

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій

Кафедра комп'ютерних наук

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з курсу

“ЛЮДИНО-МАШИННА ВЗАЄМОДІЯ”

ТЕРНОПІЛЬ – 2022

Конспект лекцій із дисципліни «Людино-машинна взаємодія» для здобувачів освітнього рівня спеціальності 121 – Інженерія програмного забезпечення галузі знань 12 – Інформаційні технології // Є.О.Марценюк, Л.І. Гончар – Тернопіль: ПП «Шпак» 2022. – 49с.

Укладачі: Марценюк Євгенія Олексіївна, к.т.н., доцент кафедри комп'ютерних наук ЗУНУ

Гончар Людмила Іванівна, к.е.н., доцент кафедри комп'ютерних наук ЗУНУ

Відповідальний за випуск: Пукас Андрій Васильович, к.т.н., доцент, завідувач кафедри комп'ютерних наук ЗУНУ

*Конспект лекцій розглянуто та рекомендовано до друку
на засіданні кафедри комп'ютерних наук ЗУНУ
(протокол № 2 від 14 вересня 2022 року)*

*Конспект лекцій затверджено на засіданні групи забезпечення спеціальності
«Інженерія програмного забезпечення»
(протокол № 2 від 14 вересня 2022 року)*

© Марценюк Є. О.,
Гончар Л. І., 2020

ЗМІСТ

ВСТУП.....	4
Лабораторна робота № 1	6
Лабораторна робота № 2	10
Лабораторна робота № 3	13
Лабораторна робота № 4	18
Лабораторна робота № 5	21
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	25
Додаток А.....	26

ВСТУП

Навчальна дисципліна «Людино-машинна взаємодія» є нормативною і входить до циклу теоретичної і практичної підготовки бакалавра за спеціальністю 121 «Інженерія програмного забезпечення».

Мета навчальної дисципліни – надати в повному обсязі систематизовані знання про сучасні концепції і напрямки розвитку комп'ютерних технологій проектування широкого кола програмних інтерфейсів.

Завдання навчальної дисципліни:

- 1) довести психологічні принципи людино-машинної взаємодії;
- 2) з'ясувати ергономічні аспекти якості призначеного для користувача інтерфейсу;
- 3) ознайомити студентів з методологіями та технологіями проектування досвіду взаємодії;
- 4) визначити етапи проектування призначених для користувача інтерфейсів;
- 5) виявити стандарти, що забезпечують взаємодію користувача з операційним середовищем;

Структурно навчальний посібник складається з лабораторного практикуму, що містить опис дев'яти лабораторних робіт, і теоретичних відомостей, які розташовані в логічній послідовності відповідно до порядку виконання лабораторних робіт і робочої навчальної програми з предмета «Людино-машинна взаємодія».

У кожній лабораторній роботі наведено мету, перелік знань і вмінь, які будуть сформовані у студентів унаслідок виконання лабораторної роботи, а також докладний опис практичних завдань до лабораторної роботи. Завдання до лабораторних робіт відповідають практичним завданням, які виникають у процесі аналізу вимог, проектування, розроблення і використання призначеного для користувача інтерфейсу.

У теоретичних відомостях пояснено ключові терміни, наведено приклади виконання завдань до відповідної лабораторної роботи та ілюстративний матеріал, який полегшує розуміння порядку виконання лабораторної роботи.

У посібнику подано довідкові матеріали, які містять інформацію, необхідну для вирішення практичних завдань, пов'язаних з проектуванням та

оцінюванням зручності використання призначеного для користувача інтерфейсу.

Навчальний посібник допоможе майбутнім розробникам програмного забезпечення оволодіти практичними навичками використання принципів людино-машинної взаємодії при розробленні та оцінюванні зручності використання призначених для користувача інтерфейсів програмних систем

Лабораторна робота №1

Тема Дослідження інтерфейсу мобільного телефону з використанням евристичних оцінок Джекоба Нільсена.

Мета Дослідити інтерфейсу мобільного телефону з використанням евристичних оцінок Джекоба Нільсена.

Теоретичні відомості

Основні принципи евристичної оцінки інтерфейсу системи визначені Джекобом Нільсеном у 1994 році, які ще іноді називають евристичними Нільсена, дають змогу визначити наскільки привабливим та функціональним є інтерфейс.

Прозорість стану системи

Система завжди повинна інформувати користувача про ті процеси, які відбуваються, використовуючи зворотній зв'язок в межах певного часового інтервалу.

Відповідність між системою та реальним оточенням

Система повинна використовувати для спілкування з користувачем слова, фрази та поняття добре знайомі користувачу, замість системно-орієнтованих термінів. Потрібно дотримуватися природного та логічного порядку подачі інформації, який відповідає аналогічним поняттям оточуючого світу.

Користувацький контроль та свобода

Користувачі часто помилково обирають ту чи іншу функцію, внаслідок чого виникає потреба чіткого позначення “екстреного виходу”, який надав би змогу залишити небажане місце, без зайвих процедур у формі розширених діалогів. Підтримка “undo” та “redo”.

Послідовність та стандарти.

Недоцільно використовувати різні слова, стани або дії для позначення однакових речей.

Потрібно дотримуватись конвенцій даної платформи.

Запобігання помилкам.

Обміркований дизайн, що здатний запобігти помилкам завжди кращий за гарне повідомлення про помилку. Потрібно виключити передумови, що сприяють появі помилок або у разі появи таких передумов проінформувати про це

користувача, використовуючи опцію підтвердження перед безпосереднім виконанням дії.

Впізнавання замість згадування.

Мінімізувати навантаження на пам'ять користувача, зробивши об'єкти, дії та опції очевидними. Користувач не повинен пам'ятати інформацію з однієї частини діалогу до наступної. Інструкції щодо використання системи повинні бути чіткими та зрозумілими та легко досяжними у разі потреби.

