

Dokumentacja „Prosty system okien – komponenty, które mogą być umieszczone na formularzu” Michał Stolarz AiR

Założenia wstępne:

1. Komponentami nazywa się wszelkiego rodzaju obiekty reprezentujące faktyczne : pola tekstowe, przyciski, grafiki, listy itp. Ich tworzenie odbywa się poprzez dynamiczną alokację pamięci.
2. Formularzem nazywa się wirtualną przestrzeń, w której są zamieszczane tworzone komponenty (w aplikacji rozmiar formularzu wynosi 2000x2000)
3. Zarządzanie stworzonymi obiektami jest realizowane poprzez kontener zawierający wskazania na przydzielone miejsca w pamięci dla obiektów
4. Komponenty są tworzone i dodawane przez użytkownika. Oprócz wybrania rodzaju komponentu i jego współrzędnych użytkownik ma również możliwość usuwania oraz „używania” stworzonych komponentów (tzn. użytkownika zgodnie z przeznaczonym dla nich celem (np.: wpisanie tekstu do rubryki, lub kliknięcie na przycisk)).
5. Wszystkie czynności podejmowane przez użytkownika odbywają się za pomocą interfejsu użytkownika. Pojawiają się w nim listy czynności, lub opcji do wyboru. Użytkownik wybiera odpowiednią pozycję poprzez wprowadzenie jej numeru za pomocą standardowego wejścia.
6. W przypadku wprowadzenia niepoprawnych danych program wyświetla zapytanie do użytkownika o wprowadzenie poprawnych danych (do skutku)
7. W przypadku gdy dwa komponenty nachodzą na siebie , a użytkownik chce „kliknąć” na dane współrzędne (na których znajdują się nachodzące na siebie komponenty), wówczas „kliknięty” zostanie obiekt, który był aktywowany jako ostatni (później niż drugi). Aktywowany oznacza tutaj utworzony, lub „kliknięty” (na innych współrzędnych nie kolidujących z innym komponentem). Jeśli komponent jest w całości przysłonięty przez drugi komponent i był aktywowany wcześniej niż przysłaniający go komponent, wówczas nie można już go aktywować (należy wówczas wykasować, lub przesunąć komponent przysłaniający)
8. Historia aktywacji obiektów (dzięki której jest rozstrzygnięty problem dwóch nachodzących na siebie komponentów) jest realizowana poprzez zachowanie odpowiedniej kolejności wskazań (na komponenty) w przechowyującym kontenerze

Proponowany podział na klasy:

Aplikacja składa się z klas umożliwiających utworzenie następujących elementów:

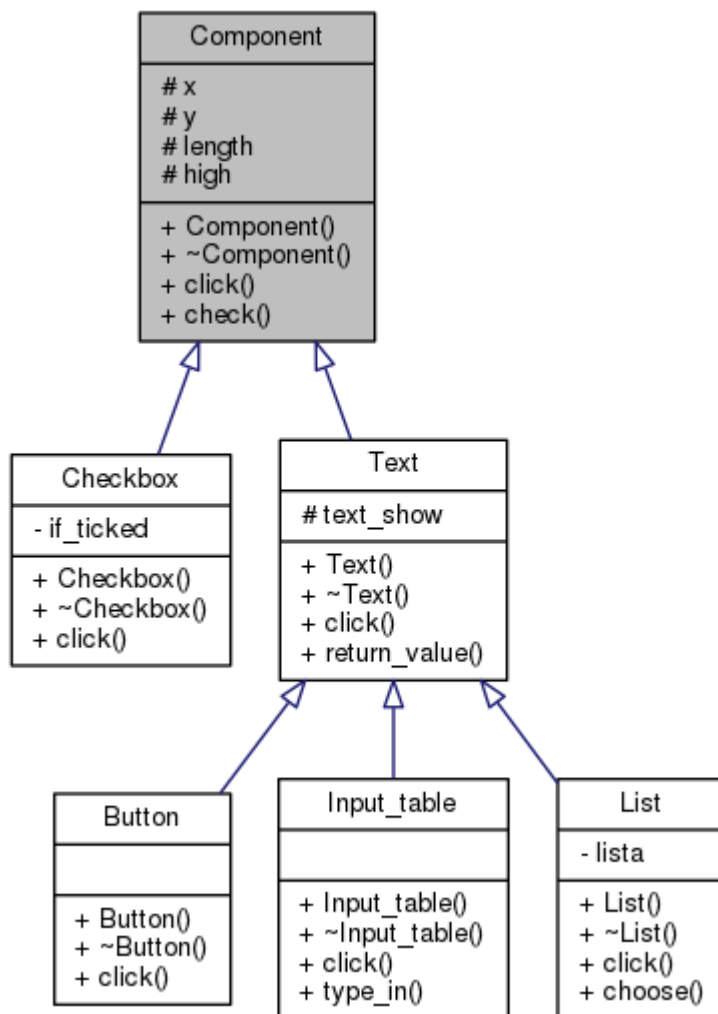
- pole tekstowe (klasa: Text)
- pole do wpisania tekstu (klasa: Input_table)
- checkbox (klasa: Checkbox)
- przycisk (klasa: Button)
- listę (klasa: List)

Wszystkie klasy są pochodną podstawowej klasy „Component” , z której dziedziczą podstawowe atrybuty i funkcje potrzebne każdej klasie, takie jak:

- współrzędne x, y
- długość i szerokość komponentu
- funkcja check() - sprawdza czy „kliknięto” na obiektami

- funkcja click() - jest inna dla każdej klasy (w klasie Component: click()=0)

Klasy dziedziczą po sobie atrybuty i funkcje według poniższego schematu.



Schemat dziedziczenia

Interfejs użytkownika:

Interfejs użytkownika składa się z kilku etapów:

1. Najpierw użytkownik wybiera z menu akcję, którą chce wykonać. Menu wygląda następująco:

Wybierz akcję do wykonania:

1. **Dodaj komponent**
2. **Usuń komponent**
3. **Kliknij**
4. **Zakończ**

2. Po wciśnięciu **Dodaj komponent** wyświetli się panel wyboru komponentu, który użytkownik chce dodać :

Wybierz komponent do edycji:

- 1. Pole z tekstem**
- 2. Przycisk**
- 3. Checkbox**
- 4. Pole do wpisania tekstu**
- 5. Lista**

3. Następnie użytkownik zostanie poproszony o wpisanie współrzędnych w jakich znajdować się ma na formularzu nowy komponent.

4. Użytkownik zostanie poproszony o wpisanie współrzędnych również w przypadku wybrania opcji **Usuń komponent** (służącej do wykasowania komponentu znajdującego się na podanych współrzędnych) oraz **Kliknij** (służącej do „kliknięcia” komponentu, czyli wymuszenia na nim określonej reakcji).

5. Natomiast po wybraniu opcji **Zakończ** działanie aplikacji zostanie zakończone (wszystkie obiekty (dynamicznie przydzielone), czyli komponenty zostaną zdealokowane).