

Opis

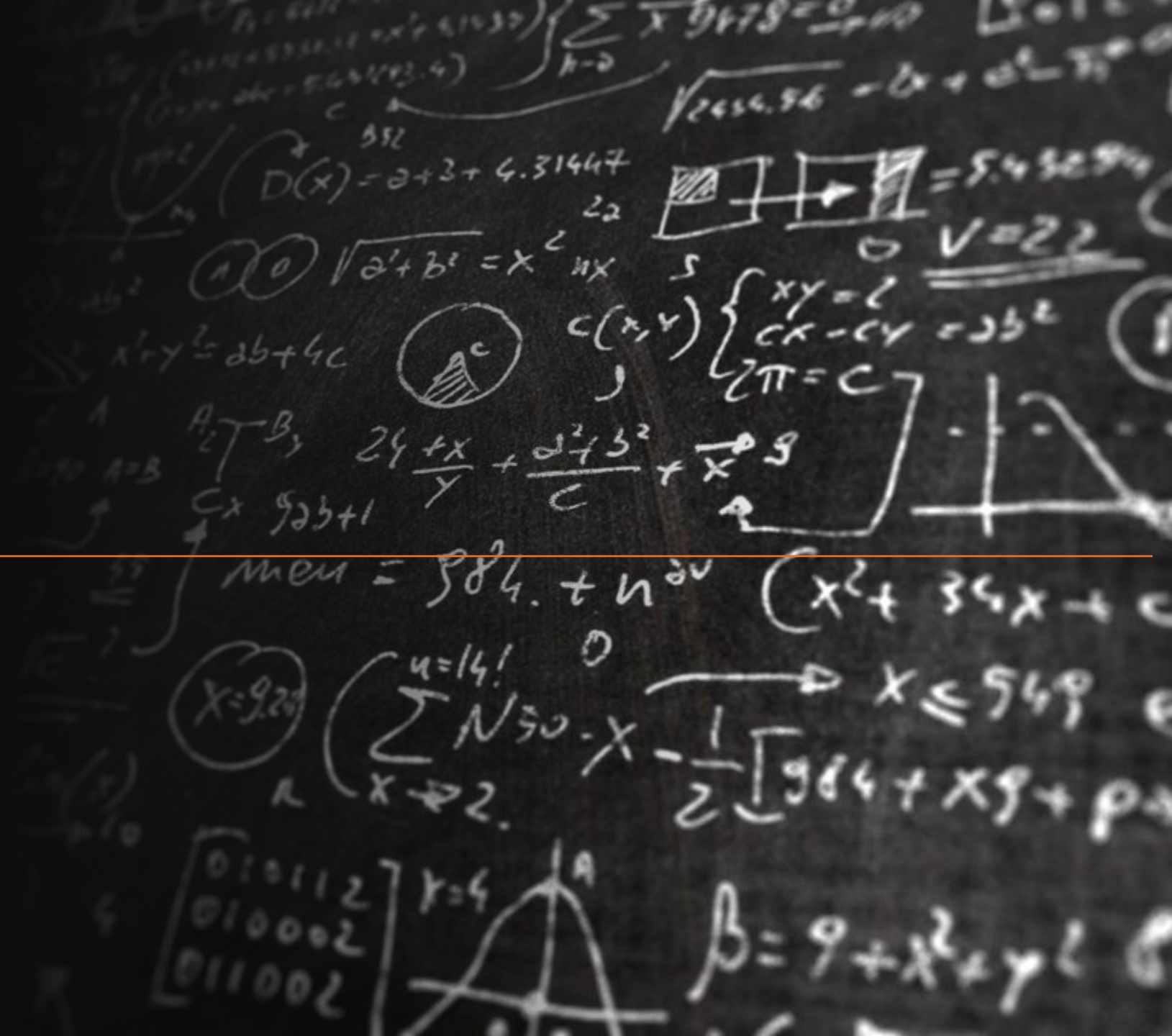
- Projekt składa się z dwóch głównych części: serwera i klienta. Serwer jest odpowiedzialny za przechowywanie i udostępnianie obiektów trzech klas (Kot, Pies, Ptak) na żądanie klientów. Klienci łączą się z serwerem, wysyłają swoje ID, a następnie mogą żądać list obiektów poszczególnych klas.



Struktura plikow:

- Server.java
- Client.java

Opis klas, oraz metod - Server.java



1. **Serwer:** Główna klasa serwera, odpowiedzialna za uruchamianie serwera, przyjmowanie połączeń i delegowanie obsługi klienta do osobnych wątków.

