

ДОДАТОК А
Програмний код

```

package evolution2;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
/**
 *
 * @author
 */
public class Settings extends Frame {
    private static ArrayList<Checkbox> checkboxeseff = new
ArrayList(11);
    private static ArrayList<Checkbox> checkboxesrec = new
ArrayList(12);

    Textfield wight = new Textfield();
    Label widthl = new Label ("Висота поля:");
    Textfield height = new Textfield();
    Label heightl = new Label ("Ширина поля:");
    Textfield foodchance = new Textfield();
    Label foodchancel = new Label ("Імовірність ресурсу:");
    Textfield numagents = new Textfield();
    Label numagentsl = new Label ("Кількість агентів, що
заселяються:");
    Textfield delay = new Textfield();
    Label delayl = new Label ("Затримка, мс:");
    Panel panel = new Panel();
    Button sub = new Button("Установити");
    Gridlayout manager;

    Textfield reprflow = new Textfield();
    Label reprflowl = new Label ("Витрати на розмноження:");
    Textfield reprfailflow = new Textfield();
    Label reprfailflowl = new Label ("Витрати при невдалім
розмноженні:");
    Textfield moveflow = new Textfield();
    Label moveflowl = new Label ("Витрати на переміщення:");
    Textfield movefailflow = new Textfield();
    Label movefailflowl = new Label ("Витрати при невдалому
переміщення:");
    Textfield rotateflow = new Textfield();
    Label rotateflowl = new Label ("Витрати на поворот:");

    Textfield waitflow = new Textfield();
    Label waitflowl = new Label ("Витрати при відпочинку:");
    Textfield eatfailflow = new Textfield();
    Label eatfailflowl = new Label ("Витрати при невдалім
харчуванні:");
    Textfield eatreplenish = new Textfield();
    Label eatreplenishl = new Label ("Ефективність споживання
ресурсу:");
    Textfield reprenergylim = new Textfield();
    Label reprenergyliml = new Label ("Обмеження на розмноження по
ресурсу:");

```

```

Textfield scionenergypart = new Textfield();
Label scionenergypartl = new Label ("Частина ресурсу нащадка:");

Textfield donateflow = new Textfield();
Label donateflowl = new Label (" ресурс, Що Віддається:");
Textfield donatefailflow = new Textfield();
Label donatefailflowl = new Label ("Витрати при невдалій
віддачі:");
Textfield snatchreplenish = new Textfield();
Label snatchreplenishl = new Label (" ресурс, Що
Відбирається:");
Textfield snatchfailflow = new Textfield();
Label snatchfailflowl = new Label ("Витрати при невдалому
відборі:");

Textfield energy = new Textfield();
Label energyl = new Label ("Внутрішній ресурс:");
Textfield direction = new Textfield();
Label directionl = new Label ("Напрямок:");

Textfield ampl = new Textfield();
Label ampll = new Label ("Амплітуда мутацій:");
Textfield startampl = new Textfield();
Label startampll = new Label ("Амплітуда мутацій:");

{
checkboxeseff.add(new Checkbox("Переміщення"));
checkboxeseff.add(new Checkbox("Поворот праворуч"));
checkboxeseff.add(new Checkbox("Поворот ліворуч"));
checkboxeseff.add(new Checkbox("Поглинання ресурсу "));
checkboxeseff.add(new Checkbox("Розмноження"));
checkboxeseff.add(new Checkbox("Очікування"));
checkboxeseff.add(new Checkbox("Віддати ресурс"));
checkboxeseff.add(new Checkbox("Відібрати ресурс"));
checkboxeseff.add(new Checkbox("Нащадок спереду"));
checkboxeseff.add(new Checkbox("Нащадок праворуч"));
checkboxeseff.add(new Checkbox("Нащадок ліворуч"));
checkboxeseff.add(new Checkbox("Нащадок позаду"));

checkboxesrec.add(new Checkbox("Внутрішня енергія"));
checkboxesrec.add(new Checkbox("Ресурс у гнізді"));
checkboxesrec.add(new Checkbox("Ресурс спереду"));
checkboxesrec.add(new Checkbox("Ресурс праворуч"));
checkboxesrec.add(new Checkbox("Ресурс ліворуч"));
checkboxesrec.add(new Checkbox("Агент спереду"));
checkboxesrec.add(new Checkbox("Агент праворуч"));
checkboxesrec.add(new Checkbox("Агент ліворуч"));
checkboxesrec.add(new Checkbox("Енергія мети"));
checkboxesrec.add(new Checkbox("Споріднення мети"));
checkboxesrec.add(new Checkbox("Споріднення сусідів"));
}

Label receptors = new Label("Рецептори");

