

## ДОДАТОК А

Лістинг фрагментів коду програми

```
package LU_classes;

import java.io.Serializable;

/**
 * Описывает товар ( название, цена, полезность, ..)
 * @author std
 */
public class Commodity implements Serializable {
    private String name=null;
    private Double price = 0.0;
    private String unit=null; // единицы измерения
    private Double utility = 0.0;
    private Double decrease_coefficient = 0.0;

    public Commodity(String name, double price, String unit, double
utility, double decr) {
        this.name = name;
        this.price = price;
        this.unit = unit;
        this.utility = utility;
        decrease_coefficient = decr;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Double getPrice() {
        return price;
    }

    public void setPrice(Double price) {
        this.price = price;
    }

    public String getUnit() {
        return unit;
    }

    public void setUnit(String unit) {
        this.unit = unit;
    }

    public Double getUtility() {
        return utility;
    }
}
```

```

public void setUtility(Double utility) {
    this.utility = utility;
}

/**
 * @return the decrease_coefficient
 */
public Double getDecrease_coefficient() {
    return decrease_coefficient;
}

/**
 * @param decrease_coefficient the decrease_coefficient to set
 */
public void setDecrease_coefficient(Double decrease_coefficient)
{
    this.decrease_coefficient = decrease_coefficient;
}
}

package LU_classes;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.HashSet;
import java.util.Iterator;

/**
 * База хранения товаров ( добавление, удаление, хранение )
 * @author std
 */
public class Goods implements Serializable{
    private HashSet<Commodity> all_goods = new HashSet();

    public Goods(){
        super();
    }

    public void add_commodity(Commodity lap) {
        all_goods.add(lap);
        save_goods();
    }

    public boolean del_commodity(Commodity lap) {
        boolean result = false;
        Iterator interator = all_goods.iterator();
        Commodity delel = null;

```

```

while (interator.hasNext()) {
    Commodity element = (Commodity) interator.next();
    if (element.getName().equals(lap.getName()) &&
        element.getPrice().equals(lap.getPrice())) {
        //all_log_pass.remove(element);
        delel = element;
        result = true;
        break;
    }
}
all_goods.remove(delel);
save_goods();
return result;
}

public boolean contains_commodity(Commodity lap) {
    boolean result = false;
    Iterator interator = all_goods.iterator();
    while (interator.hasNext()) {
        Commodity element = (Commodity) interator.next();
        if (element.getName().equals(lap.getName()) ) {
            result = true;
        }
    }
    return result;
}

/* public boolean contains_log_and_pass(Commodity lap) {
    boolean result = false;
    Iterator interator = all_goods.iterator();
    while (interator.hasNext()) {
        Commodity element = (Commodity) interator.next();
        if (element.getLogin().equals(lap.getLogin()) &&
            element.getPassword().equals(lap.getPassword()))
    {
        result = true;
    }
    }
    return result;
}*/

public String[] getAllNames() {
    int i = 0;
    String[] allLogins = new String[all_goods.size()];
    Iterator interator = all_goods.iterator();
    // LoginsAndPasswords[] qwe = (LoginsAndPasswords[])
all_log_pass.toArray();
    while (interator.hasNext()) {
        Commodity element = (Commodity) interator.next();
        allLogins[i] = element.getName();
        // allLogins[i] = qwe[i].getLogin();
    /* while ( i < all_log_pass.length) {
        allLogins[i] = all_log_pass[i].getLogin();*/
}