`main(String[] args):`
Uruchamia serwer,
nasłuchuje na porcie i
przyjmuje połączenia
klientów.

`dodajPrzykladoweObiekty():`
Dodaje przykładowe obiekty
do mapy obiektów.

`getObiektyByKlasa(String
className):` Pobiera
obiekty z mapy dla podanej
klasy.

2. Kot, Pies, Ptak: Klasy reprezentujące różne typy obiektów, które mogą być przechowywane na serwerze. Każda z tych klas implementuje Serializable i zawiera jedno pole (imie lub gatunek) oraz konstruktor inicjalizujący to pole.

- toString(): Zwraca reprezentację obiektu jako ciąg znaków.

3. ObsługaKlienta:

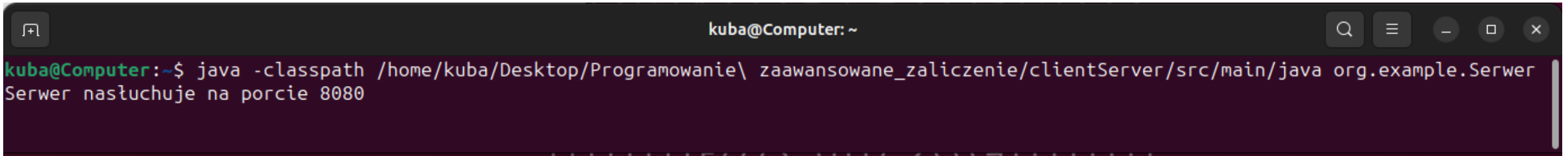
Klasa
implementująca
Runnable,
odpowiedzialna za
obsługę
pojedynczego
klienta.

run(): Metoda główna wątku,
która obsługuje komunikację
z klientem.

Przebieg zdarzen:



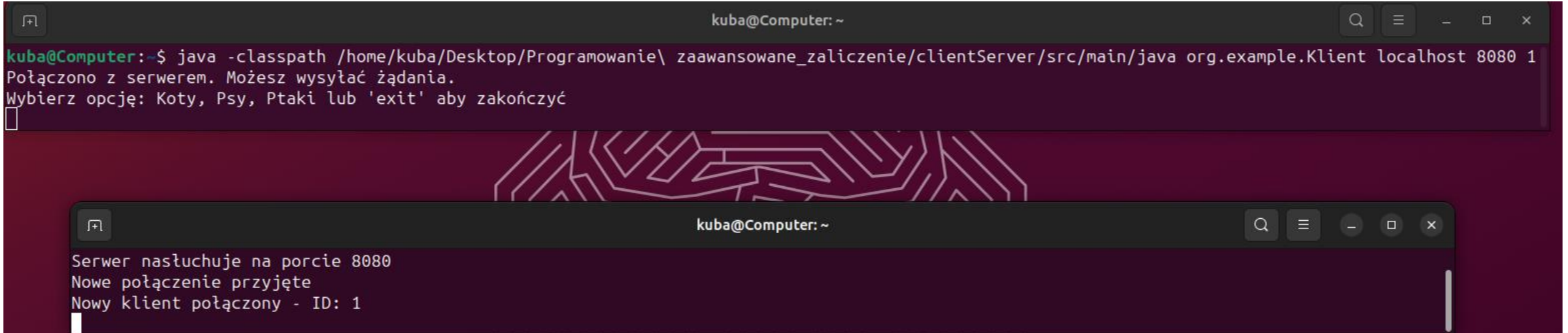
Inicjacja serwera:

A terminal window with a dark background. The title bar shows 'kuba@Computer: ~' and standard window controls. The command prompt shows the execution of a Java command to start a server. The output indicates the server is listening on port 8080.

```
kuba@Computer: ~  
kuba@Computer:~$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_zaliczenie/clientServer/src/main/java org.example.Serwer  
Serwer nasłuchuje na porcie 8080
```


Proba podlaczzenia sie
uzytkownika,
wraz z komunikatem na
serwerze

- * ID usera jest pobierane w komendzie:
- "java -classpath org.example.Klient localhost 8080 X",
- gdzie "8080" to port, a "X" jest numerem ID usera.



The screenshot shows two terminal windows. The top window is titled 'kuba@Computer: ~' and contains the following text:
kuba@Computer:~\$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_zaliczenie/clientServer/src/main/java org.example.Klient localhost 8080 1
Połączono z serwerem. Możesz wysyłać żądania.
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
[The cursor is on a new line.]

