

1. Charakterystyka problemu

Klient potrzebuje komponentu pozwalającego na wyświetlanie różnych statystyk zbieranych z dowolnego sprzętu.

2. Wizja produktu

Komponent powinien być graficznym oknem, które można osadzić w aplikacji graficznej (o zadanym przez klienta typie). Powinien przyjmować dane, które następnie wyświetla jako wykres słupkowy. Jeden słupek ma reprezentować jeden zbiór różnych (dodatnich) danych liczbowych (traktowanych jako zebrane w jednej jednostce czasu), a każdy kolejny słupek powinien w tej samej kolejności odzwierciedlać kolejny zbiór tych samych danych. W komponencie można zamieścić legendę do danych.

3. Zakres funkcjonalności

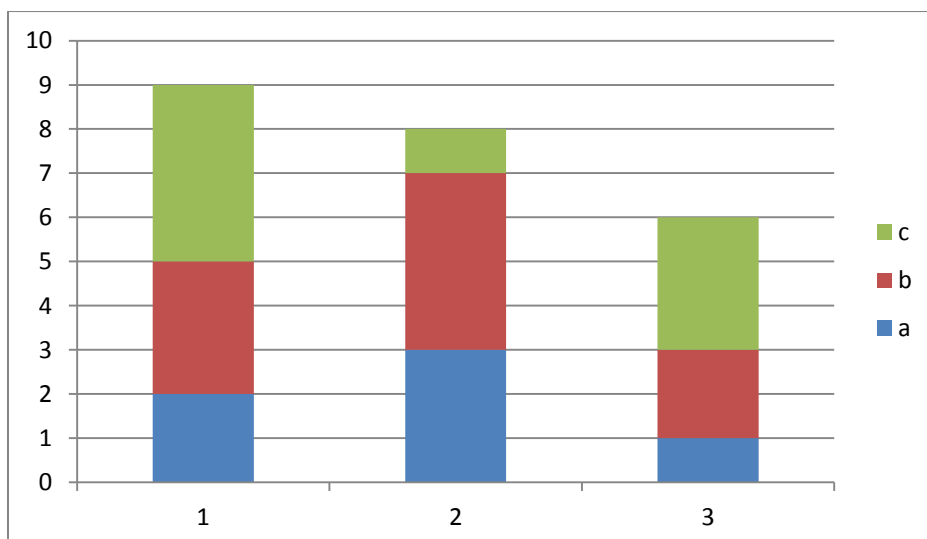
3.1. wymagania funkcjonalne

- odbieranie danej typu: (czas, rodzaj danej, wartość) i wyświetlenie jej na wykresie
- odbieranie zbioru danych typu: (czas, rodzaj danej, wartość) i uzupełnienie nim istniejącego wykresu
- dane wstawiane mogą uzupełniać wcześniejsze pomiary, nie tylko ostatni
- wyświetlenie legendy
- automatyczne skalowanie osi
- konfigurowalna liczba wyświetlanych elementów na osi X (długość historii)
- automatyczne kolorowanie zbiorów danych

3.2. wymagania niefunkcjonalne

- obsługa sytuacji, w której dodano niepasujące dane
- ochrona przed przepełnieniem

