

Specyfikacja implementacyjna programu
PrisonOptimizer

Krzysztof Maciejewski Seweryn Bieńko

12 sierpnia 2020

Spis treści

1	Wstęp	2
2	Środowisko deweloperskie	2
2.1	Środowisko deweloperskie Krzysztofa	2
2.2	Środowisko deweloperskie Seweryna	3
3	Zasady wersjonowania	3
4	Diagram klas	4
5	Istotne struktury	5
5.1	Prison	5
6	Najważniejsze algorytmy	5
6.1	Algorytm genetyczny	5
6.2	Funkcja przystosowania	6

1 Wstęp

Dokument ten jest technicznym opisem programu *PrisonOptimizer*. Przybliża on proces tworzenia programu oraz przedstawia zarys stosowanych algorytmów i struktur danych.

2 Środowisko deweloperskie

W tej sekcji przedstawione są elementy środowiska pracy, w którym będzie powstawał program *PrisonOptimizer*.

2.1 Środowisko deweloperskie Krzysztofa

Parametry komputera:

- Procesor: Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz
- Pamięć RAM: 16 GB
- System operacyjny: Windows 10.0.18362

Pozostałe parametry:

- Język programowania: Java (wersja: "13" 2019-09-17)
- Maszyna wirtualna: Java HotSpot(TM) 64-Bit Server VM (build 13+33, mixed mode, sharing)
- Środowisko programistyczne: IntelliJ IDEA 2019.2.4 (Community Edition)

2.2 Środowisko deweloperskie Seweryna

Parametry komputera:

- Procesor: Intel Core i5-7300HQ
- Pamięć RAM: 16 GB
- Dyski wewnętrznej pamięci - 2 dyski SSD o wielkości 1TB,
- System operacyjny: Windows 10.0.17763

Pozostałe parametry:

- Język programowania: Java (wersja: "8")
- Środowisko programistyczne: IntelliJ IDEA 2019.3.0 (Community Edition)