```

```

        i++;
    }
    return allLogins;
}

public String[][] getAllDataFromGoods() {
    int i = 0;
    String[][] allLogins = new
String[all_goods.size()][5]; /*name, price,
utiliti, decrease_coefficient
**/

    Iterator interator = all_goods.iterator();
    // LoginsAndPasswords[] qwe = (LoginsAndPasswords[])
all_log_pass.toArray();
    while (interator.hasNext()) {
        Commodity element = (Commodity) interator.next();
        allLogins[i][0] = element.getName();
        allLogins[i][1] = element.getPrice().toString();
        allLogins[i][2] = element.getUnit();
        allLogins[i][3] = element.getUtility().toString();
        allLogins[i][4] =
element.getDecrease_coefficient().toString();
        // allLogins[i] = qwe[i].getLogin();
        /* while ( i < all_log_pass.length){
            allLogins[i] = all_log_pass[i].getLogin();*/
        i++;
    }
    return allLogins;
}

public int length() {
    int i = 0;
    Iterator interator = all_goods.iterator();
    while (interator.hasNext() && interator.next() != null) {
        i++;
    }
    return i;
}

public void save_goods() {
    try {
        ObjectOutputStream oo = new ObjectOutputStream(new
FileOutputStream("Goods"));
        oo.writeObject(all_goods);
        oo.flush();
        oo.close();
    } catch (Exception ex) {
        System.out.println("There are some problems with writing
to file!" + "\n" + ex);
    }
}

public void load_goods() {
    try {

```

```

        ObjectInput oi = new ObjectInputStream(new
FileInputStream("Goods"));
        all_goods = ((HashSet<Commodity>)
(oi.readObject()));
    } catch (Exception ex) {
        System.out.println("There are problems with reading
from file!" + ex);
    }
}
/**
 * Функция формирования матрицы предпочтений
 * @param col_el количество товаров, которые содержатся в
памяти
 * @return матрица предпочтений клиента
 */

public Double[][] getMatrixUtility(int col_el){
    int col_st = 10;
    Double[][] res = new Double[this.length()][col_st];
    int i;
    int j = 0;
    Iterator interator = all_goods.iterator();
    //interator.hasNext() && interator.next() != null
    while (j<col_el) {
        i=0;
        Commodity element = (Commodity) interator.next();
        while(i<col_st){
            res[j][i]=element.getUtility()/element.getPrice(
)//*10
            -element.getDecrease_coefficient()*i/ele
ment.getPrice();
            i++;
        }
        j++;
    }
    return res;
}

}

package LU_classes;

/**

```

ДОДАТОК Б  
Слайди презентації

**Міністерство освіти і науки України  
Харківський національний університет  
радіоелектроніки**

**Атестаційна робота магістра**

**Дослідження методів формування  
агрегативних моделей штучних агентів**

**Керівник:**

проф. Шостак І.В.

**Виконав:**

ст.гр. ІПЗм-18-2  
Сербін В.В.

2020

1

**Мета дослідження**

**– підвищення ефективності роботи системи  
Інтернет речей шляхом оптимізації формування  
списку товарів для закупівлі.**

В результаті потрібно удосконалити метод формування оптимального списку товарів при закупівлі в автоматизованій системі Інтернет речей шляхом застосування першого і другого законів Госсена до функції закупівлі продуктів за рахунок врахування корисності кожної одиниці товару, що замовляється.

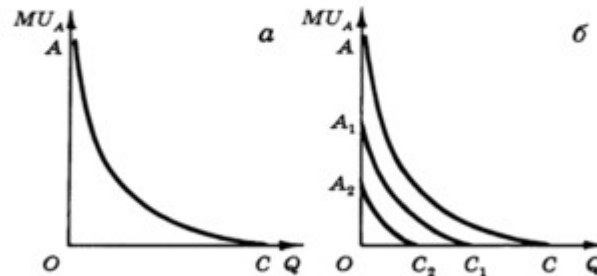
Розроблений прототип ПЗ має бути інтегровано у систему «Розумний будинок».

Прототип ПЗ дозволить задовольнити потреби споживача, враховуючи при цьому природне бажання споживача витратити мінімальну кількість коштів на покупку.

2

## Формування оптимального списку товарів згідно із законами Госсена

Зменшення корисності в одному безперервному акті споживання(а) і при повторних актах споживання(б)



- Ці закони розглядали економічні взаємозв'язки з математичної точки зору.
- Г. Госсен описав правила раціональної поведінки суб'єкта, який прагне витягти максимум користі зі своєї господарської діяльності

3

## Аналіз сучасного стану проблеми «Розумний будинок»

**Комфорт** полягає в постійному контролі й управлінні системами «Розумного будинку» з будь-якої точки будівлі або поза ним через Інтернет.

