

Komunikator III

Cel zajęć: poprawienie funkcjonalności poprzedniej wersji komunikatora.

Standard wiadomości:

1. Wiadomości mają formę słownika (dictionary).
2. Pola obowiązkowe:
 - a) **COMMAND** – określa typ wiadomości. Możliwe opcje:
 - **MESSAGE** – informuje że wiadomość zawiera wiadomość użytkownika do rozpropagowania
 - **JOIN** – dołączenie do czatu – bez tej komendy użytkownik nie będzie mógł wysyłać/odbierać wiadomości od/do innych użytkowników
 - **USERS** – komenda wysyłana przez serwer przekazująca listę aktualnie połączonych użytkowników
3. Pola opcjonalne
 - a) **MESSAGE** – zawiera treść wiadomości użytkownika – jest to pole obowiązkowe dla wiadomości typ MESSAGE
 - b) **NICK** – zawiera nick użytkownika, który wysłał wiadomość - jest to pole obowiązkowe dla wiadomości typ MESSAGE i JOIN
 - c) **USERS** – lista użytkowników - pole obowiązkowe dla wiadomości typ USERS

Innymi słowy: Przesyłane wiadomości muszą zawierać typ. Jeśli typ jest określony jako MESSAGE, czyli jest to po prostu wiadomość użytkownika, to wymagane jest, aby zawierała ona także pola MESSAGE i NICK. Jeśli pól tych nie ma, lub są puste, możemy uznać że wiadomość jest niepoprawna – serwer nie będzie jej propagował. Na tych zajęciach będziemy przysyłać tylko wiadomości użytkowników, więc w praktyce każde przesyłane wiadomości będą mieć ustawiony typ na wartość MESSAGE.

TODO:

1. Modyfikacja kodu serwera (wspólnie z prowadzącym):

- implementacja komend JOIN i USERS

2. Modyfikacja kodu klienta:

- implementacja komend JOIN i USERS

3. Modyfikacja GUI klienta:

a) podpięcie wysłania wiadomości pod klawisz ENTER:

```
self.parent.bind('<Return>', self.onEnter)
```

b) ustawienie widoku okienka wyświetlającego wiadomości tak, aby zawsze pokazywał ostatnie wiadomości:

```
self.text_output.see(END)
```

c) dodanie scrolla do okienka wyświetlającego wiadomości:

- dodanie elementu typu *Frame* który będzie kontenerem dla tego okienka i dla scrolla
- modyfikacja kodu tak aby wspomniane okienko znajdowało się w kontenerze (odpowiedni parent)
- stworzenie elementu typu *Scrollbar*

```
scroll = Scrollbar(frame, orient=VERTICAL,  
command=self.text_output.yview)
```

- podpięcie scrolla do okienka:

```
self.text_output['yscroll'] = scroll.set
```

- umieszczenie scrolla w kontenerze:

```
scroll.pack(side="right", fill="y")
```

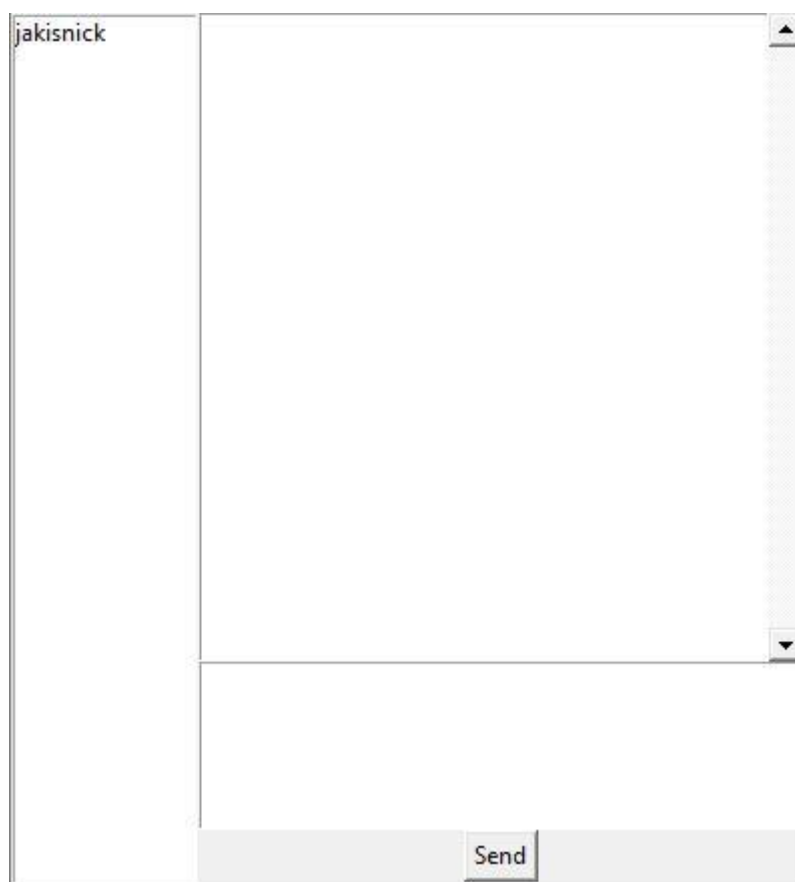
- umieszczenie kontenera w oknie

```
frame.pack()
```

d) Dodanie listy użytkowników:

- dodanie elementu *Listbox*
- umieszczenie go po lewej stronie okna programu

Ostatecznie program powinien wyglądać mniej więcej tak:



The image shows a simple chat window interface. It has a title bar at the top with the text "jakisnick" on the left and a standard window control button (a small square with a triangle) on the right. The main area is divided into two sections: a large text input area on the left and a smaller, empty rectangular area on the right. At the bottom of the window, there is a light gray bar containing a "Send" button.