

Laboratorium 4 - Refaktoryzacja

Jakub Solecki
Krzysztof Nalepa
Bartosz Szar

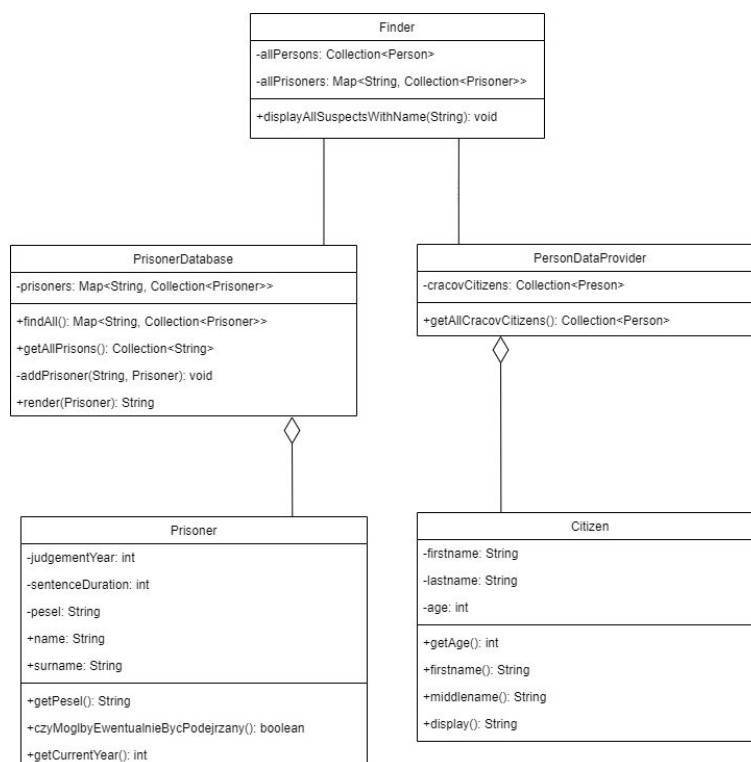
Za główną formę dokumentacji pracy z kodem uznaliśmy diagram klas UML dokładnie odzwierciedlający jego stan pod koniec każdego kroku.

Krok 1

Proponowane zmiany w kodzie:

1. Zmiana nazw atrybutów na bardziej zwięzłe
2. Zmiana nazw metod na bardziej zwięzłe i precyzyjne
3. Zmiana atrybutów publicznych na prywatne
4. Ujednolicenie nazewnictwa (wyłącznie angielskie nazwy)
5. Dodanie getterów dla pól prywatnych które tego wymagają
6. Zamiana zbędnych metod statycznych (np metoda render w klasie PrisonersDatabase) na odpowiednie metody w konkretnych klasach.
7. Wprowadzenie generalizacji odpowiednich klas

