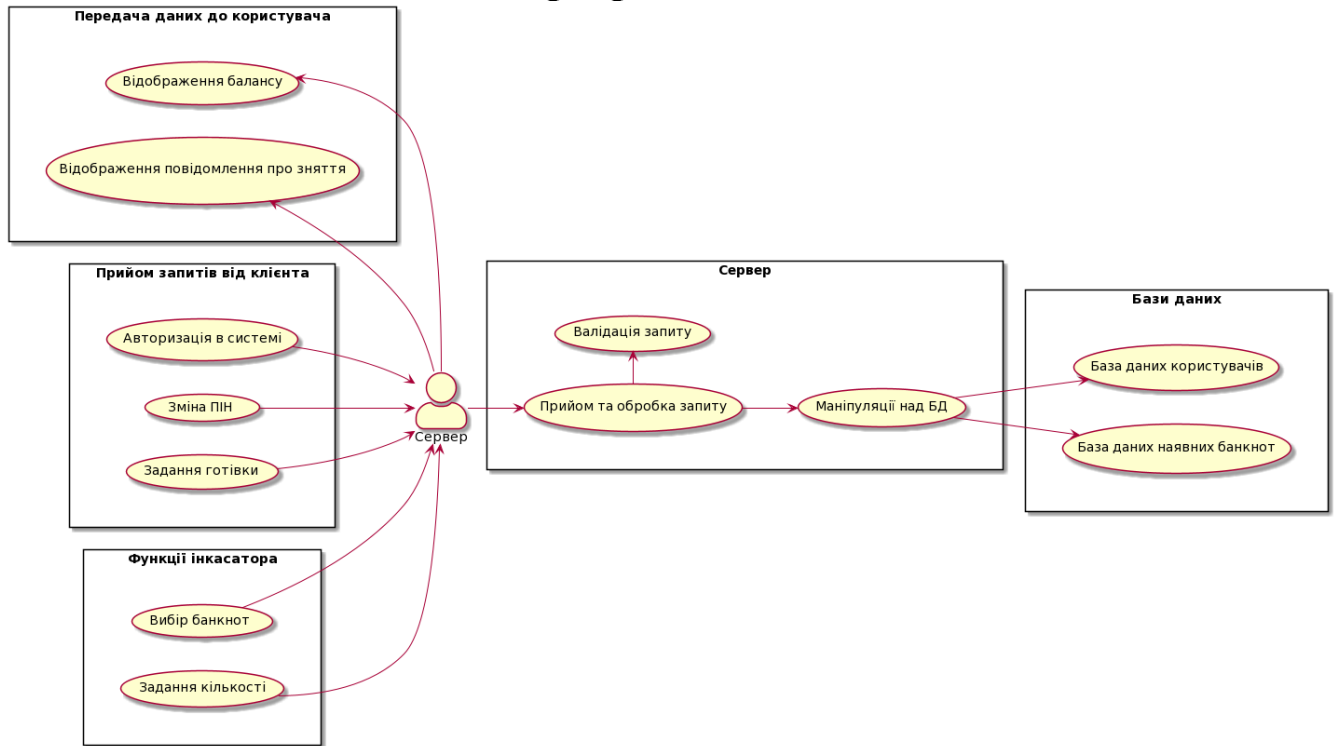


Рис. 2. UML діаграма прецедентів роботи сервера

Код UML діаграми:

```
@startuml
left to right direction
skinparam actorStyle awesome
actor "Сервер" as SERVER
rectangle "Передача даних до користувача"{
together {
usecase "Відображення повідомлення про зняття" as BB1
usecase "Відображення балансу" as BB2
}
}

rectangle "Прийом запитів від клієнта"{
together {
usecase "Авторизація в системі" as BA1
usecase "Зміна ПІН" as BA2
usecase "Задання готівки" as BA3
}
}

rectangle "Функції інкасатора"{
together {
usecase "Вибір банкнот" as BB4
usecase "Задання кількості" as BB5
}
}

rectangle "Сервер" {
together {
usecase "Прийом та обробка запиту" as BS1
usecase "Валідація запиту" as BS2
usecase "Маніпуляції над БД" as BS3
}
}
```

```

}
rectangle "Бази даних"{
together {
usecase "База даних користувачів" as BQ1
usecase "База даних наявних банкнот" as BQ2
}
}

```

```

BA1 -down-> SERVER
BA2 -down-> SERVER
BA3 -down-> SERVER
BB4 -down-> SERVER
BB5 -down-> SERVER
SERVER -up-> BB1
SERVER -up-> BB2
SERVER -down-> BS1
BS1 -right-> BS2
BS1 -down-> BS3
BS3 -down-> BQ1
BS3 -down-> BQ2
@enduml

```

Опис серверної частини

Дана діаграма зображує можливі сценарії роботи серверної частини. На сервер подаються дані авторизації облікового запису, запиту зміни ПІН картки, запитом зняття готівки, а також перевірки балансу картки.