```

```
Label effectors = new Label("Эффектори");
```

```
Checkbox newform = new Checkbox("Створювати новий вид при заселенні");
```

```
Settings (int mode) {
    switch (mode) {
        case 1 : totalsetmodeone(); break;
        case 2 : totalsetmodetwo(); break;
        case 3 : totalsetmodethree(); break;
    }
}
```

```
Settings (Agent agent, int mode) {
    if (mode == 2)
        agentsetmodetwo(agent);
    else
        agentsetmodeone(agent);
}
```

```
Settings (Agent agent, int i, int o, Outsinapses outsin) {
    double[][] sin = agent.getsin();
    Textfield text = new Textfield();
    Label label = new Label ("Вара зв'язку " + i + " " + o);
    text.setText(sin[i][o] + "");
    sub.addactionlistener((Actionevent event) -> {
        agent.setsin(new Double(text.getText()), i, o);
        outsin.sin.repaint();
        this.dispose();
    });
    panel.setLayout(manager = new GridLayout(3, 1));
    panel.add(label);
    panel.add(text);
    panel.add(sub);
    this.add(panel);
    this.setSize(100, 200);
    this.setVisible(true);
}
```

```
private void totalsetmodeone() {
    panel.setLayout(manager = new GridLayout(23, 2));
    panel.add(width1);
    panel.add(wight);
    wight.setText(Field.getwight() + "");
    panel.add(height1);
    panel.add(height);
    height.setText(Field.getlenght()+ "");
    panel.add(foodchancel);
    panel.add(foodchance);
    foodchance.setText(Field.getfoodchance()+ "");
    panel.add(numagents1);
    panel.add(numagents);
    numagents.setText(Field.getnumagents()+ "");
}
```

```

panel.add(delay1);
panel.add(delay);
delay.setText(Field.getdelay()+ "");

panel.add(amp11);
panel.add(amp1);
amp1.setText(Lifeform.getamp1()+ "");
panel.add(energy1);
panel.add(energy);
energy.setText(Lifeform.getenergy()+ "");

panel.add(reprflow1);
panel.add(reprflow);
reprflow.setText(Lifeform.getreprflow()+ "");
panel.add(reprfailflow1);
panel.add(reprfailflow);
reprfailflow.setText(Lifeform.getreprfailflow()+ "");
panel.add(moveflow1);
panel.add(moveflow);
moveflow.setText(Lifeform.getmoveflow()+ "");
panel.add(movefailflow1);
panel.add(movefailflow);
movefailflow.setText(Lifeform.getmovefailflow()+ "");
panel.add(rotateflow1);
panel.add(rotateflow);
rotateflow.setText(Lifeform.getrotateflow()+ "");

newform.setState(Lifeform.getnewform());
this.add(panel);
this.setSize(550, 500);
this.setVisible(true);
sub.addActionListener((ActionEvent event) -> {
    if (!wight.getText().equals(""))
        Field.setup("wight", wight.getText());
    if (!height.getText().equals(""))
        Field.setup("length", height.getText());
    if (!foodchance.getText().equals(""))
        Field.setfoodchance(new
Double(foodchance.getText()));
    if (!numagents.getText().equals(""))
        Field.setnumagents(Integer.parseInt(numagents.getText()));

    if (!delay.getText().equals(""))
        Field.setdelay(Integer.parseInt(delay.getText()));

    if (!energy.getText().equals(""))
        Lifeform.setenergy(new Double(energy.getText()));
    if (!reprflow.getText().equals(""))
        Lifeform.setreprflow(Integer.parseInt(reprflow.getText()));
    if (!reprfailflow.getText().equals(""))
        Lifeform.setreprfailflow(Integer.parseInt(reprfailflow.getText()));
});

