1. Wstęp
   1. Cel i założenia projektu

Celem niniejszej pracy dyplomowej jest stworzenie aplikacji generującej plan zajęć dla szkół. Aplikacja ma za zadanie wygenerować plan zajęć nie posiadający żadnych defektów[[1]](#footnote-2).

1. Zasada działania algorytmu genetycznego

Algorytm genetyczny (AG) jest jedną z wielu metod zaliczanych do nurtu ewolucyjnego. Wykorzystuje się go głównie do znalezienia optymalnego rozwiązania danego problemu. Algorytm ten przypomina proces ewolucji z biologicznego punktu widzenia, ponieważ dany proces nie polega na stworzeniu właściwego rozwiązania (wykorzystując przy tym szeregu, często bardzo skomplikowanych, operacji), lecz na „wyhodowaniu” go z pośród wielu alternatywnych rozwiązań. Taki proces polega na porównywaniu ze sobą wielu rozwiązań, które poddawane są niewielkim zmianą, w celu stworzenia coraz to lepszego rozwiązania.

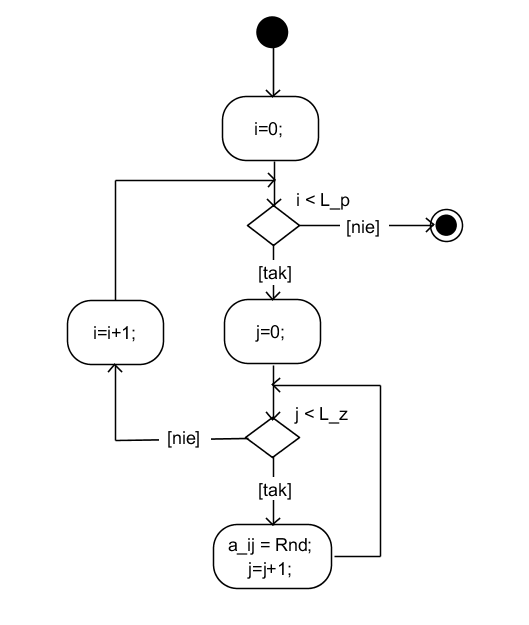
* 1. Podstawowe pojęcia

**Osobnik** jest to pojedynczy obiekt podlegający ewolucji. Celem ewolucji jest jak najlepsze przystosowanie osobnika do otaczającego go środowiska. W przypadku AG osobnikiem możemy nazwać rozwiązanie danego problemu.

**Populacja** jest to zbiór osobników o stałym rozmiarze, która w kolejnych cyklach ewolucji zastępuje obecne osobniki nowo stworzonymi.

* 1. Podstawowe operacje
     1. Generowanie populacji

Generowanie populacji jest procesem polegającym na wygenerowaniu danej ilości osobników danej populacji w sposób losowy bądź deterministyczny.



gdzie:

L\_p – ilość osobników w populacji

L\_z – ilość zmiennych w osobniku

Rnd – losowa wartość zmiennej

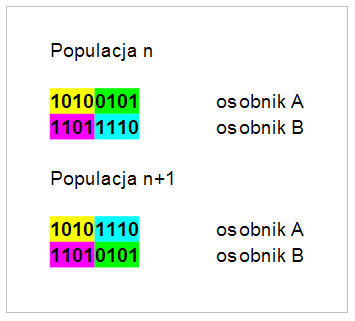
a – populacja

a\_i – osobnika

a\_ij – zmienna

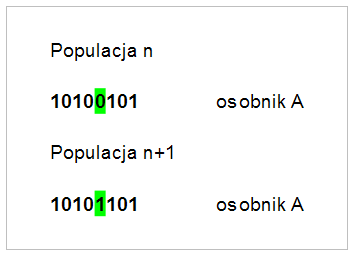
* + 1. Krzyżowanie

Krzyżowanie to proces polegający na stworzeniu nowego osobnika, wykorzystując do tego dwóch innych osobników.



* + 1. Mutacja

Mutacja to proces polegający na dokonaniu niewielkich zmian na istniejącym osobniku.



