*Національний технічний університет України*

*‘Київський політехнічний інститут’*

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

***Лабораторна робота №3***

*“ Розробка моделі інтерфейсу користувача ”*

*Виконав: Мроць Ю.Б.*

*Група: ІО-12*

*Номер зк: 1219*

*Перевірив: Абу Усбах О.Н.*

*Київ — 2013*

***Мета***

Отримання базових навичок з використання мови UML для формального опису взаємодії користувача з програмним додатком. Розробка дизайну інтерфейсу користувача з використанням елементів, що реалізовані в Java.

***Завдання***

1. Вивчити принципи розробки ілюстрованого сценарію прецедента, принципи побудови UML-діаграм класів та послідовності передачі повідомлень.

2. Ознайомитись з поняттям граничних класів, та з їхнім використанням для побудови логічної моделі інтерфейсу користувача.

3. Ознайомитись з принципами дизайну інтерфейсу користувача.

4. Ознайомитись з елементами графічного інтерфейсу користувача з бібліотеки AWT.

5. Розробити ілюстровані сценарії прецедентів для програми візуалізації табличних даних у вигляді діаграми.

6. Розробити логічну модель графічного інтерфейсу користувача.

7. Розробити дизайн графічного інтерфейсу користувача з використанням елементів, що реалізовані в Java.

8. Скласти таблицю відповідності елементів дизайну Java класам з бібліотеки AWT.

***Початковий опис потоку подій роботи з редактором кільцевих діаграм***

1. Прецедент починається, коли користувач запускає програму.
2. Користувач може виконати одну з наступних дій:
3. Створити файл.
4. Зберегти файл.
5. Відкрити збережений файл.
6. Додати сектор в діаграму.
7. Видалити сектор з діаграми.
8. Змінити назву сектора.
9. Змінити колір сектора.
10. Змінити розмір сектора.
11. Закрити файл.
12. Очистити діаграму.
13. Прецедент закінчується, коли користувач завершує роботу з програмою.

***Розробка орієнтирів***

1. Прецедент починається, коли користувач запускає програму.
2. Користувач може виконати одну з наступних дій:
   1. Створити файл. [Створюється кільцева діаграма з назвою diagram1 і сектором, що займає 100% місця і називається sector1.]
3. Зберегти файл. [Файл повинен мати розширення CSV.]
4. Відкрити збережений файл. [Файл відкривається в новій вкладці, яка після відкриття буде активною.]
5. Додати сектор в діаграму. [Користувач повинен мати можливість ввести назву, розмір або відсоток, що займатиме новий сектор.]
6. Видалити сектор з діаграми.
7. Змінити колір сектора.
8. Змінити назву сектора.
9. Змінити розмір сектора.
10. Закрити файл. [Якщо файл не був збережений, користувач має бути повідомлений про це.]
11. Очистити діаграму. [Користувач повинен підтвердити очищення всієї діаграми.]
12. Прецедент закінчується, коли користувач завершує роботу з програмою.[Якщо є незбережені файли, користувач отримує повідомлення про це.]

***Розробка атрибутів***

1. Прецедент починається, коли користувач запускає програму.
2. Користувач може виконати одну з наступних дій:
   1. Створити файл.
3. Зберегти файл. {Назва файлу в середньому 8 символів.}
4. Відкрити збережений файл. {Назва файлу в середньому 8 символів.}
5. Додати сектор в діаграму. {Середній розмір сектора - 200}
6. Видалити сектор з діаграми.
7. Змінити колір сектора.
8. Змінити назву сектора. { Назва сектору в середньому 8 символів.}
9. Змінити розмір сектора. {Середній розмір сектора - 200}
10. Закрити файл.
11. Очистити діаграму.
12. Прецедент закінчується, коли користувач завершує роботу з програмою.