4. Przykład rysowanego wykresu



5. Architektura komponentu

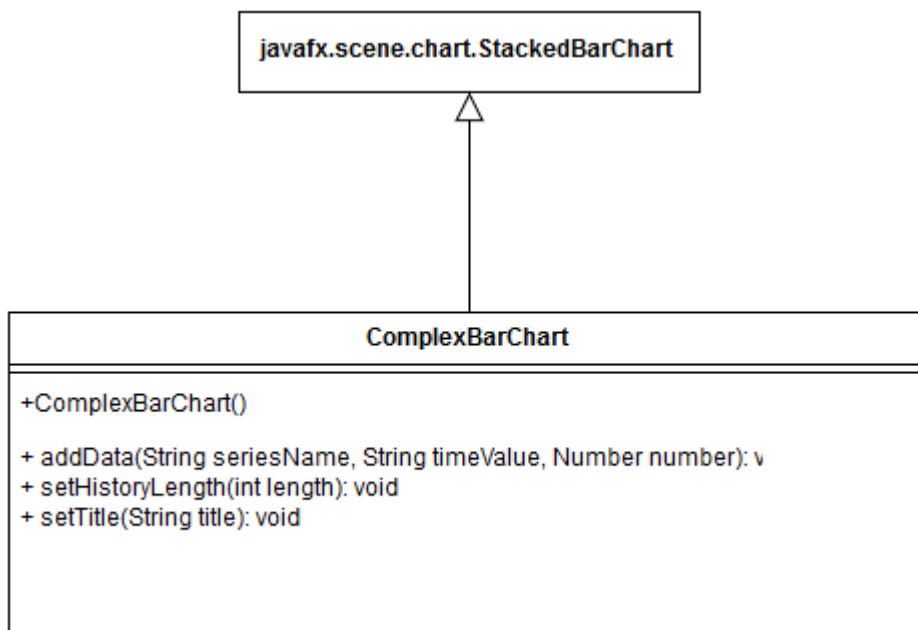
- 5.1. Komponent zawiera klasę, która dostarcza implementacji interfejsu **IComplexBarChart** oraz dziedziczy po klasie

javafx.scene.chart.StackedBarChart<java.lang.String, java.lang.Number>

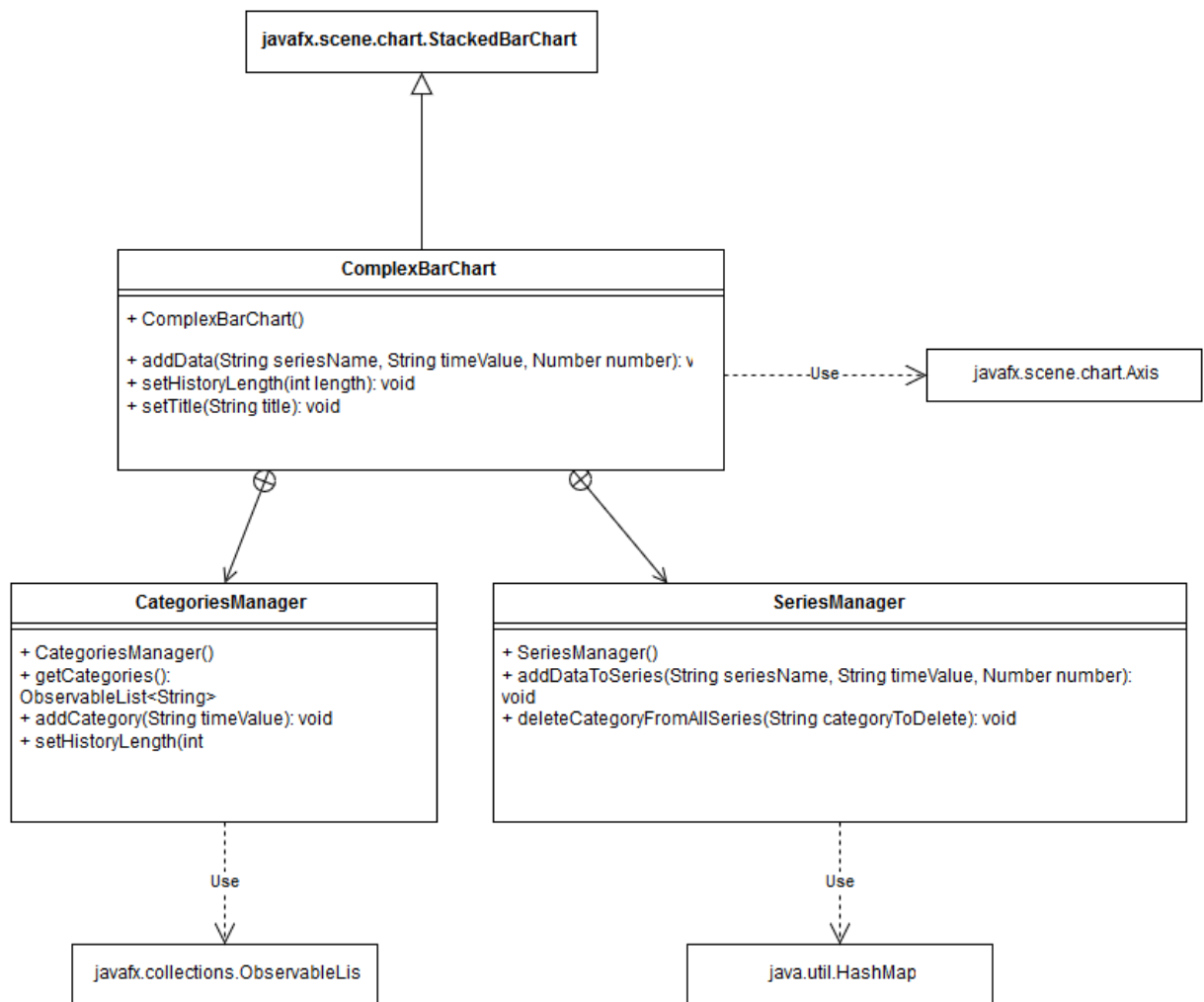
Jest więc elementem graficznym biblioteki JavaFX (dziedziczy po **javafx.scene.Parent**) i może być umieszczona na dowolnej scenie (javafx.scene.Scene).