* + 1. Ocena populacji

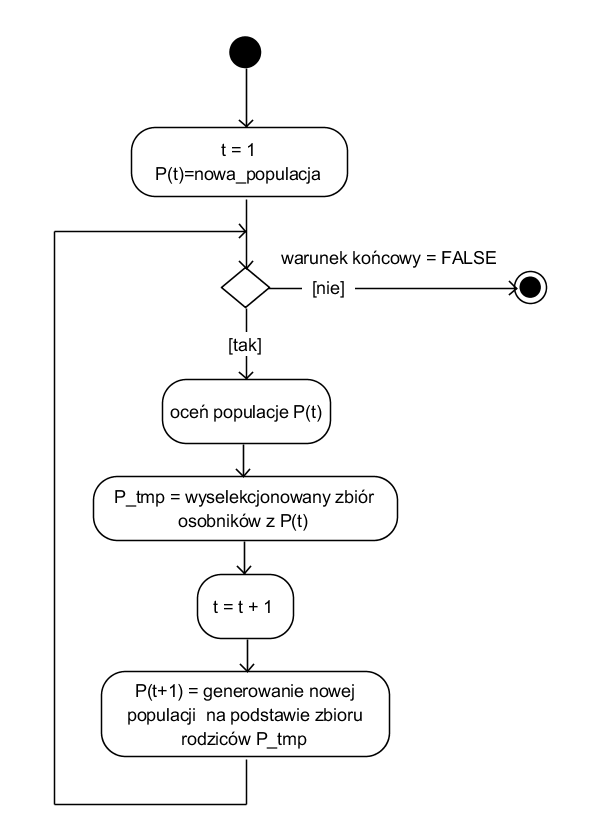
Ocena populacji jest to indywidualny stopień przystosowania każdego z osobników w populacji. Za ocenę odpowiada funkcja przystosowania, która podczas oceny bierze pod uwagę kilka czynników.

* + 1. Selekcja

Selekcja jest to proces tworzenia nowej populacji z obecnie istniejącej, wykorzystując do tego proces mutacji oraz krzyżowania. Każdą populacje można podzielić na osobników silnie, średnio i słabo przystosowanych. W przypadku AG taki podział jest istotny, ponieważ przynależność osobnika do danej grupy określa jego stopień przydatności do powstania nowej populacji.

Wykorzystanie osobników z grupy:

1. Silnie przystosowani:
   * reprodukcja,
   * krzyżowanie z osobnikami ślinie i średnio przystosowanymi,
   * mutacja.
2. Średnio przystosowani:
   * krzyżowanie z osobnikami ślinie i średnio przystosowanymi,
   * mutacja.
3. Słabo przystosowani:
   * porzucenie.
   1. Przebieg ewolucji



gdzie:

t – krok czasowy (epoka)

P – populacja

P\_tmp – lista wyselekcjonowanych osobników

warunek końcowy – warunek ograniczający czas trwania ewolucji

1. Specyfikacja Problemu
   1. Cel problemu

Plan zajęć musi się składać z planów klas, nauczycieli oraz sal lekcyjnych. Aplikacja ma za zadanie wygenerować plan zajęć nie posiadający defektów takich jak:

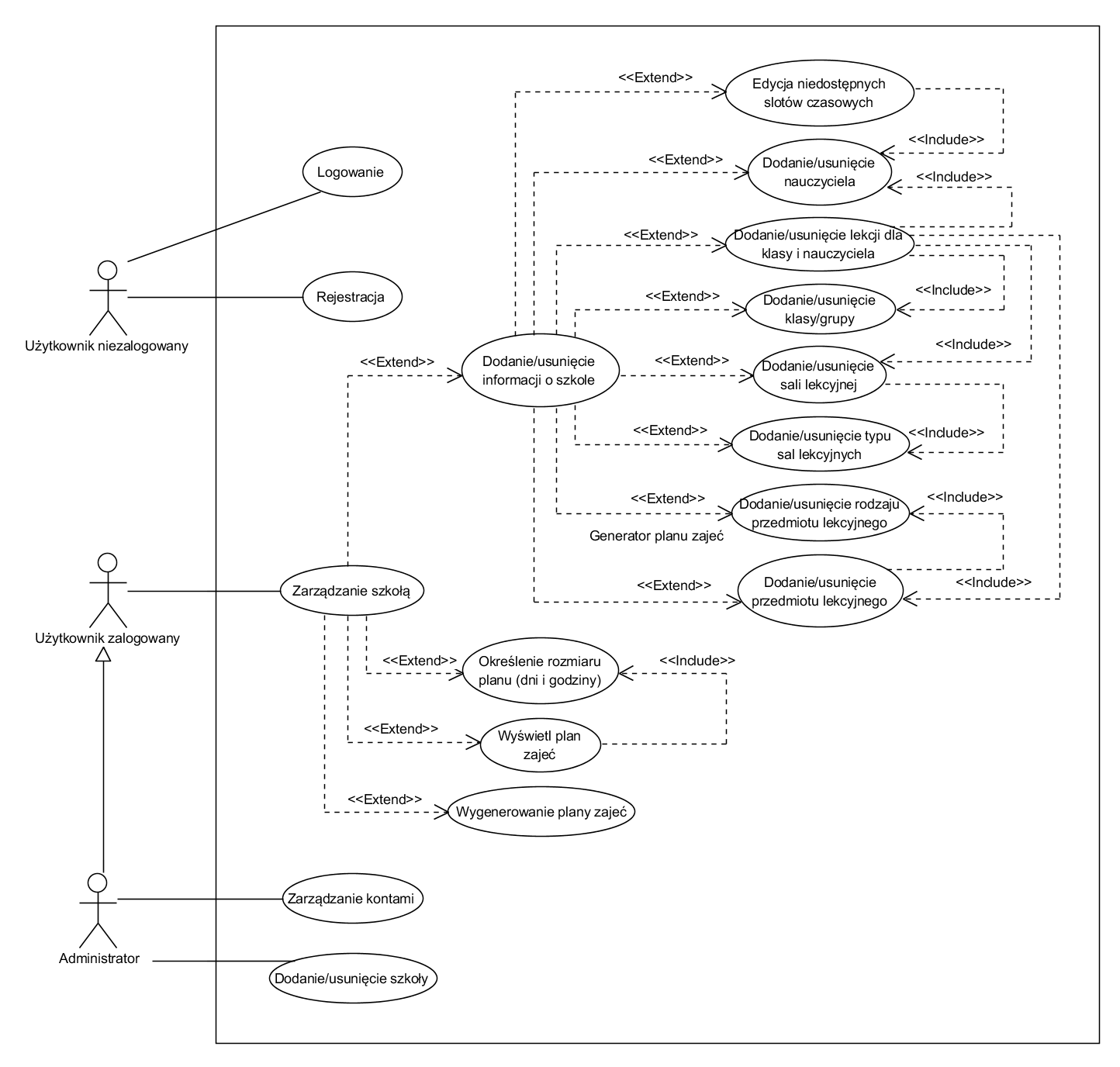
* więcej niż jednej lekcji w jednym slocie czasowym,
* braku lekcji w planie,
* lekcji w nieodpowiedniej sali lekcyjnej (zły typ sali oraz niewystarczająca ilość miejsc).

Najważniejszymi cechami planu zajęć muszą być:

* Minimalna ilość okienek w planie zajęć grup
* Określona oraz minimalna ilość okienek w planie zajęć nauczycieli
* Minimalna ilość lekcji tego samego przedmiotu w dniu tygodnia
* Równomierny rozkład zajęć różnego typu w dniu tygodnia
* Równomierny rozkład okienek w planach zajęć nauczycieli (minimalna ilość nauczycieli posiadających okienko w tym samym slocie czasowym)
* Minimalna ilość lekcji o wysokim priorytecie w ostatnich godzinach dni.