**Безпека** – засоби безпеки дають звіт про те, що відбувається всередині або біля будинку.

Функціонує імітація присутності господарів,

**Енергозбереження** – системи енергозбереження контролює роботу електрообладнання, запобігають витраті електроенергії даремно шляхом своєчасного відключення невикористаного обладнання, використання сценаріїв освітлення і температурних сценаріїв.

Так само практикується установка систем, що виробляють енергію.

Основною особливістю **інтелектуальної будівлі** є об'єднання окремих підсистем різних виробників в єдиний керований комплекс.

4

## Інтернет речей на прикладі “розумного холодильника”

- **самоконтроль** – у більшості моделей вбудована система сигналізації, яка сканує середу і оповіщає господаря в режимі он-лайн, або чи змінилася температура в морозильній камері або будь-якому відсіку, відключилося електроживлення, залишилася відкритою дверцята, «розумний» холодильник пізнає проблему і відправляє СМС-повідомлення господареві, а також виведе інформацію на екран;
- **контроль вмісту** «Розумного холодильника» і автоматизоване поповнення запасів продуктів. Холодильник сам відправляє господареві список продуктів, які необхідно поповнити згідно заздалегідь введеною програмою або рецептом на вечерю. Господар підтверджує його, вносить коригування, оплачує електронним платежем покупки;
- **забезпечення функціонування бази даних.** Інтернет речей є своєрідною базою даних: в нього можна завантажити рецепти приготування страв, керівництво користувача, нагадування для членів сім'ї, важливі дати, список справ на визначені періоди ;
- **комунікація** з іншими системами «Розумного будинку»;
- **управління системами** «Розумного будинку». Холодильник є пультом управління роботою домашніх приладів;
- **підтримка інтернету** - можливість доступу в інтернет, здійснення покупки в інтернет-магазині, перевірки електронного банківського рахунку та інших функцій
- **перегляд вмісту** холодильника через інтернет;
- **забезпечення роботи функцій** холодильної камери, відтавання інею з випарника, автоматичне доведення незакритих дверцят.

5

## Постановка завдання

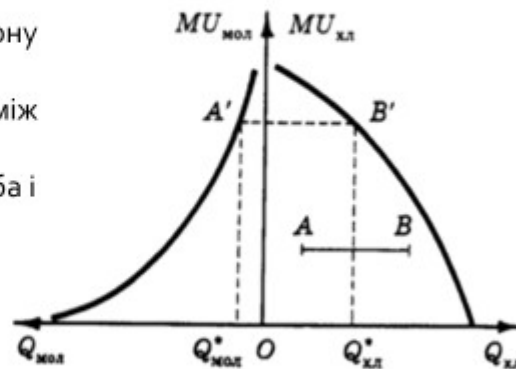
Для даної роботи вибрано завдання контролю вмісту «Розумного холодильника» шляхом закупівлі товарів в інтернет-магазинах, а конкретно – формування оптимального списку товарів для закупівлі.

- У атестаційній роботі пропонується застосування законів Госсена для формування оптимального списку товарів в «Розумному холодильнику».
- Пропоноване рішення має переваги перед існуючими системами, а саме: воно дозволить контролювати кількість грошових коштів на кожну покупку, дозволить формувати список покупок, спираючись на кількість грошових коштів, сформований список покупок буде оптимально задовольняти заданим перевагам користувача системи.

6

## Рекомендація Госсена по оптимізації на прикладі двох продуктів

Графічна ілюстрація закону Госсена, взаємозв'язок між граничною корисністю хліба і молока



7

## Застосування теорії штучних агентів

При побудові штучного агента мінімальний набір базових характеристик включає такі властивості як:

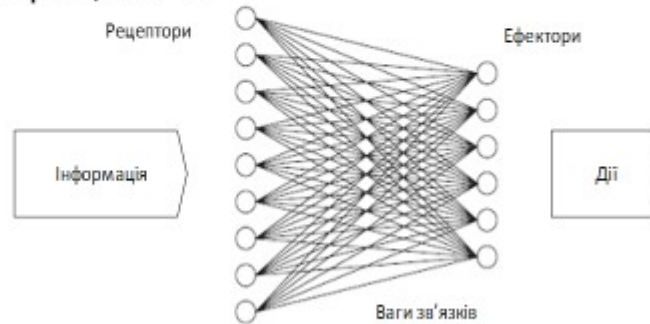
- активність, здатність до організації й реалізації дій;
- реактивність, здатність сприймати стан середовища;
- автономність, відносна незалежність від навколишнього середовища або наявність деякої «волі», що обумовлює власну поведінку, яка повинна мати гарне ресурсне забезпечення;
- товариськість, що впливає з необхідності вирішувати свої завдання разом з іншими агентами й забезпечується розвиненими протоколами комунікації;
- цілеспрямованість, що припускає наявність власних джерел мотивації, а в більш широкому плані, особливих інтенціональних характеристик.

Агент вважається здатним сприймати інформацію із зовнішнього середовища з обмеженим дозволом, обробляти її на основі власних ресурсів, взаємодіяти з іншими агентами й впливати на середовище, переслідуючи свої власні цілі

8

## Геном агента

- Геном  $g$  агента  $A$  являє собою структуру зв'язків нейронної мережі, яку можна представити у вигляді матриці ваг  $W$



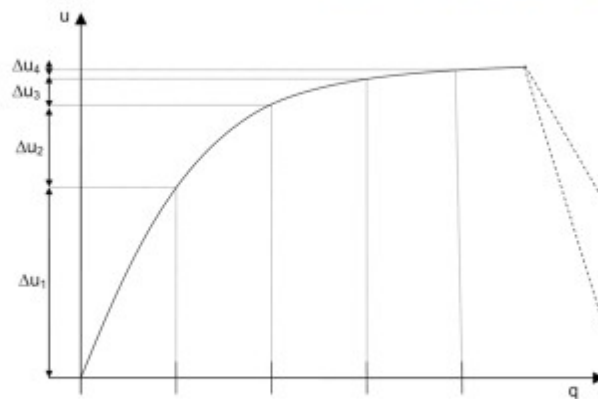
9

Значення вихідних змінних, що визначає дії агента й відповідних коефіцієнтів зміни внутрішнього ресурсу агента  $k_i$ , що використовувалися при моделюванні, наведено в таблиці

Змінна	Значення
$s_1$	$k \cdot r / m \cdot r$ – внутрішній ресурс агента; $r_n$ – номінальний ресурс
$s_2, s_3, s_4, s_5$	$k$ , якщо в даній клітці поля зору є ресурс (0, якщо ні, то)
$s_6, s_7, s_8$	$k$ , якщо в даній клітці поля зору є агент (0, якщо клітка вільна)
$s_9$	$k \cdot r' / m \cdot r'$ – внутрішній ресурс партнера по взаємодії (0, якщо партнер відсутній)
$s_{10}$	$\Delta = \sqrt{\frac{1}{n \cdot m} \cdot \sum_i \sum_j (s_{ij} - \bar{s}_{ij})^2}$ – середня відмінність ваг генома агента й партнера по взаємодії; (0, якщо партнер відсутній)
$s_{11}$	$\bar{\Delta} = \sqrt{\frac{1}{n \cdot m} \cdot \sum_i \sum_j (s_{ij} - \bar{s}_{ij})^2}$ , середня відмінність ваг генома агента й сусідів; (0, якщо сусіди відсутні). $\bar{s}_{ij}$ – центроїд геномів сусідів

10

### Графік залежності граничної корисності товару від кількості спожитих одиниць продукту



Поведінка споживача розглядається як розумна поведінка. Типовий споживач прагне отримати за свої гроші «все, що можна», або, використовуючи спеціальну термінологію, **максимізувати сукупну корисність**

11

### Математична модель формування рекомендованого списку товарів

Формування списку товарів інтерпретується як n-етапний процес прийняття рішень.

На кожному етапі вибирається найкраща одиниця товару для покупки за критеріями ціни та граничної корисності.