```

```

        if (!moveflow.getText().equals(""))
Lifeform.setmoveflow(Integer.parseInt(moveflow.getText()));

Lifeform.setsnatchfailflow(Integer.parseInt(snatchfailflow.getText()
));
        if (!startampl.getText().equals(""))
            Lifeform.setstartampl(new
Double(startampl.getText()));
        Lifeform.setnewform(newform.getState());

        this.dispose();
    });
}

private void totalsetmodetwo() {
    for (int i = 0; i < 11; i++)
        if (Lifeform.getinavail()[i] == 1)
            checkboxesrec.get(i).setState(true);
    for (int i = 0; i < 12; i++)
        if (Lifeform.getoutavail()[i] == 1)
            checkboxeseff.get(i).setState(true);
    panel.setLayout(manager = new GridLayout(13, 2));
    panel.add(effectors);
    panel.add(receptors);
    for (int i = 0; i < 11; i++) {
        panel.add(checkboxeseff.get(i));
        panel.add(checkboxesrec.get(i));
    }
    panel.add(checkboxeseff.get(11));
    panel.add(sub);
    this.add(panel);
    this.setSize(320, 400);
    this.setVisible(true);

    sub.addActionListener((ActionEvent event) -> {
        for (int i = 0; i < 11; i++) {
            if (checkboxesrec.get(i).getState())
                Lifeform.getInavail()[i] = 1;
            else
                Lifeform.getInavail()[i] = 0;
        }
        for (int i = 0; i < 12; i++) {
            if (checkboxeseff.get(i).getState())
                Lifeform.getOutavail()[i] = 1;
            else
                Lifeform.getOutavail()[i] = 0;
        }
        this.dispose();
    });
}
}
}

```

ДОДАТОК Б
Слайди презентації



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

**Атестаційна робота
Освітньо-науковий рівень - магістр**

**ДОСЛІДЖЕННЯ МЕТОДІВ ПОБУДОВИ
ІНТЕРНЕТ-МАГАЗИНІВ З ІНТЕЛЕКТУАЛЬНОЮ
ПІДТРИМКОЮ ВИБОРУ ТОВАРІВ**

Керівник:

проф. Шостак І.В.

Виконав:

ст.гр. ІПЗм-18-4
Швечіков І.С.

2020

1 1



Мета роботи

Кожний електронний магазин прагне збільшити кількість відвідувачів, організувати торгівлю максимально ефективно, щоб кількість продажів збільшувалася, для цього застосовуються останні розробки в сфері інтернет технологій.

Метою атестаційної роботи є використання агрегатних моделей в організації й функціонуванню інтернет-магазинів та розробка прототипу ПЗ для перевірки запропонованих алгоритмів.