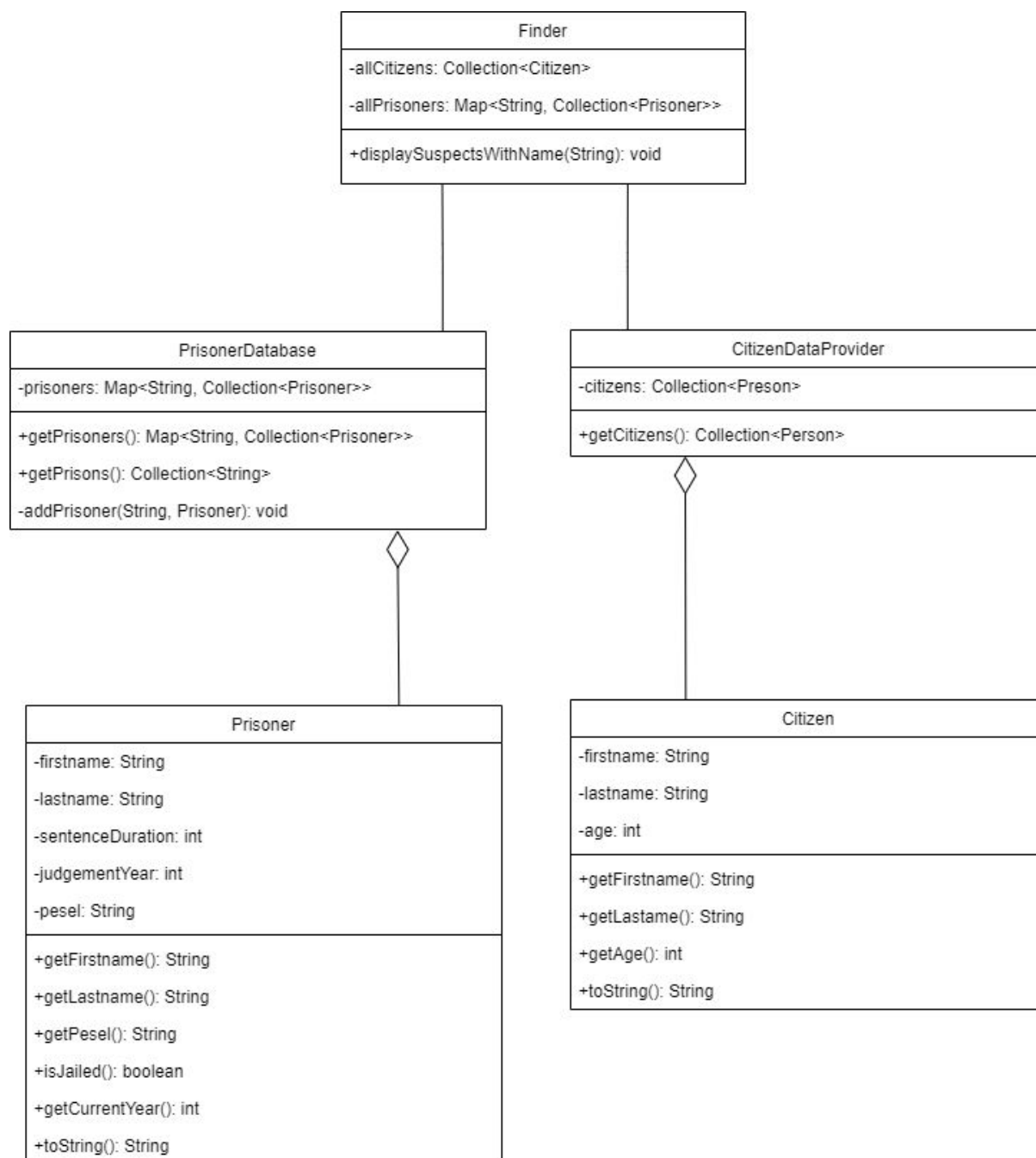
Diagram UML stworzony na podstawie analizy kodu:



Krok 2

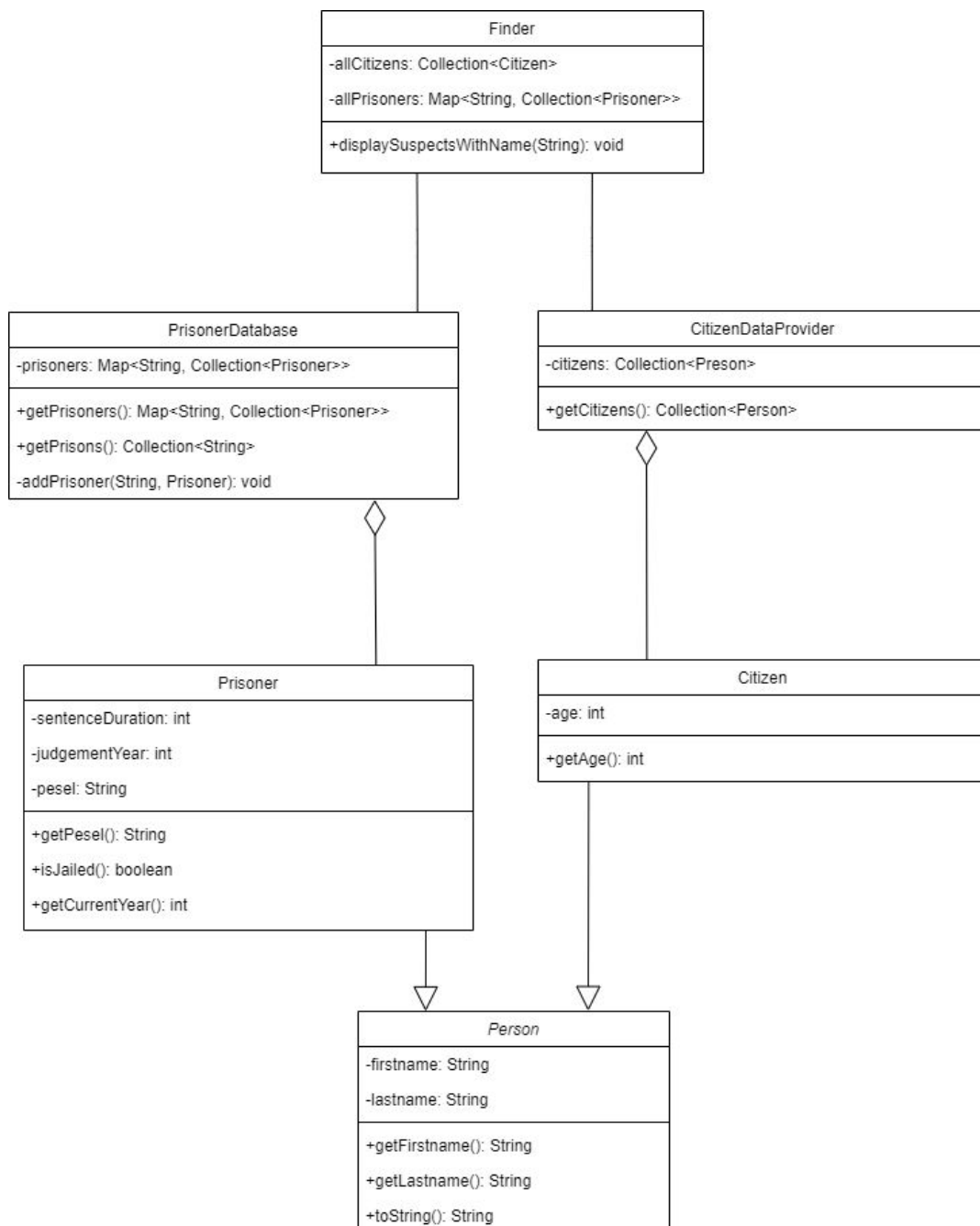
W tym kroku poprawiliśmy błędy zidentyfikowane w kroku 1.

Diagram UML stworzony na podstawie zaproponowanych zmian:



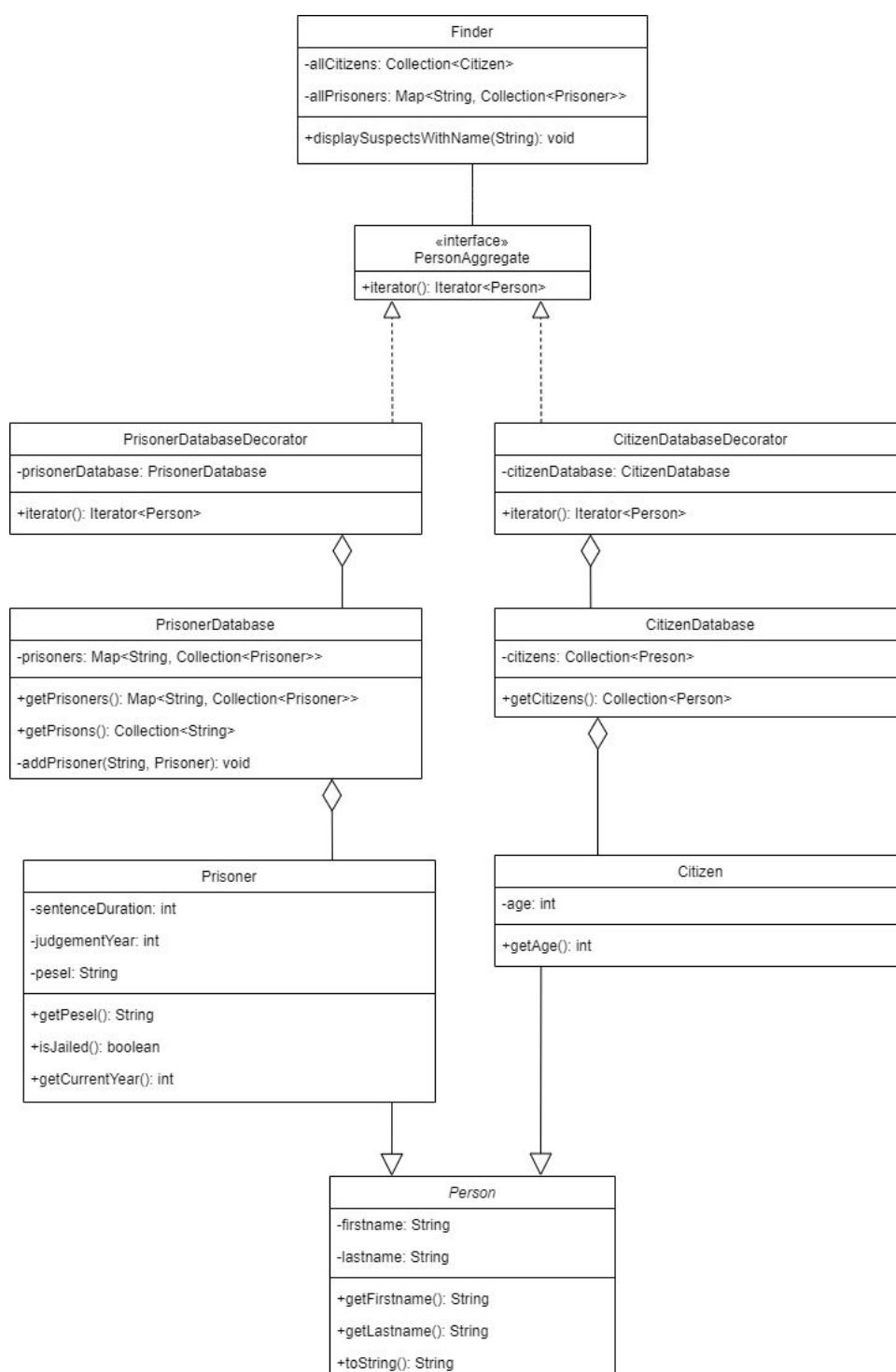
Krok 3

Wprowadziliśmy generalizację klas Prisoner oraz Citizen. Od teraz dziedziczą one po klasie abstrakcyjnej Person, zawierającej wspólne dla obu podklas atrybuty oraz metody (getterzy dla tych atrybutów). Aktualną strukturę obrazuje poniższy diagram UML:



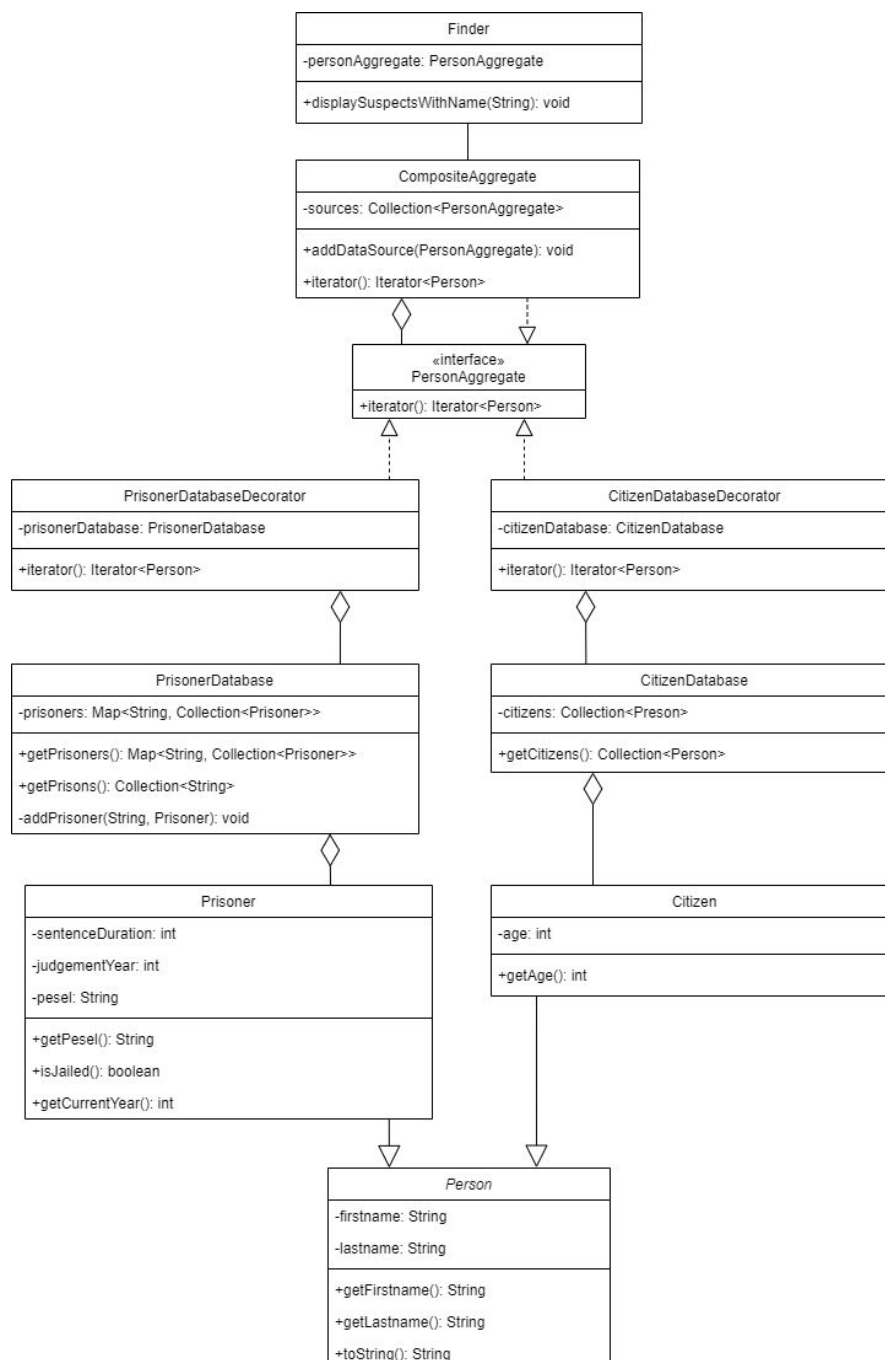
Krok 4

W 4 kroku użyliśmy wzorca dekorator aby wprowadzić generalizację klas dostarczających dane do systemu. Stworzyliśmy klasy PrisonerDatabaseDecorator i CitizenDatabaseDecorator które “dekorują” klasy przechowujące dane dodając do nich implementacje metody zwracającą iterator definiowaną przez interfejs PersonAggregate.



Krok 5

Zmiany wprowadzone w poprzednim kroku nie były optymalne. W celu poprawienia tego zaimplementowaliśmy klasę pośrednią CompositeAggregate, której zadaniem jest zarządzanie iteratorami zwróconymi przez dekoratory klas dostarczających dane. Klasa CompositeAggregate zwraca “wspólny” iterator pozwalający uzyskać dostęp do danych dostarczonych przez wszystkie klasy implementujące interfejs PersonAggregate



Krok 6

W ostatnim kroku usprawniliśmy wyszukiwanie podejrzanych, biorąc pod uwagę konkretne parametry takie jak wiek, imię, czy informacja o tym czy osoba, której dotyczy wyszukiwanie odbywa obecnie wyrok więzienia. Modyfikacja polegała na dodaniu interfejsu SearchStrategy oraz wyspecjalizowanych klas go implementujących. Wszystkich zmian w tym kroku dokonaliśmy w oparciu o wzorec projektowy Composite.