Далі ця одиниця додається в список покупок.

$$\sum_{i=1}^n x_i c_i \leq c \quad x_i \geq 0 (i = \overline{1, n}).$$

Така інтерпретація передбачає можливість застосування для вирішення завдань

методу динамічного програмування

12

## Принцип роботи розроблюваного ПЗ

- У процесі роботи ПЗ проводиться зміна суб'єктивної корисності товарів згідно призначенням для користувача коректувань в списках покупок.
- Якщо користувач вносить додаткові одиниці товару в сформований системою список, то корисність цього товару збільшується.
- Прибирає – зменшується.
- Але збільшення або зменшення корисності товару обмежена первинним відношенням користувача до товару, згідно меж зміни корисності

При формуванні матриці переваг вводяться наступні обмеження:

- товари мають тільки одне найменування;
- матрицю корисності формує тільки одна людина;
- коефіцієнт зменшення граничної корисності для  $j$ -го товару постійний.

Для формування оптимального списку товару використовується нормалізована матриця корисності товарів. Нормалізована матриця виходить з матриці переваг шляхом ділення кожного елемента матриці на вартість товару.

13

## Ілюстративна модель формування рекомендованого списку товарів

Ілюстративна модель формування рекомендованого списку товарів покроково і з приведенням проміжних значень має показати роботу алгоритму, описаного Госсеном.

Робота алгоритму розглянута на прикладі.

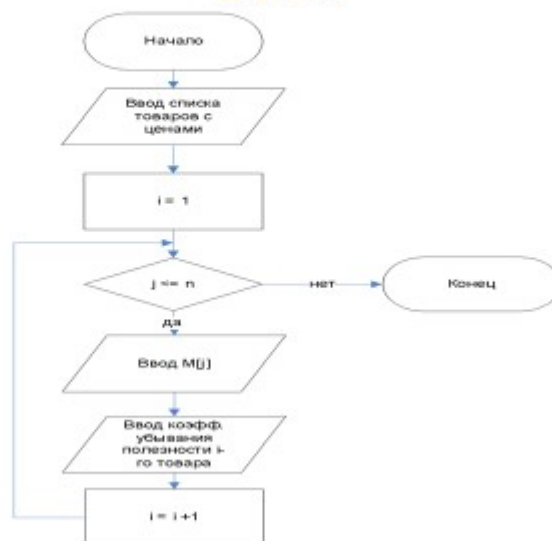
представлені вхідні дані для формування рекомендованого списку товарів.

Використовуючи методику розрахунку рекомендованого списку товарів і вхідні дані сформована матриця корисностей товарів

Негативні елементи матриці переваг знизять корисність сформованого рекомендованого списку товарів, тому вони не будуть обрані, навіть якщо вартість сформованого списку товарів не досягне відведеного кількості грошових коштів на покупки.