Głównym celem jest zoptymalizowanie planu pod względem:

* minimalna ilości wolnych slotów czasowych pomiędzy zajęciami na planie nauczycieli i grup,
* równomierne rozłożenie lekcji różnego typu w dniu,
* minimalna ilość powtórzeń tych samych lekcji w jednym dniu,
* jak najlepszemu wykorzystaniu sal z najmniejszą ilością miejsc.
  1. Wymagania funkcjonalne

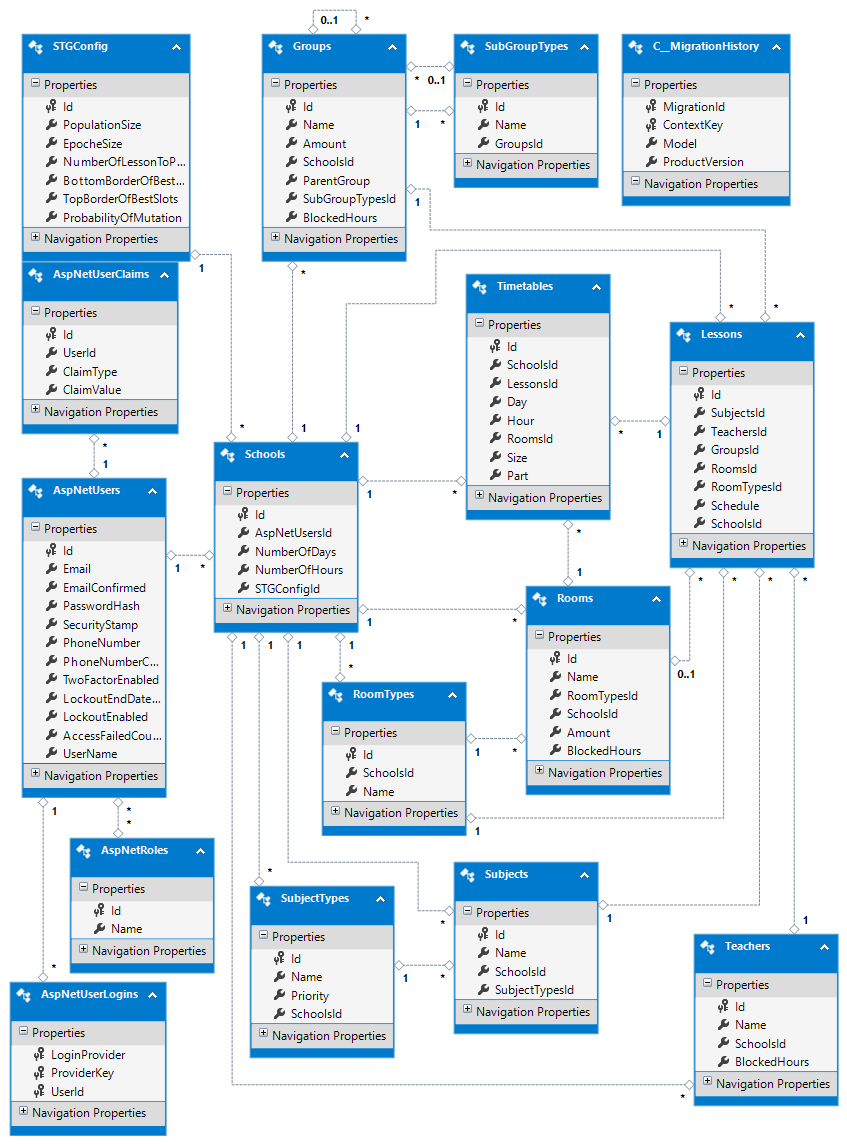


* 1. Wymagania niefunkcjonalne