2

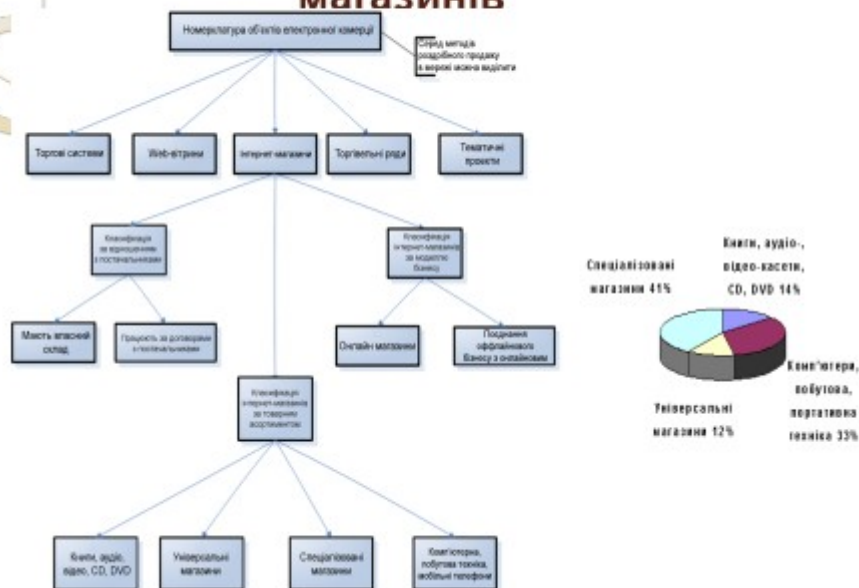
Актуальність

- 2 місяці карантинних обмежень краще усяких слів показав актуальність теми, бо закриття традиційних магазинів дало поштовх для переходу усіх торгівельних мереж та малих магазинів в он-лайн
- зростання інтернет-торгівлі, поява нових інтернет-магазинів спричиняє боротьбу за покупця, за кількість продаваного товару й за престиж компанії.
- кожний електронний магазин прагне збільшити кількість відвідувачів, організувати торгівлю максимально ефективно, щоб кількість продажів збільшувалася, для цього застосовуються останні розробки в сфері інтернет-технологій.

Об'єктом дослідження атестаційної роботи магістра є процеси функціонування систем електронної комерції.

3

Класифікація інтернет-магазинів



4



Аналіз використання агентних систем

Завданнями, що мають виконуватися подібними системами, мають бути автоматичне реагування на вхідну інформацію - це може бути задоволення заявок, що надходять на склад, обробка запитів клієнтів електронних магазинів і багато іншого.

Під агентом розуміється самостійна програмна система можливість, що має, ухвалювати вплив із зовнішнього миру, визначати свою реакцію на цей вплив і здійснювати цю реакцію.

Під зовнішнім миром тут розуміється середовище, що оточує агента.

Класифікуються агенти на **чотири основні типи**:

- *прості*,
- *розумні (smart)*,
- *інтелектуальні (intelligent)*
- *дійсно інтелектуальні (truly intelligent)*.

7



Властивості агентів

автономність – агенти функціонують без прямого втручання людей або кого-небудь іншого й володіють певною здатністю контролювати свої дії й внутрішній стан;

методи (способи) спілкування – агенти взаємодіють із іншими агентами засобами деякої комунікаційної мови;

реактивність – агенти здатні сприймати навколишнє середовище і адекватно реагувати в певних тимчасових рамках на зміни, які відбуваються;

активність – агенти не просто реагують на зміни середовища, але й мають цілеспрямовану поведінку і здатність проявляти ініціативу;

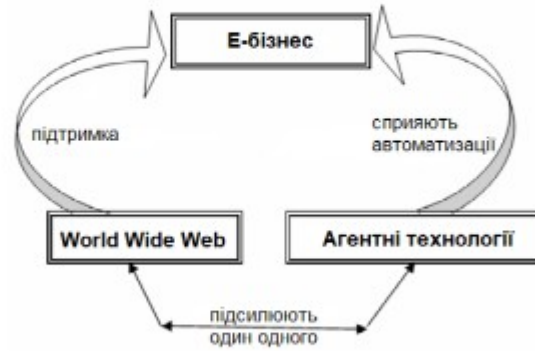
індивідуальна картина миру – кожний агент має власну модель навколишнього йому середовища - агент будує свою модель миру на основі інформації, яку одержує із зовнішнього середовища;

комунікабельність і кооперативність – агенти можуть обмінюватися інформацією з навколишнім середовищем і іншими агентами, можливість комунікацій з іншими агентами є обов'язковою умовою спільних дій для досягнення цілей;

інтелектуальна поведінка – поведінка агента включає здатність до навчання, логічної дедукції або конструювання моделі навколишнього середовища для того, щоб знаходити оптимальні способи поведінки.

8

Агентно-орієнтовані технології для інтелектуалізації завдань е-бізнесу



9

Діаграма варіантів використання для відвідувача сайту (фрагмент)



10



Моделювання агрегатних моделей бізнес-процесів

- Агрегат – абстрактна схема функціонування складної системи.
- Принципово загальна схема в стані замінити всі частки. Однак вона складніше й для окремих випадків її застосування приводить до значного ускладнення обчислень.
- Проте завдання, що існують, для яких приватні схеми не працюють або працюють погано.
- У цих випадках доводиться користуватися більш загальною схемою, поки не розроблена більш проста схема – приватна.
- Головним стимулом застосування загальних схем виявляється універсалізація математичного опису й операторів моделювання алгоритму для всіх елементів складної системи.