Гнучкість та ефективність використання

Прискорювачі – невидимі для новачків – можуть прискорювати роботу експертів. Таким чином система може обслуговувати користувачів з різним рівнем досвіду. Надайте користувачу можливість налаштувати систему відповідно до його потреб.

Естетичний та мінімалістичний дизайн

Діалоги не повинні містити непотрібну або рідко використовувану інформацію. Кожний додатковий блок інформації конкурує з відповідними блоками інформації, і, тим самим, зменшує видимість.

Допоможіть користувачу визначити, діагностувати та виправити помилки

Повідомлення про помилку повинні бути написані простою мовою (без кодів), точно визначати проблему і пропонувати конструктивні шляхи її вирішення.

Системи допомоги та документація

Звичайно добре, якщо систему можна використовувати без документації, проте необхідно забезпечити допомогу та документацію. Така інформація повинна бути легко досяжною, відповідати задачам користувача, містити перелік визначених кроків, які необхідно виконати і не бути занадто об'ємною.

Порядок виконання роботи

1.1. Підрахуйте, скільки часу займає відправка SMS на вашому телефоні (саме повідомлення відправляти не обов'язково – головне виконати всю послідовність операцій). Початкова умова – телефон заблокований.

Приклад запису результатів:

Опис операції	Кількість операцій
Знімаю блокування.	2
Заходжу в меню.	1
Заходжу в меню Messaging:	3
Обираю опцію Create:	1
Обираю Text Only (тип SMS):	1
Пишу текст повідомлення:	Операції підраховувати не треба
Завершую введення тексту повідомлення:	0
Обираю номер:	Операції підраховувати не треба
Обираю Send – Відправляю:	2
Всього: (без врахування набору тексту і вибору із телефонної книги)	10

1.2. Підрахуйте скільки часу займає відправка одного SMS на вашому телефоні. Завдання потрібно виконати п'ять разів. Результати виконання внести до таблиці 2.

Повідомлення – **Лекція з курсу “Людино-машинна взаємодія відбудеться у четвер о 9.35.**

Якщо, виконуючи операції, довелося виправляти будь-які дії (наприклад, відкрилося не те меню, або був набраний помилковий символ і довелося виправляти текст) поряд з часом напишіть, що виникла помилка.

Номер повідомлення	Час (в секундах)	Виникла помилка
1	135	Так
2	123	Так
3	124	Так
4	122	Так
5	123	Так

1.3. Підрахуйте, скільки операцій у вашому мобільному телефоні треба зробити, щоб встановити будильник.

1.4. Використовуючи наведені у теоретичних відомостях 10 евристик Нільсена виконайте аналіз інтерфейсу Вашого телефону. У звіті необхідно подати детальний опис того, яким чином виконується або не виконується вказане правило і навести приклади.

1.5 Порівняйте інтерфейс вашого мобільного телефону з будь-якою моделлю іншої фірми виробника. Які переваги та недоліки інтерфейсу даної моделі ви виявили?

3. Зміст звіту

2.1. Тема роботи.

2.2. Результати виконання роботи (пункти 1.1 – 1.5).

2.3. Висновки.

Лабораторна робота №2

Тема: Дослідження характеристик людини-оператора

Мета: Розробити математичну модель людини-оператора, оцінити ефективність діяльності оператора

ТЕОРЕТИЧНІ ВІДОМОСТІ

Існує декілька методів апріорного аналізу діяльності людини: узагальнено-структурний, операційно-психологічний, метод статистичного еталона, логіко-інформаційний і т.д. Використовуються також три типи моделей поведінки людини: когнітивні моделі, моделі теорії керування, мережні моделі задач. В даному пункті розглядається один з можливих аналітичних методів – метод математичного моделювання діяльності.

В основу такого моделювання покладена ідея використання методів Монте-Карло для імітації імовірісно-часових характеристик діяльності операторів.

В загальному випадку час виконання окремої операції складається з двох складових:

$$t = t_{o.d.} + t_p. \quad (1)$$

де $t_{o.d.}$ – час основної діяльності оператора всередині операції; $t_p.$ – час резерву всередині операції.

Час резерву не є показником яких-небудь індивідуальних якостей оператора і характеризує конкретну технічну реалізацію системи "людина–машина, виробничу обстановку в зоні виконання роботи і т.д. Тому можна вважати, що $t_p. = \text{const}$ для кожної операції.

Час основної діяльності $t_{o.d.}$ навпаки характеризує індивідуальні якості операторів, міру їх тренованості і т.д. і розподілений по усіченому нормальному закону.

Весь вибір операцій ранжується на операції істотні і неістотні. Істотна операція – операція, невиконання якої приводить до зриву всієї задачі. Операція, невиконання або пропуск якої не приводить до зриву задачі, а лише погіршує кінцевий ефект, наприклад точність, називається неістотною.

Різниця між часом, відведеним на виконання задачі, і необхідним – зовнішній резерв, або дефіцит, часу. Сумарний час всіх неістотних операцій є внутрішнім резервом. В дефіциті часу оператор може жертвувати найближчими, в порядку проходження, неістотними операціями. Якщо упущений час

надолужується, оператор приступає до виконання всіх подальших операцій без виключення.

В процесі моделювання діяльності для кожної операції визначається коефіцієнт часової напруженості, який кількісно виражається

$$S_{ij} = \frac{\sum_{l=i+1}^n t_{lj} | B_l = 1}{t_u - t_{замр.}} \quad (2)$$

S_{ij} – коефіцієнт часової напруженості i операції всередині j реалізації; чисельник – час, необхідний на виконання істотних операцій, що залишилися (після i); B – індекс істотності операції ($B=1$ – операція істотна, $B=0$ – неістотна); t_u – час, відведений на виконання всієї задачі; $t_{замр.}$ – час, реально затрачений на виконання операцій до $(i+1)$ в j реалізації; n – загальна кількість операцій в задачі; i – номер модельованої операції; j – номер реалізації алгоритму.

