

Blog

Treść zadania

Chcemy stworzyć architekturę dla bloga internetowego. Blog ma swoich użytkowników, którzy mogą tworzyć posty i komentować posty innych użytkowników. Potrzebujemy następujących klas:

- **Blog**, który zawiera listę użytkowników i postów. Ponadto ma metody umożliwiające dodanie nowego posta, skomentowanie posta i wyświetlenie wszystkich wpisów dla danego użytkownika.
- **User**, który zawiera id, imię, nazwisko oraz nick użytkownika.
- **Post**, który zawiera id, datę dodania, autora, zawartość oraz listę komentarzy.
- **Comment**, który zawiera id, datę dodania, autora oraz zawartość.

Jako że **Post** różni się od **Comment** tylko tym, że posiada komentarze, to zgodnie z zasadą DRY chcemy, aby obie te klasy dziedziczyły z innej, abstrakcyjnej klasy.

Ponadto, id dla każdego użytkownika i wpisu ma być unikalne - stworzona architektura powinna ustawiać nowo tworzone obiekty kolejne id.

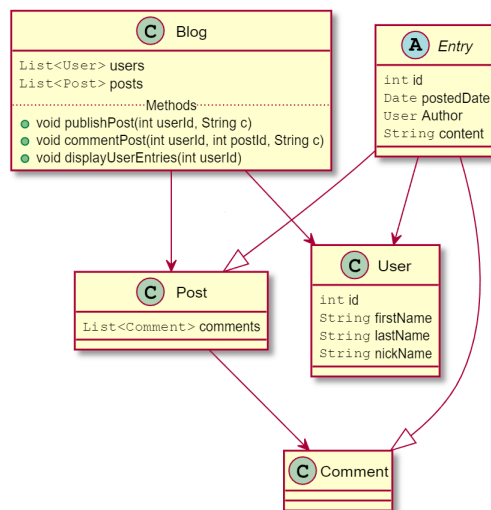
Przykładowy output:

```
janeK95 posted just now: Żyrafa najwyższą formą życia.
xxmichuXX posted just now: Czas nie zawsze przemija w jednakowym rytmie. To my o tym decydujemy.
marian_dev posted just now: Jaki jest obiektowy sposób na zdobycie fortuny? Dziedziczenie!
Cannot publish post - user of id 100 doesn't exist.

marian_dev commented janeK95's post: Najwyższą formą życia jest żyrafa stojąca na taborecie!
janeK95 commented xxmichuXX's post: To bardzo prawdziwe. Chyba cytuję z Paulo Coelho.
xxmichuXX commented xxmichuXX's post: Mądre słowa.
janeK95 commented marian_dev's post: Tak, a grupa krwi programisty to C++!
xxmichuXX commented marian_dev's post: Haha, usmiałem się!

janeK95's entries:
(POST) Żyrafa najwyższą formą życia.
(COMMENT) To bardzo prawdziwe. Chyba cytuję z Paulo Coelho.
(COMMENT) Tak, a grupa krwi programisty to C++!
```

Diagram UML



Uwagi

1. Należy zadbać o podstawowe zasady programowania obiektowego - enkapsulację, gettery, settery, odpowiednią dostępność zmiennych... Te rzeczy nie są uwzględnione w treści zadania ani diagramie, ale są to dobre praktyki, które należy stosować.