Після прийому даних, сервер виконує запит до відповідних баз даних: бази даних користувачів, та наявних банкнот.

Клієнтська частина

Процес авторизації

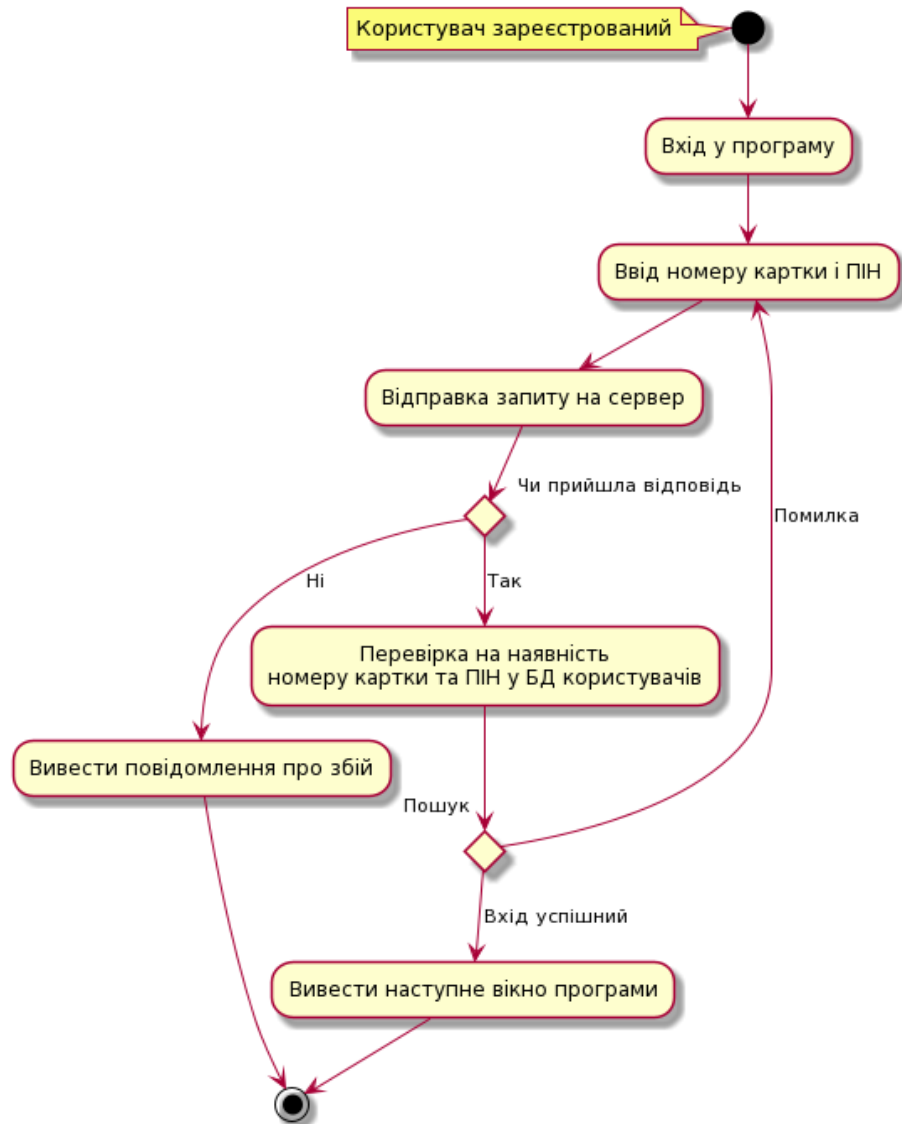


Рис. 3. UML-діаграма процесу авторизації

UML код діаграми:

```
@startuml
note left: Користувач зареєстрований
(*) -down-> "Вхід у програму"
--> "Ввід номеру картки і ПІН"
--> "Відправка запиту на сервер"
--> if "Чи прийшла відповідь" then

--> [Так] "Перевірка на наявність номеру картки та ПІН у БД користувачів"

--> if "Пошук" then
--> [Вхід успішний] "Вивести наступне вікно програми"
--> (*)
else
---> [Помилка] "Ввід номеру картки і ПІН"
endif

else
-->[Ні] "Вивести повідомлення про збій"
-->(*)
endif
@enduml
```

Опис роботи діаграми:

Дані діаграми описують послідовність дій, які використовуються при вході користувача у систему. Користувач у головному вікні авторизується в системі.