11



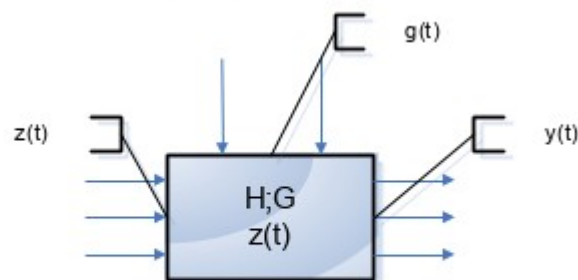
Агрегат

Агрегат має особливий вхідний полюс, до якого надходять у момент часу t керуючі сигнали.

Керуючий сигнал g , є елементом множини G .

На виході агрегату утворюються вихідні сигнали.

Вихідний сигнал u є елементом деякої множини Y , і визначається по станах агрегату $z(t)$ за допомогою оператора G

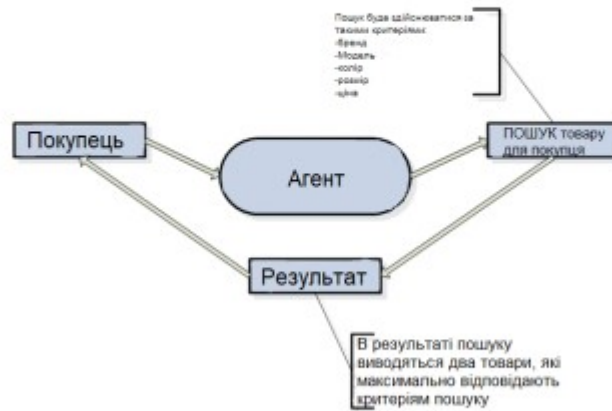


12



Моделі агентів пошуку