The bottom window is also titled 'kuba@Computer: ~' and contains the following text:
Serwer nasłuchuje na porcie 8080
Nowe połączenie przyjęte
Nowy klient połączony - ID: 1
[The cursor is on a new line.]

Przekroczenie ilości użytkowników.

```
kuba@Computer: ~/Desktop/Programowanie zaawansowane_zaliczenie/cli...
kuba@Computer:~/Desktop/Programowanie zaawansowane_zaliczenie/clientServer/src/main/java/org/example$ javac -Xlint:unchecked *.java
kuba@Computer:~/Desktop/Programowanie zaawansowane_zaliczenie/clientServer/src/main/java/org/example$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_zaliczenie/clientServer/src/main/java org.example.Serwer
Serwer nasłuchuje na porcie 8080
Nowe połączenie przyjęte
Nowy klient połączony - ID: 1
Nowe połączenie przyjęte
Nowy klient połączony - ID: 2
Nowe połączenie przyjęte
Nowy klient połączony - ID: 3
Nowe połączenie przyjęte
Nowy klient połączony - ID: 4
Nowe połączenie przyjęte
Nowy klient połączony - ID: 5
Nowe połączenie przyjęte
Odrzucono połączenie - przekroczono maksymalną liczbę klientów

```

```
kuba@Computer: ~$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_zaliczenie/clientServer/src/main/java org.example.Klient localhost 8080 6
Połączenie z serwerem zostało przerwane: Broken pipe
kuba@Computer: ~$
```

- W momencie przekroczenia userów (limit jest 5) wyświetla się odpowiedni komunikat po stronie serwera jak i usera

Przykładowe działanie:

```
kuba@Computer: ~/Desktop/Programowanie zaawansowane_zaliczenie/cli...
wane_zaliczenie/clientServer/src/main/java org.example.Serwer
Serwer nasłuchuje na porcie 8080
Nowe połączenie przyjęte
Nowy klient połączony - ID: 1
Nowe połączenie przyjęte
Nowy klient połączony - ID: 2
Nowe połączenie przyjęte
Nowy klient połączony - ID: 3
Nowe połączenie przyjęte
Nowy klient połączony - ID: 4
Nowe połączenie przyjęte
Nowy klient połączony - ID: 5
Nowe połączenie przyjęte
Odrzucono połączenie - przekroczono maksymalną liczbę klientów
Klient 5 wysłał żądanie: get_Koty
Wysłano kolekcję obiektów klasy koty do klienta 5
Klient 1 wysłał żądanie: get_Psy
Wysłano kolekcję obiektów klasy psy do klienta 1
Klient 1 wysłał żądanie: get_Ptaki
Wysłano kolekcję obiektów klasy ptaki do klienta 1
Klient 3 wysłał żądanie: get_Psy
Wysłano kolekcję obiektów klasy psy do klienta 3
Klient 3 wysłał żądanie: exit

```

```
kuba@Computer: ~$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_
zaliczenie/clientServer/src/main/java org.example.Klient localhost 8080 1
Połączono z serwerem. Możesz wysyłać żądania.
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
Psy
Otrzymano kolekcję obiektów Psy: [Pies{imie='Burek'}, Pies{imie='Azor'}]
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
Ptaki
Otrzymano kolekcję obiektów Ptaki: [Ptak{gatunek='Wróbel'}, Ptak{gatunek='Sikork
a'}]
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć

```

```
kuba@Computer: ~$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_
zaliczenie/clientServer/src/main/java org.example.Klient localhost 8080 3
Połączono z serwerem. Możesz wysyłać żądania.
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
Psy
Otrzymano kolekcję obiektów Psy: [Pies{imie='Burek'}, Pies{imie='Azor'}]
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
exit
kuba@Computer: ~$

```

```
kuba@Computer: ~$ java -classpath /home/kuba/Desktop/Programowanie\ zaawansowane_
zaliczenie/clientServer/src/main/java org.example.Klient localhost 8080 5
Połączono z serwerem. Możesz wysyłać żądania.
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć
Koty
Otrzymano kolekcję obiektów Koty: [Kot{imie='Mruczek'}, Kot{imie='Filemon'}]
Wybierz opcję: Koty, Psy, Ptaki lub 'exit' aby zakończyć

```



Dziękuję za uwagę!

Wykonał Jakub Bialek C1 159815