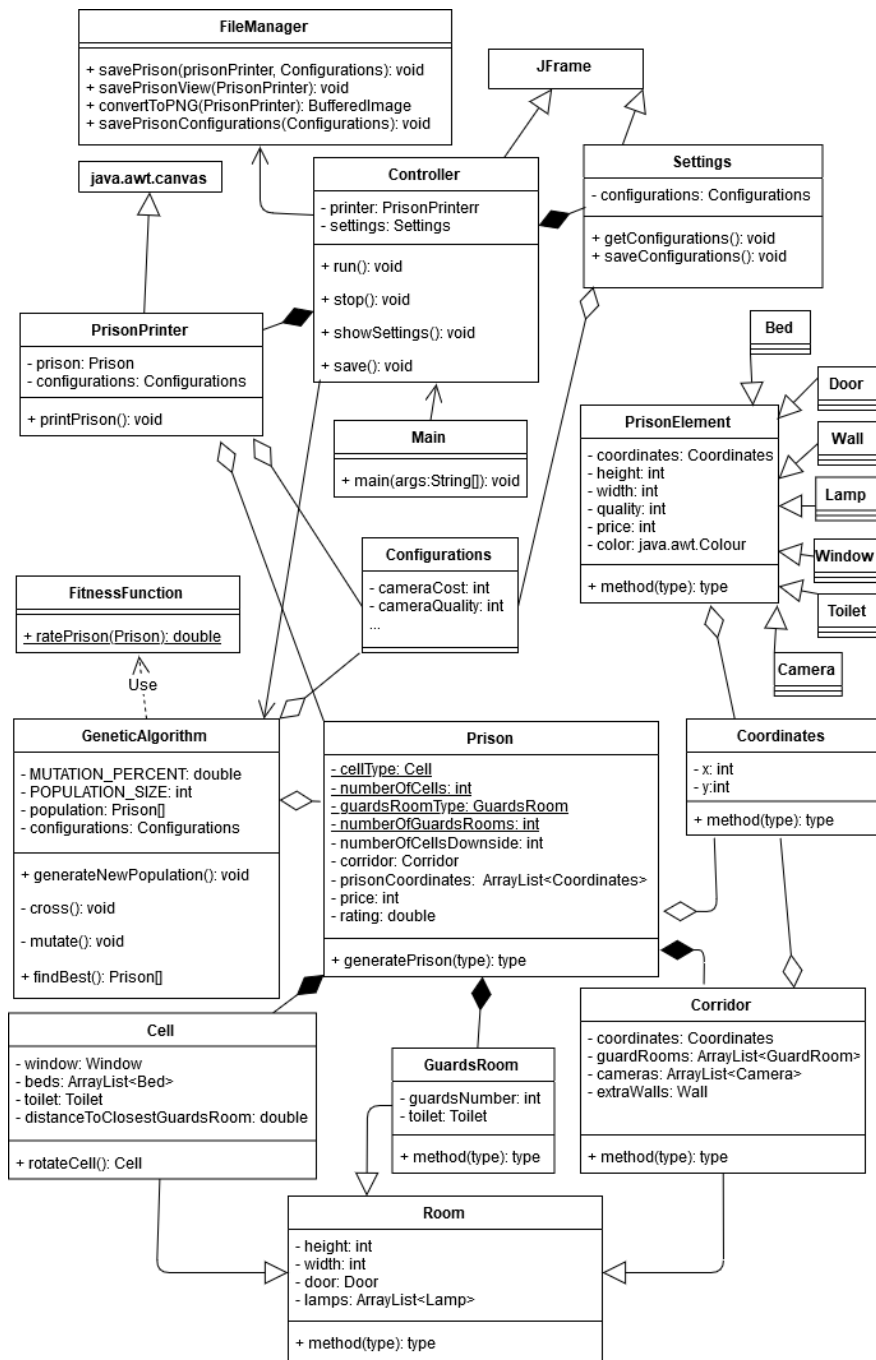
3 Zasady wersjonowania

Ze względu na charakter pracy zespołowej w tym projekcie, przewiduje się utworzenie dwóch gałęzi roboczych dla każdego członka zespołu. Nazwa gałęzi danego członka to jego nazwa użytkownika w Wirtualnym Dziekanacie. Każda wersja, utworzona przez członka, będzie umieszczana w tej gałęzi. Dodatkowo w momencie umieszczenia w gałęzi roboczej wersji stabilnej, czyli takiej która się kompiluje i poprawnie realizuje swoją funkcjonalność, zostanie ona jednocześnie umieszczona w głównej gałęzi *master*.

Konwencja nazewnicza wersji:

1. Wersje deweloperskie programu, przed uzyskaniem pierwszej wersji stabilnej, będą w formacie: `0.X - dev`, gdzie `X` to numer wersji deweloperskiej.
2. Pierwsza wersja stabilna będzie oznaczona: `1.0`.
3. Poprawki to stabilnej wersji będą oznaczane: `1.Y`, gdzie `Y` to numer poprawionej wersji.

4 Diagram klas



Rysunek 1: Diagram klas.

Krótki opis poszczególnych klas:

- Klasa *PrisonElement* reprezentuje element więzienia. Rozszerzają ją następujące klasy: *Bed*, *Door*, *Wall*, *Lamp*, *Window*, *Toilet*, *Camera*.
- Klasa *Coordinates* przechowuje parę liczb będących współrzędnymi. Wykorzystywana jest do określania pozycji obiektów.
- Klasa *Room* reprezentuje pokoje w więzieniu — korytarz, cele oraz pokoje Służby Więziennej. Rozszerzają ją klasy: *Corridor*, *GuardsRoom*, *Cell*.
- Klasa *Prison* reprezentuje więzienie.
- Klasa *GeneticAlgorithm* przechowuje populację i posiada metody realizujące algorytm genetyczny.
- Klasa *FitnessFunction* posiada statyczną metodę reprezentującą funkcję przystosowania, która ocenia osobników.
- Klasa *PrisonPrinter* służy do wyświetlania więzienia.
- Klasa *Controller* reprezentuje okno początkowe interfejsu graficznego.
- Klasa *Settings* reprezentuje okno ustawień interfejsu graficznego.
- Klasa *Configurations* przechowuje wartości parametrów możliwych do zmiany z okna ustawień.
- Klasa *FileManager* służy do zapisu informacji do pliku.

5 Istotne struktury

5.1 Prison

Klasa *Prison* reprezentuje więzienie — zawiera wszystkie elementy składowe więzienia oraz ich położenie. Instancje tej klasy wykorzystywane są jako osobniki w algorytmie genetycznym.

6 Najważniejsze algorytmy

6.1 Algorytm genetyczny

Algorytm genetyczny zalicza się do grupy algorytmów ewolucyjnych. Wzorowany jest na biologicznej ewolucji i jest często wykorzystywany do zadań optymalizacyjnych.

W algorytmie genetycznym mamy populację złożoną z osobników posiadających genotyp, czyli zbiór informacji.

Przebieg algorytmu wygląda następująco:

1. Losowana jest pewna populacja początkowa.
2. Populacja poddawana jest ocenie..
3. Osobniki z najlepszą oceną biorą udział w procesie reprodukcji.
4. Genotypy wybranych osobników poddawane są operacjom krzyżowania lub mutacji:
 - Krzyżowanie polega na łączeniu genotypu dwóch osobników u ich potomka.
 - Mutacja wprowadza do genotypu losowe zmiany.
5. Powstaje nowe pokolenie. Algorytm powraca do punktu drugiego, dopóki nie wykona się zadaną liczbę iteracji.

W programie *PrisonOptimizer* osobnikiem jest obiekt *Prison*, a jego genotyp wyznaczają wartości jego atrybutów.

6.2 Funkcja przystosowania

Funkcja przystosowania będzie wykorzystywana do oceny każdego osobnika. Następnie, w procesie selekcji, faworyzowane będą najlepiej osobniki z najlepszą oceną i to one staną się "rodzicami" dla populacji. Użyta w programie *PrisonOptimizer* funkcja przystosowania będzie oceniała osobnika na podstawie następujących kryteriów:

- Ilość lamp i powierzchnia którą obejmują;
- Ilość kamer i powierzchnia którą obejmują;
- Odległość z pokoiów Służby Więziennej do cel;
- Jakość elementów więzienia;
- Istnienie drogi do wyjścia z każdego punktu więzienia.