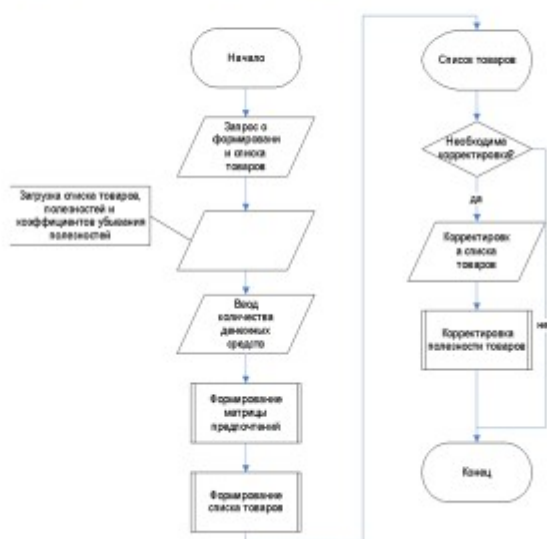
14

## Алгоритмізація методів вибору рекомендованого списку автоматизованої системи



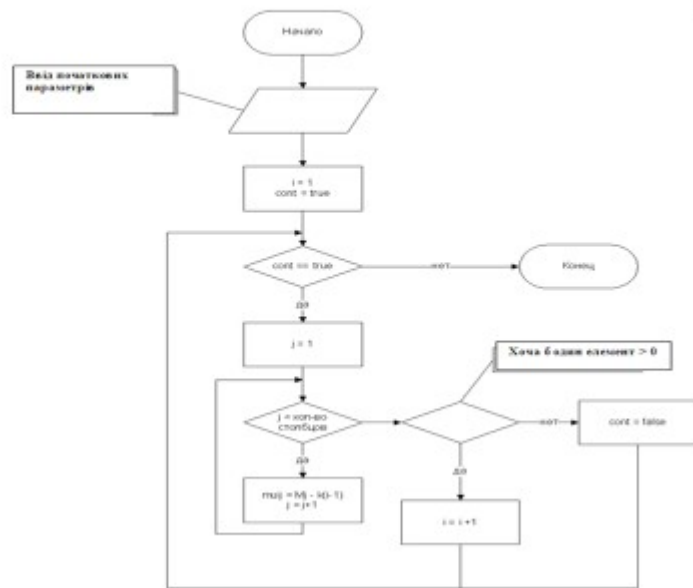
15

## Алгоритм формування списку товарів, що враховує корисності товарів згідно із законами Госсена



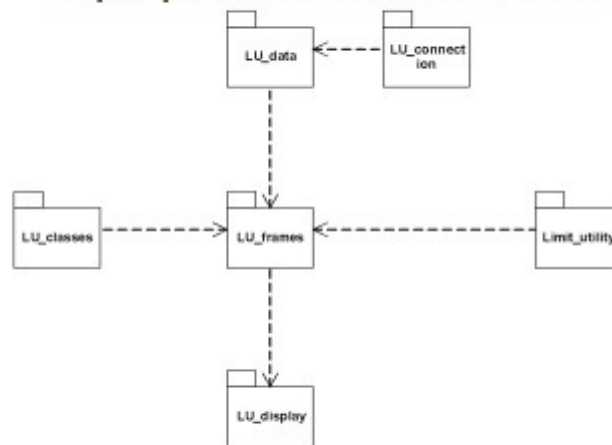
16

## Алгоритм формування матриці переваг



17

## Діаграми варіантів використання програмного забезпечення



Діаграма пакетів ПЗ

18

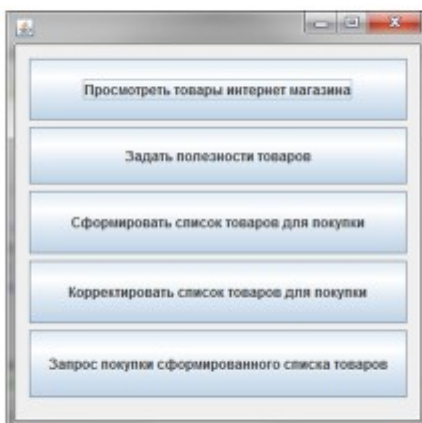
## UML діаграма класів

ПЗ



19

## Інтерфейс користувача програми



20

## Форма перегляду товарів інтернет-магазину

№	Назвами товарів	Цена единицы товара, грн	Едизм.	Мак. кол-во	Полезность	Мак. кол-во ед. произв.
1	Банан	10 кг		0,1	0,2	4
2	Апельсин	12 кг		0,1	0,5	7
3	Яблоко	0 кг		0,1	0,0	10
4	Хлеб	4,75 Колонка		1	0,8	3
5	Колбаса	130 кг		0,1	0,7	3
6	Сыр	60 кг		0,1	0,6	3
7	Томат-паста	12 Банка		1	0,1	1
8	Месо	60 кг		0,1	0,7	1
9	Чай	100 кг		0,1	0,8	2
10	Яйца курные	1 шт		1	0,7	10
11	Резка	13 кг		0,5	0,7	1
12	Рис	15 кг		0,5	0,3	2
13	Макарони	16 кг		0,5	0,3	2
14	Молоко	7 кг		1	0,7	2
15	Тварог	30 кг		0,25	0,6	3
16	Салат	90 кг		0,1	0,0	10
17	Рыбные консервы	8 Банка		1	0,5	6
18	Фрукты сушеные	30 кг		0,1	0,5	6
19	Майонез	18 кг		0,25	0,7	3
20	Соусом	55 кг		0,1	0,7	10

Назад      Сбросить изменения данных      Сохранить      Внести изменения к товару