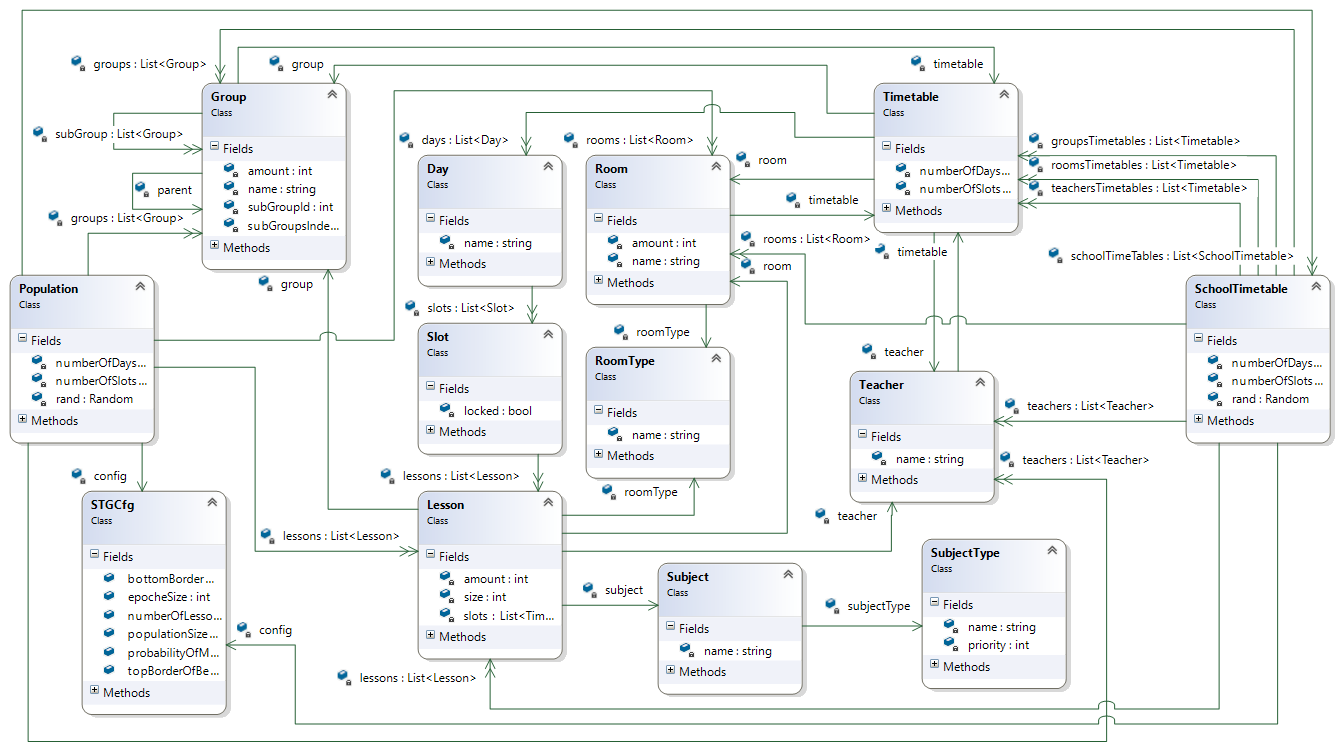
1. Projekt i implementacja

Analiza możliwych sposobów implementacji wykazała, że najlepszym rozwiązaniem takiego problemu, będzie zaprojektowanie aplikacji jako strona internetowa. Do implementacji został wykorzystany język C# w oparciu o Framework ASP.NET MVC. Takie rozwiązanie pozwala połączyć zalety jakimi są wieloplatformowość stron internetowych oraz obiektowość silnika algorytmu.