Модель агента пошуку товару



13



Моделі агентів пошуку

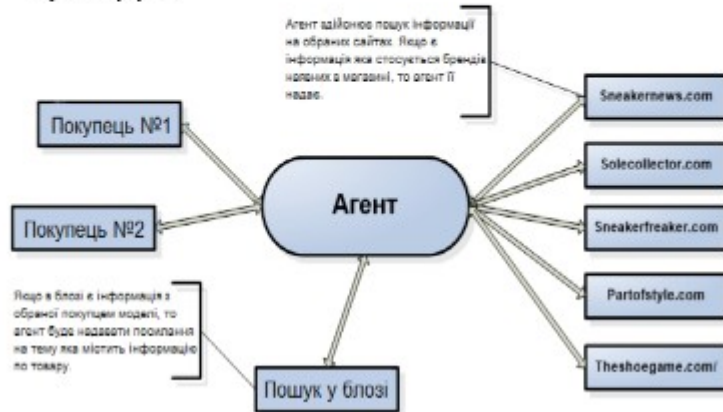
Модель агента пошуку серед брендів



14

Моделі агентів пошуку

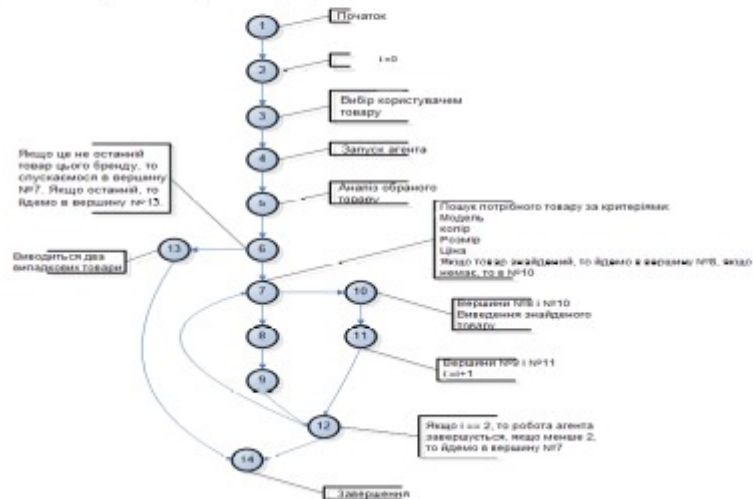
Модель агента пошуку інформації трендів



15

Модель бізнес-процесів електронного магазину

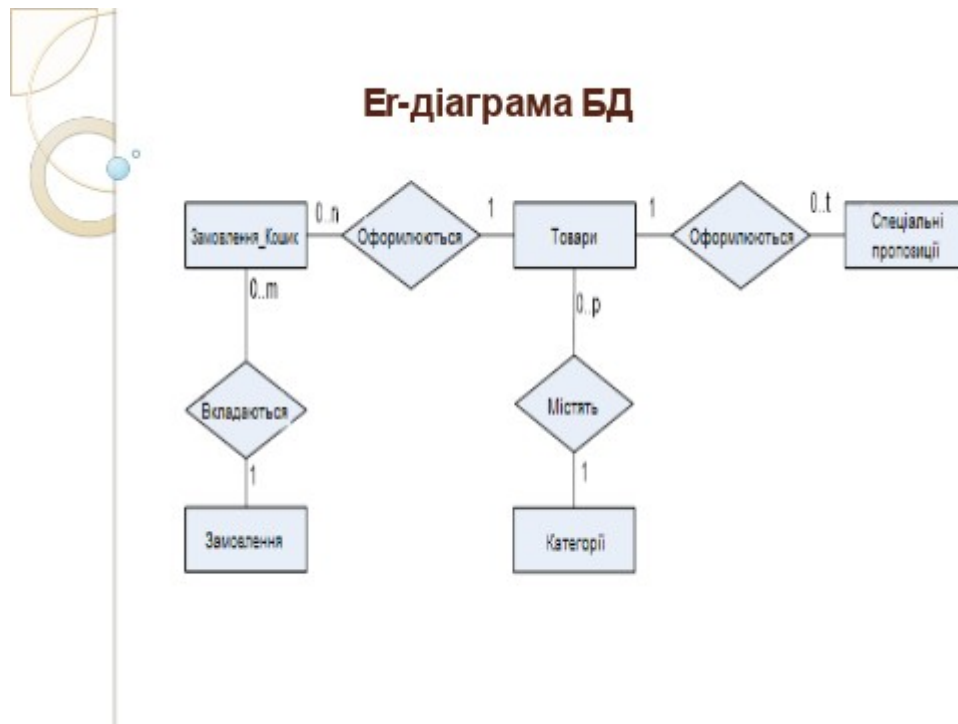
модель агента пошуку необхідного товару усередині бренду зображена у вигляді графа