21

## Форма вывода рекомендованного списка товаров для покупки

№	Название товара	Цена единицы товара	Едизм.	Мак. кол-во	Кол-во для покупки
1	Банан	10 кг		0,1	0,1
2	Апельсин	12 кг		0,1	0,1
3	Яблоко	0 кг		0,1	0,1
4	Хлеб	4,75 Колонка		1	0,2
5	Колбаса	130 кг		0,1	0,1
6	Сыр	60 кг		0,1	0,1
7	Томат-паста	12 Банка		1	0,1
8	Месо	60 кг		0,1	0,1
9	Чай	100 кг		0,1	0,1
10	Яйца курные	1 шт		1	0,1
11	Резка	13 кг		0,5	0,1
12	Рис	15 кг		0,5	0,1
13	Макарони	16 кг		0,5	0,1
14	Молоко	7 кг		1	0,1
15	Тварог	30 кг		0,25	0,1
16	Салат	90 кг		0,1	0,1
17	Рыбные консервы	8 Банка		1	0,1
18	Фрукты сушеные	30 кг		0,1	0,1
19	Майонез	18 кг		0,25	0,1
20	Соусом	55 кг		0,1	0,1

Оплачено средств: 100 грн      Цена покупки составляет: 98,7 грн.  
Остаток: 1,3 грн.  
Полезность составляет: 15,99 единиц

Назад      Сбросить изменения      Готово

22

## Результати програмного експерименту

Основні методів, що застосовано на різних етапах дослідження:

- планування експерименту, основне значення якого – це виділення з усієї сукупності факторів групи істотних факторів, які підлягають детальному вивченню;
- планування експерименту складання планів для об'єктів з якісними факторами;
- планування регресійного експерименту, що дозволяє отримувати регресійні моделі;
- планування екстремального експерименту, в якому головне завдання – експериментальна оптимізація об'єкта дослідження;
- планування при вивченні динамічних процесів.

Таким чином, експеримент має довести оптимальність сформованого списку товарів з використанням законів Госсена, так як будь-які зміни в даному списку призвели до зменшення загальної корисності списку товарів.

**Корисність списку товарів, що отримана при використанні законів Госсена, є максимальною**

23

## Висновки

В наш час «Розумні будинки» є новою технологією, що бурхливо розвивається. Попит на інтелектуальні прилади зростає і аналітики прогнозують цієї галузі масове поширення. Наслідком цього є високий потенційний попит на супутнє ПЗ для «Розумного будинку», чим і є програмне забезпечення, що описується в даній роботі.

В роботі розглянуті основні завдання підсистеми Інтернет речей. Функцію формування оптимального списку товарів при використанні першого і другого законів Госсена запропоновано додати в задачу автоматизованого поповнення запасів продуктів «Розумного холодильника». Проведено огляд існуючих методів формування списку товарів для автоматизованої покупки в системі Інтернет речей.

В роботі представлена математична модель формування оптимального списку товарів.

Розроблено алгоритми, що описують послідовність вироблених системою дій для формування оптимального списку товарів

Наведено структуру взаємодії користувача і системи Інтернет речей, побудована діаграма варіантів використання програмного забезпечення, обрано інструментальне середовище розробки, описаний інтерфейс користувача програми і архітектура ПЗ.

Розроблений прототип ПЗ для формування оптимального списку товарів.

Експериментально доведено оптимальність сформованого списку товарів при використанні законів Госсена для автоматизованої системи Інтернет речей.

Доказом оптимальності стало зменшення загальної корисності покупки при будь-яких змінах кількості продуктів в сформованому списку товарів при використанні законів Госсена.

24

ДОДАТОК В  
Апробація результатів роботи

## Подано тези на Міжнародний молодіжний форум «Радіоелектроніка і молодь в ХХІ сторіччі»

### ДОСЛІДЖЕННЯ МЕТОДІВ ФОРМУВАННЯ АГРЕГАТИВНИХ МОДЕЛЕЙ ШТУЧНИХ АГЕНТІВ

Сербія В.В.

Науковий керівник – проф. Шоста І.В.  
Харківський національний університет радіоелектроніки  
(61166, Харків, пр. Науки, 14, каф. Програмної інженерії,  
тел.: (057) 702-14-46)  
E-mail: i.v.shostak@gmail.com

In the work the method of estimating the maturity (perfection) of the projects performed in the testing of software systems is developed. This method is based on TMM's five-level maturity model. Also, the use of mathematical model, the basic process of testing software data processing systems under the conditions of limited resources, as well as several methods aimed at improving the quality of software products, have been developed and substantiated.