До певної межі (для пересічного оператора $S_{ij \text{ перед}} = M_0 = 2.3$) час виконання окремої операції і його розкид лінійно скорочуються – ефект так званої інтенсифікації внутрішніх резервів оператора.

Із зростанням коефіцієнтів часової напруженості імовірність безпомилкового виконання операції падає від значення \bar{p}_i (задається як вхідний параметр) до величини

$$p_{ij} = \bar{p}_i - \frac{\bar{p}_i(S_{ij} - 1)}{M_0 - 1} \quad \text{при } 1 \leq S_{ij} \leq M_0 \quad (3)$$

де \bar{p}_i – імовірність безпомилкового виконання операції в нормальних умовах.

Експериментальні дослідження діяльності оператора показали, що якщо оператор жорстко не був лімітований інструкцією, то до виконання неістотної операції він приступає залежно від часу, що залишився у нього. Імовірність того, що оператор приступить до виконання неістотної операції, залежно від витраченого часу має вид:

$$p_n = \exp(-\lambda_i t_{n1} | B_i = 0) \quad (4)$$

Можна скласти матрицю несумісних подій:

$$\begin{array}{c|cccc} x_{11} & \dots & x_{1i} & \dots & x_{1n} & F_1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{j1} & \dots & x_{ji} & \dots & x_{jn} & F_j \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{m1} & \dots & x_{mi} & \dots & x_{mn} & F_m \end{array} \quad (5)$$

де x_{ji} – стан i операції в j реалізації, 1 – безпомилкове виконання, 0 – виконання з помилкою або пропуск операції; F_j – ефективність в j реалізації.

ХІД РОБОТИ

1. Отримати у викладача індивідуальні дані для виконання лабораторної роботи: час основної діяльності всередині операцій, час резерву виконання операцій, ефективність комбінацій виконань операцій.
2. Побудувати для кожної операції закон розподілу значення часу її виконання на основі усіченого нормального закону, часу основної діяльності і часу резерву.
3. Моделювати час виконання оператором операцій та їх успішність за методом Монте-Карло і законів розподілу випадкових величин.
4. Після кожної ітерації моделювання визначити комбінацію, що їй відповідає і її ефективність.
5. Оцінити імовірності комбінацій на основі даних моделювання.
6. Обчислити середнє значення ефективності діяльності оператора.
7. Зробити висновки про виконання роботи.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Методи апріорного аналізу діяльності людини-оператора.
2. Моделюванні часу виконання операції оператором.
3. Коефіцієнт часової напруженості.
4. Імовірність безпомилкового виконання операції оператором.
5. Імовірність початку виконання неістотної операції оператором.
6. Матриця несумісних подій в моделюванні діяльності оператора.

Лабораторна робота №3

Тема: Проектування інтерфейсу користувача

Мета: Розробити проект програмного інтерфейсу користувача, оцінити інтерфейси користувача відомих програм

ТЕОРЕТИЧНІ ВІДОМОСТІ

Правила проектування інтерфейсу користувача:

1. Керування системою користувачем

Одне з основних правил проектування – дати користувачу контроль над системою. Це правило при виконанні наступних принципів:

1. Використовуйте режими розсудливо.
2. Надайте користувачу можливість вибирати: працювати з мишкою, клавіатурою або їх комбінацією (гнучкість).
3. Дозвольте користувачу сфокусувати увагу (переривання).
4. Демонструйте повідомлення, які допоможуть користувачу в роботі (корисність).
5. Створіть умови для негайних і зворотних дій та зворотного зв'язку (поблажливість).
6. Забезпечте відповідні шляхи і виходи (здатність орієнтування).
7. Пристосовуйте систему до користувачів з різним рівнем підготовки (доступність).
8. Зробіть користувацький інтерфейс більш зрозумілим (легкість в користуванні).
9. Дайте користувачу можливість налаштовувати інтерфейс на свій смак (пристосовуваність).
10. Дозвольте користувачу напряду маніпулювати об'єктами інтерфейсу (інтерактивність).
11. Дозвольте користувачу думати, що він контролює ситуацію.

2. Навантаження на пам'ять користувача

Друге правило проектування інтерфейсу – інтерфейс повинен захищати пам'ять людини від зайвої завантаженості. Принципи для виконання цього правила наступні:

1. Не завантажуйте короткотривалу пам'ять (запам'ятовування).
2. Покладайтесь на розпізнавання, а не на повторення.
3. Забезпечте візуальні підказки (інформування).
4. Передбачте функції відміни останньої дії, її повтору, установки по-замовчуванню (толерантність).

5. Передбачте швидкі шляхи (швидкість).
6. Активізуйте синтаксис дій з об'єктами (інтуїтивність).
7. Використовуйте метафори реального світу (перенос).
8. Поясніть поняття і дії (контекст).
9. Збільшіть візуальну ясність (організація).

3. Сумісність інтерфейсу

Третє правило – сумісність інтерфейсу. Його перевага полягає у тому, що користувачі можуть переносити свої знання і навички здобуті в одній програмі в іншу. Принципи для виконання цього правила наступні:

1. Проектування послідовного інтерфейсу (послідовність).
2. Загальна сумісність всіх програм (досвід).
3. Збереження результатів взаємодії (прогнозування).
4. Естетична привабливість і цілісність (відношення).
5. Заохочення навчання (передбачуваність).