16

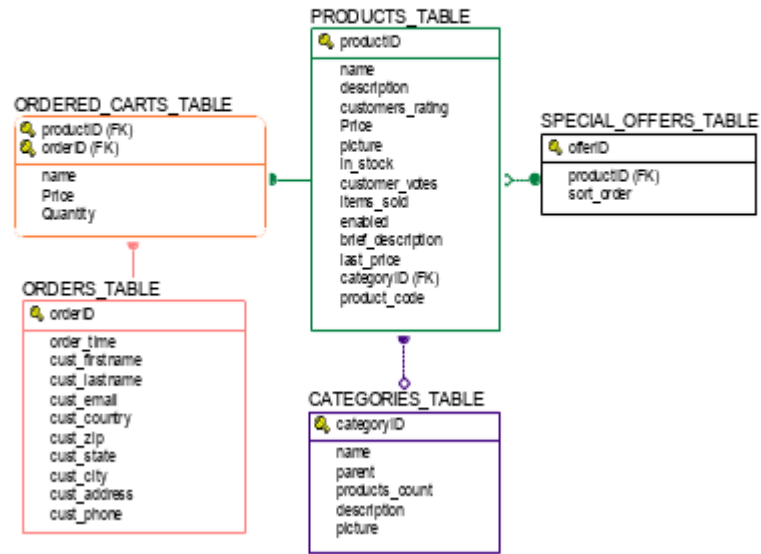


17



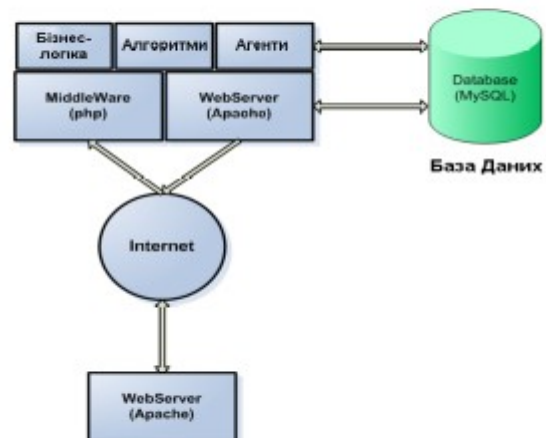
18

Логічна модель БД інтернет-магазину



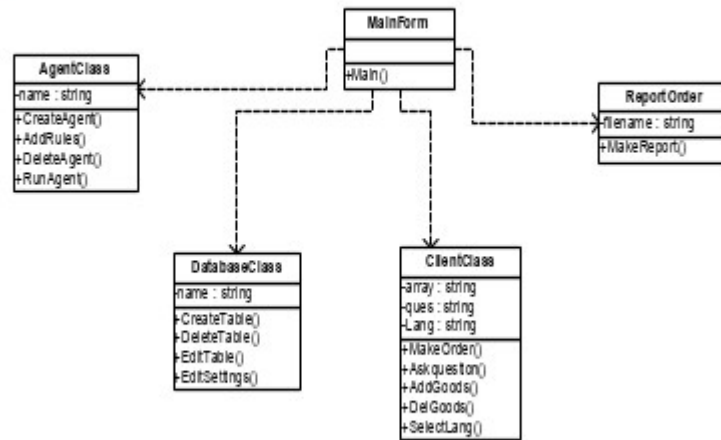
19

Архітектура ПЗ агентного типу



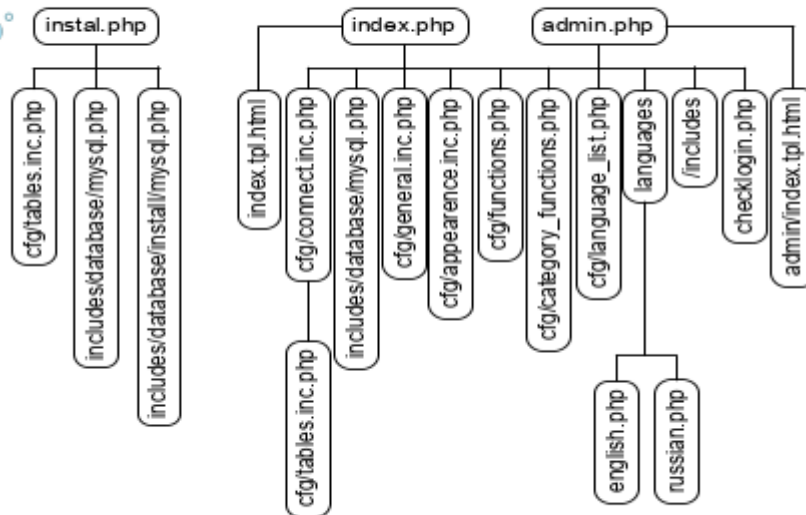
20

Діаграма класів



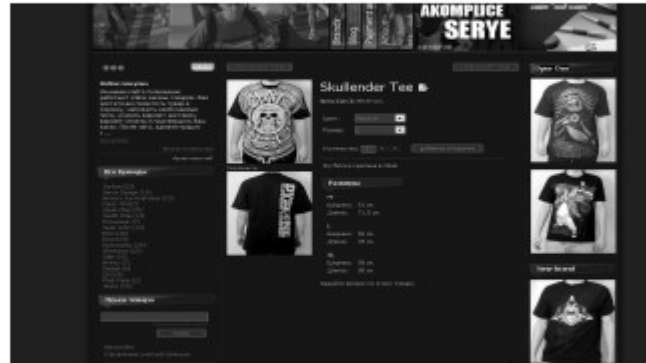
21

Структурна схема ПЗ



22

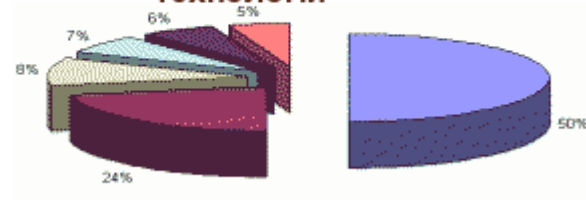
Екранна форма інтернет-магазину Originalstore