До головних напрямків програмної інженерії відносяться задачі вдосконалення процесів життєвого циклу ПС, зокрема процесу тестування. Неухильне підвищення якості програмних продуктів – основна мета програмної інженерії та предмет турботи розробників ПС.

Відповідь на питання, як підвищити конкурентоспроможність українських програмних продуктів, як знизити ризик проєкту, як досягти балансу сторін трикутника «тривалість – вартість – цілі проєкту», протягом останніх років намагаються знайти відповідь не лише керівники організацій-розробників ПЗ і менеджери проєктів, але і замовники, і споживачі, що використовують програмні продукти низької якості. Наслідки постачання програмних продуктів низької якості – це завжди збитки користувачів: не лише матеріальні і фінансові втрати, але і падіння престижу. Ці проблеми, в свою чергу, негативно відображаються на конкурентоспроможності організації-розробника програмних продуктів. Для забезпечення необхідного рівня якості ПС в міжнародній практиці знаходять застосування два підходи: продукто-орієнтований і процес-орієнтований. В першому акцент робиться на оцінку якості шляхом тестування готового програмного продукту. Цей підхід базується на припущенні, що чим більше знайдено і усунуто дефектів в ПС при тестуванні, тим вище його якість.

Тестування – важливий етап розроблення ПС, оскільки, з одного боку, вимагає значних витрат на проведення, а з іншого – робить великий внесок у його якість. Через теоретично доведену неможливість вичерпного тестування та велику потенційну вартість витрат через відмови у ПС, потрібен чітко визначений та ефективний процес тестування, базований на ухваленні збалансованих рішень щодо тривалості та вартості тестування для досягнення необхідного рівня довіри до якості ПС.

Методи визначення кількісних критеріїв завершення тестування та керування процесом тестування з використанням кількісних вимірів ще мало використовуються в проєктах створення ПС, що призводить до того, що якість та надійність записуються не передбачуваним. Лише за наявності достовірної та своєчасної інформації щодо стану ПС, видів ризиків і можливих витрат через відмови може бути забезпечено ефективне виконання процесу тестування.

Метою роботи є проведення комплексу досліджень з інженерії тестування ПС: оброблення даних, формування ефективної стратегії тестування програмних систем, спрямованої на зниження ризику відмов під час експлуатації. Для цього в роботі розв'язуються наступні задачі:

- дослідження сучасних вигод до процесу тестування ПС;
- аналіз існуючих моделей надійності ПС, розроблення алгоритмів та програм їх реалізації;
- побудова моделі визначення оптимального часу тестування модуля ПС з урахуванням ризиків відмов;
- визначення структури базового процесу, що регламентує всі дії з підготовки, проведення та оцінювання результатів тестування, та розроблення методики виконання процесу тестування ПС оброблення даних;
- проєктування та реалізації програмного комплексу підтримки інженерії тестування;
- аналіз сучасних підходів до визначення рівня зрілості процесу тестування ПС та розроблення методики оцінювання процесу тестування;
- впровадження запропонованих підходів, моделей та методик інженерії тестування в проєкти з розроблення ПС оброблення даних та опис результатів випробування запропонованих моделей оцінювання оптимального часу тестування та методу оцінювання ризиків відмов програмних модулів.

З метою визначення ефективності впровадження базового процесу тестування був розроблений метод оцінювання зрілості (досконалості) виконуваних у проєктах дій з тестування ПС. Цей метод ґрунтується на п'ятирівневій моделі зрілості TMM.

Випробування моделей мають виконуватися в конкретній інформаційно-аналітичній системі підтримки прийняття управлінських рішень, яка має складатися з програмних комплексів, об'єднаних алгоритмом оброблення даних. Для п'яти модулів цього ПК були виконані оцінки ризику відмов та часу тестування.

Отримано результати оцінок параметрів надійності модулів, їх внесків, ризиків відмов, оптимального часу тестування, а також графіки, які відображають поведінку функції корисності тестування.