- 5.2. **timeValue** w funkcji addData jest formatu String i poza szczególnymi przypadkami powinien być podawany w formacie **RRRR-MM-DD-GG-MM-SS**

5.3.



6. Szczegółowa architektura komponentu



Wewnątrz komponentu powstają dwie nowe klasy, **CategoriesManager** i **SeriesManager**, które zajmują się obsługą odpowiednio kategorii (czas na osi x) oraz serii danych (jeden kolor w każdej kolumnie) na wykresie.

Pozostałe używane klasy są także przedstawione na wykresie.

7. Przykład użycia komponentu

Komponent może być użyty w aplikacji JavaFX (`javafx.application.Application`) jako element do utworzenia sceny.

```
ComplexBarChart sbc = new ComplexBarChart();
sbc.setHistoryLength(7);
sbc.setTitle("Inside Component Title");

Scene scene = new Scene(sbc, 800, 600);
stage.setScene(scene);
stage.show();
```

Dodawanie danych do wykresu wykorzystuje funkcję **`void addData(String seriesName, String timeValue, Number number)`**. Należy jednak pamiętać, że wewnątrz aplikacji `javafx.application.Application` operacje na danych używanych przez komponenty graficzne mają swoje ograniczenia. Proponowany sposób wywoływania funkcji wygląda następująco:

```
Platform.runLater(() -> {
    sbc.addData("seria3", "1", 60);
    sbc.addData("seria3", "2", 60);
    sbc.addData("seria3", "5", 60);
});
```

Szczególną uwagę należy zwrócić na format kategorii danych. **`timeValue`** w funkcji `addData` jest formatu `String`, co pozwala na wygodną interpretację kolejnych kolumn, nie tylko jako kolejne punkty czasowe. Pamiętać należy jednak, że kolumny sortowane są po ich nazwie jako `String`, co wymusza format czasu **`RRRR-MM-DD-GG-MM-SS`** lub inny, który pozwoli zachować odpowiednią kolejność danych podczas działania komponentu (np. przy założeniu, że dane pochodzą tylko z jednego dnia, możemy opuścić część odpowiadającą za rok, miesiąc i dzień).

8.