ХІД РОБОТИ

1. Отримати у викладача індивідуальні дані для виконання лабораторної роботи. Принципи проектування програмного інтерфейсу користувача взяти із перших трьох цифр завдання, перша цифра – номер першого принципу першого правила, друга цифра – номер принципу другого правила, третя цифра – номер принципу третього правила. Елемент графічного інтерфейсу користувача взяти відповідно до четвертої цифри завдання. Назву програми взяти відповідно до п'ятої цифри завдання.
2. Розробити програмний інтерфейс користувача з дотриманням заданих принципів проектування інтерфейсу користувача за допомогою засобу візуального програмування. Показати як зміниться інтерфейс, якщо недотримуватись принципів.
3. Провести аналіз використання при проектуванні програмних інтерфейсів користувача заданих в індивідуальному завданні елементів графічного інтерфейсу користувача з точки зору дотримання принципів проектування інтерфейсу.
4. Дослідити інтерфейс заданої програми. Показати як виконуються, чи не виконуються правила і принципи проектування інтерфейсу користувача у програмі.
5. Зробити висновки про виконання роботи.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Принципи дотримання почуття контролю користувача над системою при проектуванні програмного інтерфейсу користувача.
2. Принципи зменшення навантаження на пам'ять користувача при проектуванні програмного інтерфейсу користувача.
3. Принципи дотримання сумісності програмного інтерфейсу користувача.

Варіанти

1) 2 8	3	1	6	31) 4 8	2	3	9
2) 7 7	3	7	2	32) 4 7	5	9	8
3) 3 4	4	4	4	33) 6 2	4	6	9
4) 1 8	4	8	5	34) 3 5	5	10	2
5) 5 2	2	1	5	35) 10 5	2	10	9
6) 6 6	3	9	1	36) 5 6	2	6	7
7) 11 8	1	1	2	37) 6 4	1	9	3
8) 10 2	5	5	10	38) 8 5	4	5	10
9) 6 4	2	10	3	39) 8 9	2	4	3
10) 7 9	4	2	6	40) 1 4	4	9	4
11) 1 6	2	5	1	41) 3 3	4	8	2
12) 4 2	1	10	10	42) 5 5	4	1	9
13) 11 8	5	3	5	43) 5 8	3	3	2
14) 4 2	3	5	7	44) 6 4	3	2	7
15) 6 9	5	9	10	45) 2 3	4	5	6
16) 3 5	5	2	5	46) 9 4	4	7	6
17) 11 4	3	9	2	47) 4 6	5	8	6
18) 9 5	2	6	9	48) 9 9	1	5	7
19) 8 8	3	10	7	49) 10 8	4	4	5
20) 7 6	3	5	2	50) 9 4	5	7	7
21) 9 1	2	6	4	51) 3 8	1	3	2
22) 4 4	3	4	2	52) 3 3	5	8	4
23) 11 4	4	6	10	53) 10 8	1	10	10
24) 8 1	1	8	7	54) 3 9	5	6	9
25) 7 8	4	6	6	55) 2 1	1	6	4
26) 5 5	2	7	4	56) 4 2	3	7	9
27) 9 2	1	9	7	57) 4 8	5	7	4
28) 8 7	4	7	3	58) 6 9	4	8	2
29) 8 6	4	9	10	59) 8 2	4	5	9
30) 7 4	3	8	7	60) 9 6	3	6	4

61)	6	5	1	8	7
62)	5	6	1	8	6
63)	1	1	3	10	2
64)	3	7	2	6	2
65)	4	1	2	10	10
66)	5	3	2	7	1
67)	3	6	4	2	3
68)	11	2	2	5	4
69)	4	6	2	1	3
70)	11	9	1	4	4
71)	7	1	3	4	6
72)	6	5	5	4	8
73)	11	8	3	2	5
74)	8	9	3	7	1
75)	8	5	2	5	4
76)	6	1	3	4	5
77)	1	2	4	9	1
78)	6	6	3	9	10
79)	4	3	4	3	4
80)	1	1	3	9	8
81)	10	2	2	9	5
82)	10	8	5	7	3
83)	5	5	4	3	7
84)	2	7	5	5	8
85)	6	4	5	6	2
86)	10	6	5	8	6
87)	8	8	1	1	7
88)	11	7	1	2	6
89)	6	9	2	2	1
90)	3	7	1	9	3
91)	11	2	5	3	10
92)	10	3	1	6	10
93)	1	2	2	9	10
94)	5	1	2	10	4
95)	8	2	4	8	10
96)	9	5	3	7	2
97)	6	1	2	2	5
98)	9	6	5	4	2
99)	5	3	5	9	7

Елементи інтерфейсу користувача

Номер елемента інтерфейсу користувача у завданні	Назва елемента інтерфейсу користувача
1.	меню
2.	список
3.	прапорці та перемикачі
4.	панелі інструментів
5.	вкладки
6.	спливаючі підказки
7.	командні кнопки
8.	таблиця
9.	об'єкт OLE
10.	поле для вводу

Програма

Номер програми у завданні	Назва програми
1.	MS Word
2.	MS Excel
3.	Mathcad
4.	Explorer
5.	калькулятор MS Windows
6.	Paint Brush
7.	MatLab
8.	PhotoShop
9.	CorelDraw
10.	Acrobat Reader

Лабораторна робота №4

Тема Квантифікація інтерфейсів. Модель GOMS. Закони Хіка і Фітса

Мета Навчитися проектувати інтерфейси з урахуванням методів квантифікації: за допомогою моделі GOMS та законів Хіка і Фітса.

Теоретичні відомості

Необхідно вміти проектувати інтерфейси чітко уявляючи для себе, які користувачі будуть працювати з ним та за яких умов це буде відбуватися. Для різних груп користувачів (звичайні люди, літні, малюки, працівники рятувальних служб, швидкої допомоги, call- центрів тощо) необхідно розробляти інтерфейси враховуючи їх особливості або вимоги до їх умов праці (необхідність мінімізації часу відгуку).

Для кожної задачі, яку необхідно вирішити, завжди існують декілька варіантів рішення. Створення інтерфейсу для заданої проблемної області можливо як за допомогою комбінацій стандартних елементів, так і за допомогою поєднання графічних примітивів, малюнків з технологіями click-to-choose, click-to-activate, drag-and-drop або можливостями Web-інтерфейсів.