Після цього дані відправляються на сервер, йде перевірка чи дані дійшли до сервера. Потім йде перевірка чи цей логін був знайдений у БД. Якщо був знайдений, то користувачу надається доступ до системи та виводиться на сторінку повідомлення про успішну авторизацію в системі.

Процес зняття готівки

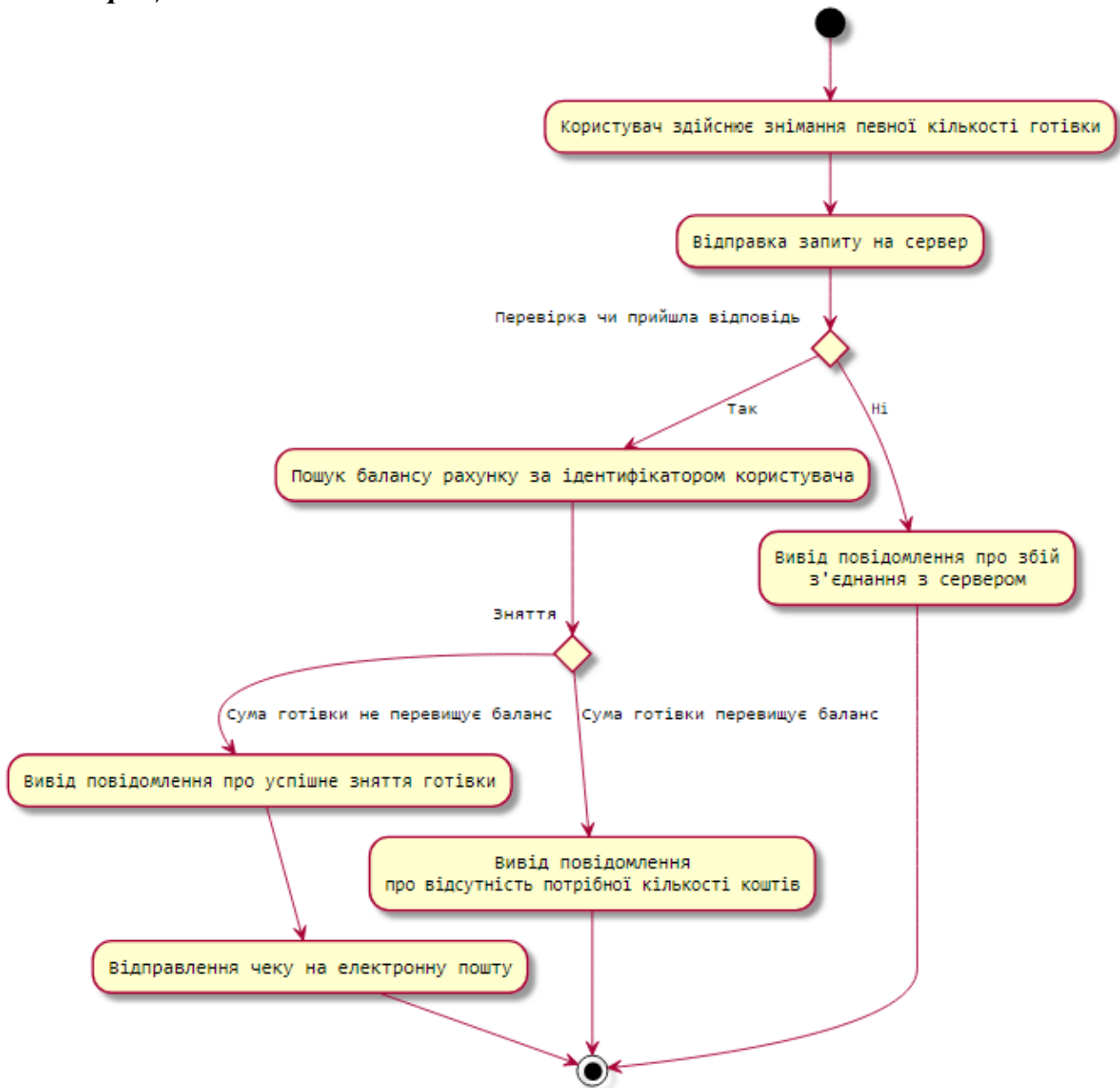


Рис. 4. Діаграма процесу зняття готівки

Опис діаграми:

Дана діаграма описує процес зняття готівки зі сторони клієнтської частини. Користувач вводить суму зняття або обирає стандартне значення. Далі надсилається запит на сервер, де йде перевірка, чи є вистачає коштів на балансі користувача. При знаходженні, виводить повідомлення про успішне зняття. Якщо не знаходиться, то виводиться повідомлення про відсутність потягів.

Процес наповнення готівкою інкасатором

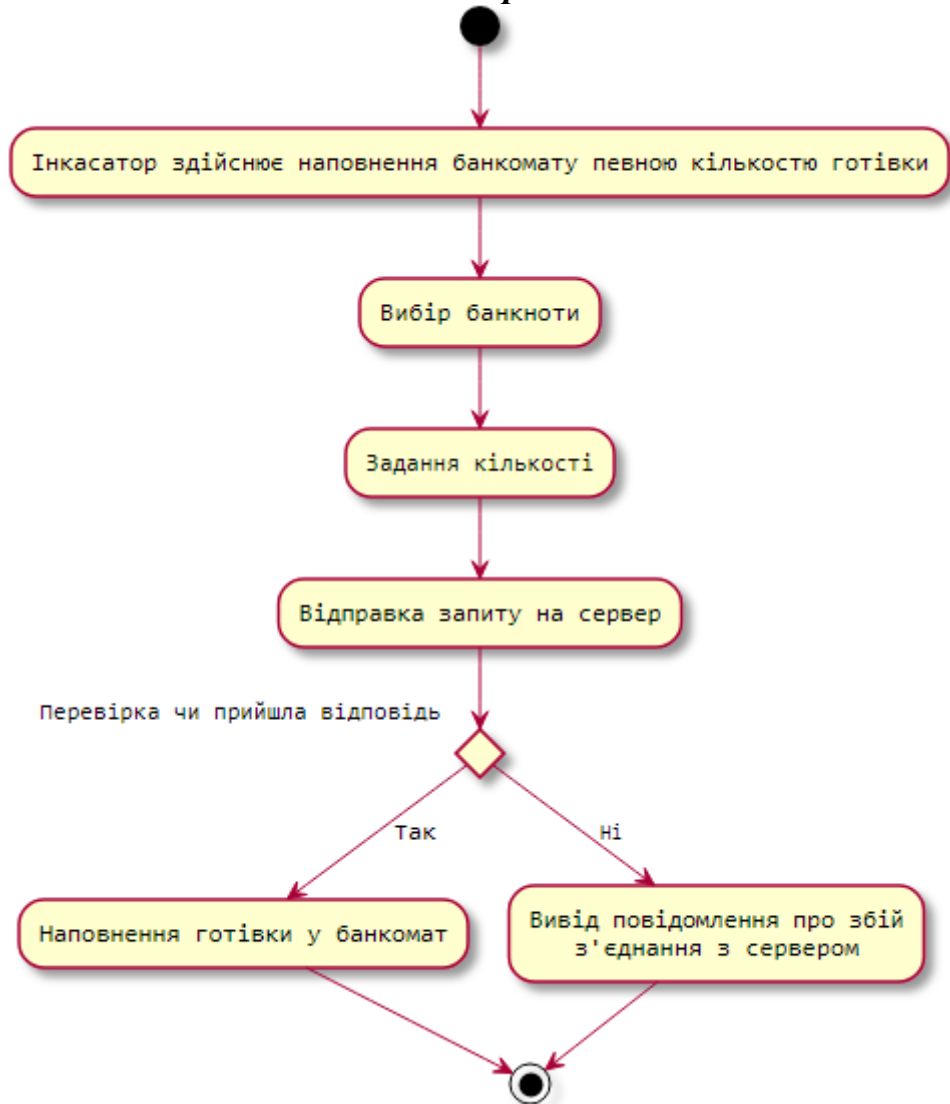


Рис. 5. Діаграма процесу наповнення готівкою інкасатором

Опис діаграми:

Дана діаграма описує процес наповнення готівки інкасатором. У клієнтській програмі інкасатор вказує номінал, потім кількість готівки. Потім надсилається запит на сервер. Якщо прийшла відповідь – виводиться повідомлення про успішне наповнення. Інакше, виводить повідомлення про збій з'єднання з сервером.