***Розробка інтенсивності використання***

1. Прецедент починається, коли користувач запускає програму.
2. Користувач може виконати одну з наступних дій:
   1. Створити файл. (Використовується в 60% випадків.)
3. Зберегти файл. (Використовується в 95% випадків.)
4. Відкрити збережений файл. (Використовується в 40% випадків.)

d) Додати сектор в діаграму. (Використовується в 85% випадків.)

e) Видалити сектор з діаграми. (Використовується в 70% випадків.)

f) Змінити колір сектора. (Використовується в 50% випадків.)

g) Змінити назву сектора. (Використовується в 55% випадків.)

h) Змінити розмір сектора. (Використовується в 40% випадків.)

i) Закрити файл. (Використовується в 100% випадків.)

j) Очистити діаграму. (Використовується в 35% випадків.)

1. Прецедент закінчується, коли користувач завершує роботу з програмою.

***Результуючий опис потоку подій програми***

1) Прецедент починається, коли користувач запускає програму.

2) Користувач може виконати одну з наступних дій:

a) Створити файл. (Використовується в 60% випадків.); [Створюється кільцева діаграма з назвою diagram1 і сектором, що займає 100% місця і називається sector1.]

b) Зберегти файл. (Використовується в 95% випадків.); {Назва файлу в середньому 8 символів.}; [Файл повинен мати розширення CSV.]

c) Відкрити збережений файл. (Використовується в 40% випадків.); {Назва файлу в середньому 8 символів.}; [Файл відкривається в новій вкладці, яка після відкриття буде активною.]

d) Додати сектор в діаграму. (Використовується в 85% випадків.); {Середній розмір сектора - 200}; [Користувач повинен мати можливість ввести назву, розмір або відсоток, що займатиме новий сектор.]

e) Видалити сектор з діаграми. (Використовується в 70% випадків.)

f) Змінити колір сектора. (Використовується в 50% випадків.)

g) Змінити назву сектора. (Використовується в 55% випадків.); { Назва сектору в середньому 8 символів.}

h) Змінити розмір сектора. (Використовується в 40% випадків.); {Середній розмір сектора - 200}

i) Закрити файл. (Використовується в 100% випадків.)

j) Очистити діаграму. (Використовується в 35% випадків.); [Користувач повинен підтвердити очищення всієї діаграми.]

3) Прецедент закінчується, коли користувач завершує роботу з програмою. [Якщо є незбережені файли, користувач отримує повідомлення про це.

***Діаграма граничних класів***

******

***Доповнення опису ілюстрованого сценарію об’єктами діаграми***

1) Прецедент починається, коли користувач запускає програму. «Редактор кільцевих діаграм»

2) Користувач може виконати одну з наступних дій:

a) Створити файл. (Використовується в 60% випадків.); [Створюється кільцева діаграма з назвою diagram1 і сектором, що займає 100% місця і називається sector1.]; «Кнопка Новий файл»

b) Зберегти файл. (Використовується в 95% випадків.); {Назва файлу в середньому 8 символів.}; [Файл повинен мати розширення CSV.]; «Кнопка Зберегти файл»

c) Відкрити збережений файл. (Використовується в 40% випадків.); {Назва файлу в середньому 8 символів.}; [Файл відкривається в новій вкладці, яка після відкриття буде активною.]; «Кнопка Відкрити файл»

d) Додати сектор в діаграму. (Використовується в 85% випадків.); {Середній розмір сектора - 200}; [Користувач повинен мати можливість ввести назву, розмір або відсоток, що займатиме новий сектор.]; «Кнопка Додати сектор»

e) Видалити сектор з діаграми. (Використовується в 70% випадків.); «Кнопка Видалити сектор»

f) Змінити колір сектора. (Використовується в 50% випадків.); «Кнопка Змінити параметри сектора»

g) Змінити назву сектора. (Використовується в 55% випадків.); { Назва сектору в середньому 8 символів.} ; «Кнопка Змінити параметри сектора»