Необхідно пам'ятати, що недоречне використання стандартних елементів може призвести до дезорієнтації користувача. Надмірна кількість елементів або невдале їх групування також знижує ефективність інтерфейсу.

У деяких випадках розроблений інтерфейс буде здаватися самим швидким, але за допомогою методів квантифікації інтерфейсів можна визначити, що його переваги існують тільки, якщо параметри задані за замовчуванням збігаються з потребами користувача. При необхідності зміни масштабу або діапазону параметрів кількісні показники такого інтерфейсу різко знижуються.

За допомогою моделі GOMS визначають кількісні показники розробленого інтерфейсу. Після знайомства з основами цієї моделі досвідчені розробники не проводять детальні розрахунки кожного інтерфейсу, що розробляють, оскільки вони вже автоматично використовують принципи, закладені в цієї моделі. До формального аналізу прибігають у випадках, коли необхідно вибрати один з 2 варіантів розробки, якщо навіть невеликі розходження у швидкості можуть давати психологічний або економічний ефект.

Знання законів Хіка і Фітса дозволяє кількісно визначити деякі важливі характеристики інтерфейсів.

Порядок виконання лабораторної роботи

1. Спроекувати і програмно реалізувати декілька варіантів (≥ 5) інтерфейсу для обраної дії зі списку наведеного нижче. *Вибір варіанту необхідно узгодити з викладачем!*

Можливі варіанти:

- 1) Перевід числа з однієї системи в іншу;
- 2) Побудова показника/Відображення зросто-вагової залежності;
- 3) Вибір кольору;
- 4) Вибір місця вибуття/місця прибуття подорожі із вказівкою відстані;
- 5) Вибір лікарської засобу з довідника ліків;
- 6) Визначення дня народження;
- 7) Задання хімічної формули;
- 8) Задання лінії в просторі;
- 9) Ідентифікація тварини/рослини по класифікації видів тваринного/рослинного світу;
- 10) Відгадування загаданого слова (Поле чудес);
- 11) Ознайомлення першокласника з абеткою (з можливістю складати слова);
- 12) Набір тексту з діакритичними знаками (різні надрядкові, підрядкові, рідше внутрірядкові знаки, що застосовують в різних системах напису) на звичайній клавіатурі (див. Додаток 1);

Відображення переваги при виборі товару;

Ваша тема..

2. Визначте за допомогою моделі GOMS найефективніший та найгірший з запропонованих варіантів рішення. Обґрунтуйте відповідь.

3. Наведіть приклади використання в Ваших варіантах законів Хіка і Фітса

Контрольні запитання та завдання

1. Який інтерфейс є ергономічним?
2. Що вивчає когнітивна психологія?
3. Чим відрізняється короткострокова пам'ять від довгострокової?
4. Які 2 методи роботи з інформацією із довгострокової пам'яті існують?
5. Що зберігається у буферної пам'яті?
6. Що таке процес навчання людини?
7. Що таке процес сприйняття людини?

8. Які існують види пам'яті у людини?
9. Чим відрізняються процеси розпізнавання й відновлення інформації?
10. Які переваги та недоліки має пам'ять комп'ютера у порівнянні з пам'яттю людини?
11. На яких параметрах базуються викладки моделі GOMS?
12. Чи дає ця модель обрахувати абсолютні значення? Чому?
13. Перелічіть правила моделі GOMS.
14. Чи ефективно використовуватися один й той самий варіант інтерфейсу для різних груп користувачів? Чому?
15. Які є переваги у інтерфейсу, що розроблений за допомогою стандартних елементів?
16. Про що допомагають пересвідчитися закони Хіка і Фітса?
17. У яких випадках має сенс використовувати модель GOMS при розробці інтерфейсів?

Звіт повинен містити:

- титульний лист;
- мета роботи;
- опис варіантів розроблених інтерфейсів для обраної дії;
- обґрунтування області застосування кожного варіанту;
- скриншоти варіантів;
- розрахунки зробленого квантифікаційного аналізу;
- висновки по роботі.

Лабораторна робота №5
ТЕСТУВАННЯ (ОЦІНЮВАННЯ) ІНТЕРФЕЙСУ БЕЗ УЧАСТІ
РЕАЛЬНИХ КОРИСТУВАЧІВ

1 Мета роботи

Навчитися застосовувати на практиці при розробці інтерфейсів такі методи тестування: когнітивний наскрізний перегляд (контроль), Back-of-the-Envelope та евристичний аналіз.

2 Методичні вказівки з організації самостійної роботи студентів

При підготовці до лабораторної роботи необхідно повторити матеріал відповідних лекцій та ознайомитися з [Мандел Т. Разработка пользовательского интерфейса: Пер. с англ. –М.: ДМК Пресс, 2001, стор. 112 - 134], [<http://hcibib.org/tcuid/chap-4.html> (in English)], [http://usethics.ru/service/expert_review/], [Константайн Л., Локвуд Л. Разработка программного обеспечения: Пер. с англ. - Питер, 2004, стор. 424 - 450]. До лабораторної роботи рекомендується виконати практичне заняття №4.

2.1 Важливість тестування на зручність користування ПП

Тестування на зручність користування проводиться абсолютною більшістю компаній з розробки програмного забезпечення починаючи з 90-х років минулого сторіччя.

Проведення такого тестування дозволяє зрозуміти, чи зможуть споживачі взагалі використовувати даний продукт. Нижче наведені декілька причин, чому тестування на зручність є дуже важливим етапом розробки:

- термінологія розробників не завжди співпадає з термінологією, до якої звикли користувачі;
- інформація, що отримується від користувачів телефоном або електронною поштою, не є достатньою для проведення оцінки якості ПП;
- час, фінанси та людські ресурси, що використовуються в процесі тестування, завжди окупаються;
- програми, що створені з декількох частин, зазвичай, виявляються несумісними;
- проблеми, які виявляються на кінцевій стадії розробки, складніше й дорожче виправляти;

- оцінка зручності використання може дати перевагу над конкурентною розробкою.