Серверна частина

Процес авторизації користувачів



Рис. 6. UML діаграма активності процесу авторизації користувачів

UML код діаграми:

```
@startuml
start
:Авторизація користувача;
if (Логін та ПІН код є у БД?) then (Ні)
:Надсилання коду про невдалу авторизацію;
else (Так)
:Надсилання коду про успішну авторизацію, а також ім'я користувача та баланс його картки;
endif
stop
@enduml
```

Опис діаграми:

Дана діаграма описує послідовність дій, які виконуються при пошуку існуючих користувачів у базі даних. База даних отримує логін користувача та ПІН і перевіряє чи логін є у базі даних. Якщо логіна там немає, то надсилається код про невдалу авторизацію. Якщо логін є у базі даних - то виконується вхід та надсилається сповіщення про успішну авторизацію.

Процес зняття готівки:



Рис. 7. Діаграма активності процесу зняття готівки у сервері

UML код діаграми:

```
@startuml
start
:Прийом запиту на зняття готівки;
if(Чи достатньо коштів на балансі для зняття?) then (Так)
:Віднімання суми від балансу клієнта;
:Зменшення загальної суми у банкоматі на суму знятої готівки;
:Відправка чеку на електронну пошту;
else (Ні)
:Відправлення коду про нестачу коштів на балансі клієнта;
endif
stop
@enduml
```

Опис діаграми:

Дана діаграма описує послідовність дій при знятті готівки у сервері. При надходженні запиту користувача, сервер починає пошук балансу картки по базі даних. Якщо баланс більше, або рівне сумі знімаємих коштів, то йде процес зняття, а саме: віднімання від балансу суми, занесення у БД нового балансу, надсилання відповіді на клієнт та відправлення чеку на e-mail. У інакшому випадку (баланс менше суми), сервер надсилає повідомлення про відсутність достатньої кількості коштів.

Процес наповнення банкомата готівкою:

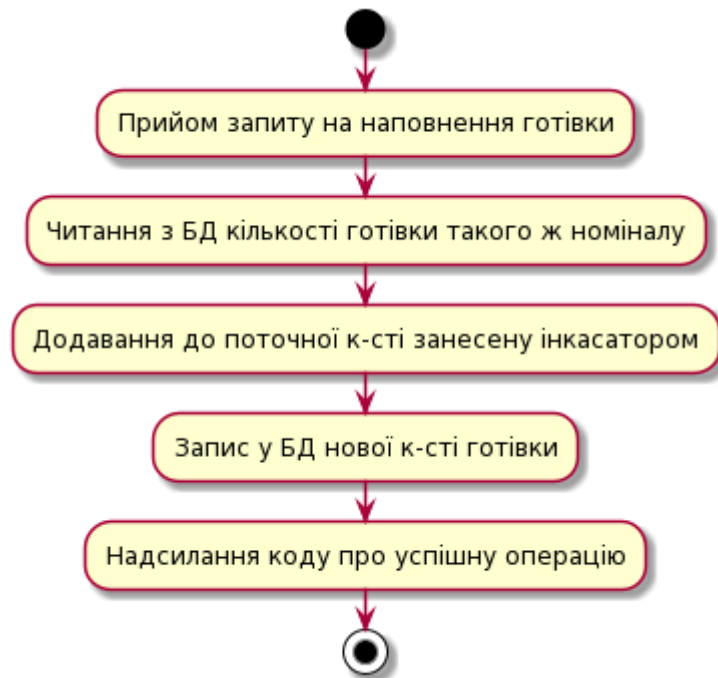


Рис. 8. Діаграма активності процесу наповнення готівки у сервері

UML код діаграми:

```
@startuml
start
:Прийом запиту на наповнення готівки;
:Читання з БД кількості готівки такого ж номіналу;
:Додавання до поточної к-сті занесену інкасатором;
:Запис у БД нової к-сті готівки;
:Надсилання коду про успішну операцію;
stop
@enduml
```

Опис діаграми:

Дана діаграма описує процес наповнення готівки до банкомату. Сервер приймає запит від інкасатора, потім з БД зчитується кількість готівки такого ж номіналу. Згодом до тої кількості додається число, введене інкасатором. Далі ця сума заноситься у БД, і надсилається код про успішну операцію.