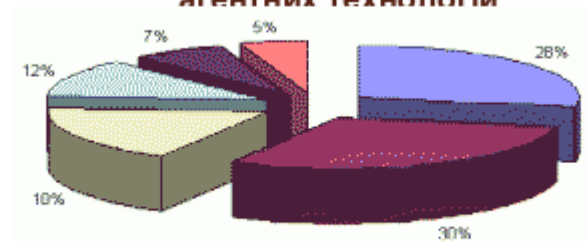
екранна форма роботи агентів розміщених на сайті <http://originalstore.com.ua/>.

23

Статистика глибини відвідування сайту до використання агентних технологій



Статистика глибини відвідування сайту після використання агентних технологій



1 сторінка	2-3 сторінки	4-5 сторінки
6-7 сторінки	8-9 сторінки	10-15 сторінки

24



Висновки

В результаті виконання атестаційної роботи розроблено:

Удосконалено алгоритми організації й функціонування інтернет-магазину, шляхом застосування агентних технологій при розробці програмних засобів у процесах продажу товарів.

- Проаналізовані та удосконалені найбільш важливі алгоритми, розроблені для рішення поставленого завдання:
- Розроблені алгоритми можуть бути використані при проектуванні програмного забезпечення інтернет-магазинів
- Розроблений прототип ПЗ був протестований на предмет працездатності й рішення поставлених завдань.
- Тестування проводилося на контрольних прикладах, у результаті чого ПЗ показало коректну роботу.

ДОДАТОК В
Апробація результатів роботи

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

ДОСЛІДЖЕННЯ МЕТОДІВ ПОБУДОВИ ІНТЕРНЕТ-МАГАЗИНІВ З
 ІНТЕЛЕКТУАЛЬНОЮ ПІДТРИМКОЮ ВИБОРУ ТОВАРІВ

Швечков І.С.

Науковий керівник – проф. Шубін І.Ю.
 Харківський національний університет радіоелектроніки
 (61166, Харків, пр. Науки, 14, каф. Програмної інженерії,
 тел.: (057) 702-14-46)
 E-mail: ilia.shvchikov@nuire.ua

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

Виповізь на питання, як підвищити конкурентоспроможність українських програмних продуктів, як знизити ризик проєктів, як досягти балансу сторін трикутника «тривалість – вартість – цілі проєкту», протягом останніх років намагалося знайти виповізь не лише керівники організацій-розробників ПС і менеджери проєктів, але і замовники, і споживачі, що використовують програмні продукти низької якості. Наслідки поставання програмних продуктів низької якості – не 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 їхню якість. Через теоретично доведену неможливість вичерпного тестування та велику потенційну вартість втрат через відмови у ПС, потрібні чітко визначені 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 вибірків ще мало використовуються в проєктах створення ПС, що призводить до того, що якість та надійність залишаються не передбачуваними. Лише за наявності достовірної та своєчасної інформації щодо стану ПС, видів ризиків і можливих втрат через відмови може бути забезпечене ефективне виконання процесу тестування.

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

- дослідження сучасних вимог до процесу тестування ПС;
- аналіз існуючих моделей надійності ПС, розроблення алгоритмів та програм їх реалізації;
- побудова моделі визначення оптимального часу тестування модулів 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 опису 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 дієвим процесом оброблення даних. В ході практичної реалізації аналіз загроз відмов системи показав, що з позицій цілісності інформації найбільший внесок у ризик її відмов робить ПК контролю і введення даних до БД ORACLE, який встановлюється та одночасно функціонує на 10 робочих місцях. Для п'яти модулів цього ПК були виконані оцінки ризику відмов та часу тестування.