Тестуванні на зручність використання здійснюється на різних етапах розробки ПП (зразок дизайну на папері, програмні прототипи) для забезпечення зворотного зв'язку з користувачем. Це допомагає удосконалювати проект в цілому, зменшує кількість помилок, а також підтверджує відповідність продукту вимогам, що ставляться до нього.

Спеціалістам відомі багато методів, за допомогою яких на різних етапах можна оцінити зручність використання інтерфейсу. Такі перевірки проводяться як самими розробниками без участі користувачів (на етапі прототипів), так і з залученням користувачів з можливої цільової аудиторії (на наступному етапі).

2.2 Перелік основних методів, що використовуються при тестуванні інтерфейсу без участі користувача

Когнітивний наскрізний перегляд (контроль) (Cognitive Walkthrough) – проводиться шляхом обмірковування можливих проблем і помилкових шляхів користувача.

Back-of-the-Envelope аналіз (у перекладі «що не потребує складних розрахунків») як один з методів аналізу дій (Action Analysis) - являє собою аналіз дій на більше поверхневому рівні деталізації в порівнянні з GOMS моделлю. В процесі цього аналізу здійснюється продумування дій та визначається ємність розумових витрат користувача, необхідних для вирішення деякого завдання, яке виконується за допомогою інтерфейсу програмної системи, що тестується. В результаті цього аналізу можуть бути знайдені «вузькі» місця інтерфейсу, в яких користувач витрачає багато часу для подолання частки поставленого завдання. Такі місця мають бути перепроектовані шляхом зняття складних функцій з користувача й передачі їх самій системі.

Евристичний аналіз (Heuristic Analysis) - свого роду перевірка за контрольними списками, які охоплюють різноманітні проблеми. Вимагає участі декількох учасників, що мають знання в області юзабиліти.

2.3 10 основних принципів (евристик) створення інтерфейсу

Ці 10 евристик не є твердими правилами, яким обов'язково необхідно слідувати, але досвід розроблювачів свідчить, що їх застосування неодмінно сприяє підвищенню зручності, збільшує ефективність і простоту використання інтерфейсів (як програмних так й будь-яких інших):

- Інформування про поточний стан системи;
- Використання метафор реального миру;
- Надання контролю й свободи користувачеві в навігації;
- Послідовність і слідування стандартам;
- Попередження помилок;
- Розпізнавання, а не відновлення інформації;
- Гнучкість і ефективність використання;
- Естетична привабливість і мінімалізм;
- Допомога користувачеві у визначенні помилкових дій, у їхній діагностиці й виправленні;
- Надання довідкової інформації й документації.

У додатку А наведений контрольний список, перевірка за яким є одним з етапів проведенні евристичного аналізу. Така перевірка є ефективним і економічним засобом підвищення якості інтерфейсів. Наведений список скоріше є шаблоном тому, що у кожному конкретному випадку необхідно розробляти свій власний контрольний список, оскільки він має урахувати специфіку програмного засобу, що розроблюється, й можливості засобів розробки. Перевагою цього списку являється наявність більшості розділів, що мають бути перевіреними.

3 Порядок виконання лабораторної роботи

1 Для інтерфейсу, який був розроблений на лабораторній роботі №3¹ іншим студентом (за вибором викладача), необхідно провести з обов'язковим письмовим звітом тестування методами:

а) когнітивного наскрізного перегляду (контролю);

Для проведення тестування оберіть 2-3 основні дії або функції, які надаються програмою.

б) Back-of-the-Envelope аналізу;

в) евристичного аналізу (перевірка за контрольними списками: 10 основних принципів зручності інтерфейсу та список з Додатку В).

4 Контрольні запитання та завдання

1. Які види тестувань інтерфейсів Ви знаєте?
2. Перелічіть методи, що використовуються, для тих видів тестувань, які були наведені Вами у відповідь на попереднє запитання?
3. Перелічіть переваги та недоліки кожного з видів тестувань?
4. У чому полягає смисл та алгоритм кожного з перелічених Вами у відповідь на попереднє запитання методів? Чи дозволяють ці методи отримати кількісні показники?
5. Якщо у результаті зробленого за одним з цих методів був знайдений недолік інтерфейсу, якими будуть Ваші дії як розробника цього інтерфейсу?
6. Наведіть список обов'язкових до розглядання питань при здійсненні когнітивного наскрізного перегляду;
7. Наведіть список обов'язкових до розглядання питань при здійсненні Back-of-the-Envelope аналізу;
8. Який з двох методів аналізу дій проводиться на більш детальному рівні дій користувача?
9. Який з трьох методів має проводитися раніше інших: Back-of-the-Envelope аналіз, когнітивний наскрізний перегляд або евристичний аналіз?
10. Яка з проблем, що можуть бути виявлені на етапі когнітивного наскрізного перегляду, є найскладнішою для усунення?
11. При розробці яких систем проведення когнітивного наскрізного перегляду є першочерговим завданням?
12. Наведіть приклади використання у повсякденному житті кожної з десяти основних евристик юзабіліті (слідування принципам підвищення зручності використання).