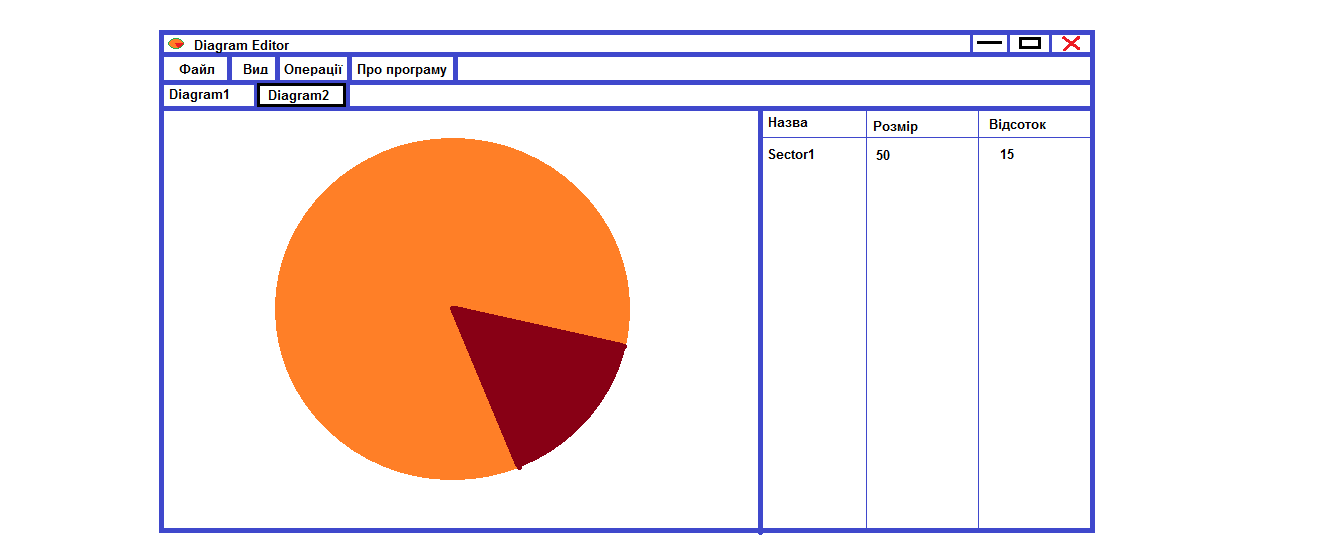
h) Змінити розмір сектора. (Використовується в 40% випадків.); {Середній розмір сектора - 200}; «Кнопка Змінити параметри сектора»

i) Закрити файл. (Використовується в 100% випадків.) ; «Кнопка Закрити файл»

j) Очистити діаграму. (Використовується в 35% випадків.); [Користувач повинен підтвердити очищення всієї діаграми.] ; «Кнопка Очистити діаграму»

3) Прецедент закінчується, коли користувач завершує роботу з програмою. [Якщо є незбережені файли, користувач отримує повідомлення про це.]

***Приблизне геометричне розташування компонентів застосунку на екрані***

**

***Для реалізації інтерфейсу можна використати***

|  |  |
| --- | --- |
| ***Елемент інтерфейсу*** | ***Клас пакету java.awt*** |
| Головне вікно | Frame |
| Панель «Меню» | MenuBar |
| Панель «Вкладки» | BorderLayout |
| Вкладка | FileDialogue |
| Область відображення діаграми | BorderLayout |
| Діаграма | Graphics |
| Інформаційна область діаграми | BorderLayout |
| Список секторів | List |
| Кнопка «Файл» | Menu |
| Кнопка «Вид» | Menu |
| Кнопка «Операції» | Menu |
| Кнопка «Про програму» | Menu |
| Кнопка «Новий файл» | MenuItem |
| Кнопка «Зберегти файл» | MenuItem |
| Кнопка «Відкрити файл» | MenuItem |
| Кнопка «Закрити файл» | MenuItem |
| Кнопка «Змінити колір вікна» | MenuItem |
| Кнопка «Змінити шрифт» | MenuItem |
| Кнопка «Додати сектор» | MenuItem |
| Кнопка «Видалити сектор» | MenuItem |
| Кнопка «Змінити параметри сектора» | MenuItem |
| Кнопка «Очистити діаграму» | MenuItem |
| Кнопка «Версія» | MenuItem |
| Кнопка «Автор» | MenuItem |
| Кнопка «Опис» | MenuItem |
| Кнопка «Змінити фон» | MenuItem |

***Висновок***

*Виконавши цю лабораторну роботу, я отримав базові навички з використання мови UML для формального опису взаємодії користувача з програмним додатком; з розробки дизайну інтерфейсу користувача з використанням елементів, що реалізовані в Java у пакеті java.awt.*