5 Зміст звіту

Звіт має містити:

- титульний лист;
- мету роботи;
- опис програмної системи, інтерфейс якої підлягає тестуванню;
- звіт проведеного когнітивний наскрізного перегляду (контролю);
- звіт проведеного Back-of-the-Envelope аналізу;
- перелік змін, що мають бути внесеними до програми, що тестувалася;
- висновки по роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Купер, А. Основи проектування взаємодії / А. Купер. – СПб. : Символ-Плюс, 2020. – 688 с.
2. Людино-машинні системи автоматизації: управління якістю, безпекою і надійністю/ Архангельський В.І., Богаєнко І.М., Грабовський Г.Г., Рюмшин М.О.– К.: НВК “КІА”, 2000.– 296с.
3. Мандел, Т. Розробка користувацького інтерфейсу / Т. Мандел. – М. : ДМК Пресс, 2019. – 416 с.
4. Марценюк Є.О. Опорний конспект лекцій з дисципліни «Людино – машинна взаємодія» /Є.О. Марценюк, Л.І. Гончар//Тернопіль, ЗУНУ, 2021.–55 с.
5. Унгер, Р. UX-дизайн : практ. посіб. З проектування досвіду взаємодії / Р. Унгер, К. Чендлер. – СПб. : Символ-Плюс, 2019. – 336 с.
6. Тідвелл, Дж. Розробка користувацького інтерфейсу / Дж. Тідвелл. – СПб. : Пітер, 2018. – 416 с.
7. UX Design. <https://designmodo.com/design/ux-design/>
8. Windows Desktop Apps. Guidelines. <https://docs.microsoft.com/en-us/windows/desktop/uxguide/guidelines>
9. User Interface Principles. <https://docs.microsoft.com/en-us/windows/desktop/appuistart/-user-interface-principles>
10. Andrews K. Human-Computer Interaction.– URL: <http://courses.iicm.edu/hci/>, 2019.
11. Lewis C., Rieman J. Task-Centered User Interface Design.– http://pages.cpsc.ucalgary.ca/~saul/hci_topics/papers/LewisRiemanBook/, 1993.
12. Jakob Nielsen's Website.– URL: <http://www.useit.com/>.
13. Official Guidelines for User Interface Developers and Designers.– URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/welcome.asp>.
14. Practical real-world design.– URL: <http://www.asktog.com/>.
15. Usability в России.– URL: <http://www.usability.ru/>.
16. Usable Web.– URL: <http://www.usableweb.com/>.

Вимоги до конкретних елементів керування

Кнопки

- Всі кнопки, що запускають дії, мають текст в інфінітивній формі дієслова (приклад: шукати), а не іншу частину мови або форму дієслова (приклад: готовий). Давати кнопці текст "ОК" можна, тільки якщо яке-небудь дієслово не вміщується на кнопці;

- Між кнопками, що розміщені поруч, має бути порожній простір, клацання по якому не спрацьовується;

- Недоступні команди не зникають із екрана, а стають заблокованими;

- Частотні кнопки супроводжуються не тільки текстом, але й піктограмами; кнопки, що використовуються рідко - тільки текстовими підписами.

-

Поля уведення

- Полях уведення вже мають найбільш імовірні значення;

- Якщо в полі вводиться чисельне значення, границі діапазону виводяться в спливаючій підказці;

- Якщо в полі вводиться чисельне значення з обмеженого діапазону, поле супроводжується Spinner-ом;

- Довжина полів не менше, і, по можливості, не більше, довжини даних, що вводяться в них.

- Якщо поле призначене для уведення помітної кількості тексту, воно многорядкове.

Списки

- Списки вже мають найбільш імовірні значення;

- Якщо список містить більше 50 елементів, використовується фільтр або режим пошуку;

- Немає коротких списків (менш п'яти елементів); такі списки представлені як групи радіокнопок або чекбоксів;

- Ширина списків не менше ширини вхідних у них елементів;

- Елементи списку відсортовані; або структурно, тобто по загальних ознаках, або за алфавітом, або по частотності (тільки списки менше 7 елементів);

- Якщо в списку більше 50 відсортованих за алфавітом елементів, першими трьома елементами є найбільш частотні елементи. Вони також повторюються на своїх алфавітних місцях.

Чекбокси й радіокнопки

- Якщо чекбоксів у групі більше 10, вводиться додатковий, що виставляє/знімає всі чекбокси;
- Чекбокси й радіокнопки усередині своїх груп розставлені по вертикалі;
- Якщо у вікні, крім термінаційних кнопок, є тільки набір радіокнопок, подвійне клацання по радіокнопці встановлює її й закриває вікно.

Взаємодія

- Система, завершивши тривалу операцію (більше мінути роботи), пищить через вбудований динамік комп'ютера;
- Якщо в інтерфейсі не використовується безпосереднього маніпулювання, система не має своїх курсорів. Якщо безпосереднє маніпулювання є, свої курсори застосовуються тільки якщо аналогів з ОС не існує.

Системні повідомлення й відпрацьовування помилок

- У формах уведення перевірка коректності значень, що вводяться, виконується прямо під час уведення; якщо вводяться некоректні дані, система відразу сповіщає про це користувачеві, не чекаючи моменту, коли користувач завершить уведення даних у всій формі;
- Повідомлення про некоректність уведених даних показуються поруч із елементом керування, дані в якому некоректні;
- Текст повідомлень про некоректність уведених даних не говорить про те, що, мов, зроблена помилка, навпроти, він тільки інформує користувача, дані якого типу й формату прийнятні;
- Якщо вікно з текстом про некоректність уведених даних спливає - воно не повинне заслоняти собою поле уведення із текстом помилки;
- Текст повідомлень про проблеми складається із трьох частин: у першій коротко описується проблема, у другій частині - як її вирішити, у третьої - описується, як не допускати виникнення цієї проблеми надалі;
- Статусні повідомлення ("Синхронізація успішно завершена") виводяться тільки в рядку статусу.

Клавіатура

- Обробка форми запускається не тільки по натисканню на термінаційну кнопку, але й по натисканню клавіші Enter на кожному полі цієї форми;
- Для найбільш частотних елементів керування (включаючи меню) установлені клавіші швидкого виклику;
- Кожному пункту меню призначені ALT-комбінації (виділені підкресленням);
- ALT-комбінації й гарячі клавіші стандартні;
- Якщо гарячих клавіш більше 40, в інтерфейсі є спосіб їх змінити;
- По натисканню клавіші Tab перехід від елемента до елемента усередині форми здійснюється зверху вниз ліворуч праворуч.

Візуалізація

Напрямок тіней у всіх елементах керування має бути однаковим: знизу праворуч.

Індикація

Індикація кольором не є єдиною; якщо вона використовується, система супроводжується й іншою індикацією (звук, анімація тощо).

Піктограми

- У групах піктограм немає піктограм, за кольором й формою подібних між собою;
- Немає піктограм зі стандартними значеннями, але нестандартними сюжетами.

Вікна

- На вікнах, що розтягуються, є індикатор розтягування;
- Заголовки вікон відповідають назвам елементів, за допомогою яких вікна були викликані. Якщо вікно викликається елементом, що не має явної назви, у заголовку вікна відбивається назва екранної форми;
- Тип вікна (модальна, немодальна, можливість мінімізації/максимізації) обирається свідомо, відповідно до завдань користувачів;
- У діалогових вікнах відсутні меню або інструментальні панелі.

Рядок статусу

- У рядку статусу виводиться інформація тільки про поточний стан системи;
- Індикатори виконання виводяться в рядку статусу. Виключення: «вікна-майстри», - у них індикатори виконання можна виводити усередині самих вікон.

Меню

- Перша буква в назві пунктів меню – заголовна;
- Всі пункти меню першого рівня активізують меню, що розкриваються;
- Використаються не більше двох підрівнів меню;
- Якщо в меню є піктограми, ними супроводжуються тільки самі частотні елементи;
- Елементи, що відкривають вкладені меню, виглядають інакше, чим термінальні елементи.

Контекстні меню

- На всіх об'єктах, видимих в інтерфейсі, є специфічне для кожного об'єкта контекстне меню;
- У контекстні меню не більше 10 елементів;
- У контекстні меню елементи відсортовані по убутанню частоти їх використання
- Всі елементи контекстних меню присутні й в інших фрагментах інтерфейсу; немає команд, що викликаються тільки з контекстних меню.

Структура форм інтерфейсу

- У групах інтерактивних елементів (поля форм, елементи меню й т.п.) цих елементів не більше семи;
- Кнопка "Скасування" завжди сама права;
- Багатосторінкові форми мають вказівка на те, що вони багатосторінкові; користувач завжди бачить кількість екранів, що залишилися (приклад: "Екран х з у");
- Якщо у формі є кілька кнопок, одна є кнопкою за замовчуванням. Небезпечні для користувача кнопки не є кнопками за замовчуванням;
- Якщо у вікні є вільне місце, найбільш частотна термінаційна кнопка більше інших;
- Кнопки перебувають у секції, на яку вони роблять безпосередній вплив;
- Термінаційні кнопки (що керують вікном) розташовані або знизу в ряд, або праворуч у стовпчик.
- Кнопки, що впливають на декілька блоків, розташовані за межами цих блоків;
- Якщо вікно або вкладка має автоматично поповнюваний вміст, наприклад, у ньому перераховані повідомлення, що надходять, у назві елемента інтерфейсу, що відкриває вікно або вкладку, виводиться число об'єктів у цьому вікні й окремо число нових об'єктів. Приклад: Документи (3/8);

- Пункти меню й кнопки, що ініціюють інші дії користувача, позначені наприкінці багатокрапкою (...). Приклади: елемент "Зберегти як..." вимагає багатокрапки, тому що користувач має вибрати назву файлу, а елемент "Про програму" багато крапки не вимагає, тому що на вікні, що відкривається, немає самостійних елементів інтерфейсу;

- Підписи до елементів інтерфейсу розміщені одномоментно;
- Недоступні в цей момент елементи інтерфейсу заблоковані, а не сховані;
- В інтерфейсі присутні повідомлення про виконання тієї або іншої дії.

Наприклад, повідомлення про те, що дані успішно збережені або щось вилучене й т.п.

Форми введення

- У форми введення є назва;
- У всіх формах, що служать для збору інформації, є пункт "Інше" або подібний;
- Всі поля, обов'язкові для заповнення, позначені, і є відповідне пояснення;
- Багатосторінкові форми введення мають кнопки "Назад" і "Далі";
- У всіх формах, що служать для збору інформації, є опис цілей збору даних, пояснюється, що із цими даними буде зроблене й що не буде.

Текст

- На всі головні елементи інтерфейсу повішені спливаючі підказки, текст яких відбиває результат використання цих елементів;
- В інтерфейсі відсутні жаргонізми;
- В інтерфейсі відсутні негативні формулювання (наприклад, чекбокс "Не показувати примітки" неприйнятний, замість нього потрібно виводити чекбокс "Показувати примітки");
- Жоден елемент не називається по-різному в різних місцях (глосарій інтерфейсу не просто зроблений у явній формі, але й вивірений);
- У тексті всіх підтверджень дається найменування об'єкта, над яким відбувається підтверджувана дія;
- Для ясності довгі числа розбиваються нерозривним пробілом по три цифри: 1 234 567;
- Будь-якому списку передують, щонайменше, один абзац тексту;
- У таблицях всі стовпці із цифрами вирівнюються по правому краю;
- Підписи до елементів інтерфейсу починаються із прописної букви.

Навчально-методичне видання

**Марценюк Євгенія Олексіївна
Гончар Людмила Іванівна**

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з курсу

“ЛЮДИНО-МАШИННА ВЗАЄМОДІЯ”

Підписано до друку 21.09.2022 р.
Формат 60x84/16. Папір офсетний.
Друк офсетний. Зам. № 22-929
Умов.-друк. арк. 1,9. Обл.-вид. арк. 2,1.
Тираж 30 прим.

Віддруковано ФО-П Шпак В. Б.
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.
м. Тернопіль, бульвар Просвіти, 6/4. тел. 097 299 38 99.
E-mail: tooums@ukr.net

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції
ДК № 7599 від 10.02.2022 р.*