* 1. Architektura aplikacji
     1. Baza danych
        1. Diagram bazy danych



* + - 1. Opis tabel bazy danych
* AspNetUser – informacje dotyczące konta użytkownika,
* Groups – informacje dotyczące grupy, ilość uczniów, kategoria podziału na podgrupy, niedostępne godziny w tygodniu,
* Lessons – informacje dotyczące lekcji, nauczyciel prowadzący, grupa, rodzaj sali lekcyjnej, przedmiot szkolny, rozkład zajęć,
* Rooms – informacje dotyczące sali lekcyjnej, ilość miejsc, rodzaj sali, niedostępne godziny w tygodniu,
* RoomTypes – informacje dotyczące rodzaju sal lekcyjnych,
* Schools – informacje dotyczące szkoły oraz rozmiaru rozkładu godzin w tygodniu, przypisanie szkoły do konta użytkownika,
* STGConfig – informacje dotyczące konfiguracji algorytmu generującego plan zajęć,
* SubGroupTypes – informacje dotyczące kategorii podgrupy (np.: podział klasy na dwie grupy z wychowania fizycznego),
* Subjects – informacje dotyczące przedmiotu szkolnego, rodzaju przedmiotu szkolnego,
* SubjectTypes – informacje dotyczące rodzaju przedmiotu szkolnego,
* Teachers – informacje dotyczące nauczyciela, niedostępne godziny w tygodniu,
* Timetables – informacje dotyczące planu zajęć, dnia, godziny, lekcji, sali lekcyjnej, długości lekcji,
  + 1. Algorytm
       1. Diagram klas



* + - 1. Opis klas algorytmu

Klasa **Day** przechowuje listę slotów czasowych danego dnia.

Klasa **Group** przechowuje informacje o klasie/grupie takie jak nazwa, ilość uczniów, identyfikator grupy bazowej (w przypadku gdy opisuje ona podgrupę), listę identyfikatorów podgrup (w przypadku gdy opisuje ona grupę bazową), identyfikator kategorii podgrupy oraz odnośnik do planu zajęć.

Opis najważniejszych funkcji:

* **getSubGroupFreeSlotToLesson** - odpowiada za wygenerowanie list slotów czasowych w których jest możliwość wstawienia przekazywanej lekcji w każdej z podgrup.

Klasa **Lesson** przechowuje informacje o lekcji takie jak identyfikator przedmiotu szkolnego, identyfikator grupy, identyfikator nauczyciela prowadzonego daną lekcje, typ sali lekcyjnej w której powinna odbyć się dana lekcja, identyfikator sali lekcyjnej w której będzie odbywała się dana lekcja, listę slotów czasowych w których znajduję się dana lekcja, rozmiar danej lekcji, ilość takich samych lekcji w planie zajęć.

Opis najważniejszych funkcji:

* **isDifferent** – zwraca informacje czy dana lekcja jest tego samego przedmiotu szkolnego co lekcja przekazywana.

Klasa **Population** odpowiada za przeprowadzenie całego procesu generowania planu zajęć. Przechowuje listę osobników (planów zajęć dla szkoły), rozmiar planu zajęć (ilość dni i godzin w tygodniu), konfigurację dla procesu generowania planu zajęć oraz listy lekcji, nauczycieli, grup, sal lekcyjnych dla danej szkoły.

Opis najważniejszych funkcji:

* **generatePopulation** – generuje populacje początkową,
* **generateNewPopulation** – generowanie nowej populacji na podstawie obecnie istniejącej,
* **start** – rozpoczęcie procesu generowania planu zajęć.

Klasa **Room** przechowuje informacje o sali lekcyjnej takie jak nazwa, ilość miejsc, typ sali lekcyjnej oraz odnośnik do planu zajęć.

Klasa **RoomType** przechowuje informacje o typie sali lekcyjnej.

Klasa **SchoolTimetable** jest najbardziej rozbudowaną klasą ponieważ odpowiada za wygenerowanie, mutacje oraz krzyżowanie planu zajęć

Opis najważniejszych funkcji:

* **genereteTimetable** – generowanie planu zajęć,
* **findAndSetBestPositionToLessons**  – wstawianie lekcji w najlepszy slot,
* **getBestTimeSlot** – znalezienie najlepszego slotu do wstawienia danej lekcji,
* **findDifferentSubjectTheSameGroup** – znajduje lekcje z różnych przedmiotów tej samej grupy,
* **removeLessonsAndFindNewPosition** – zwalnia zajęty slot,
* **sortLessons** – sortuje lekcje pod względem podgrup, długości trwania lekcji, ilość lekcji w tygodniu, typu przedmiotu szkolnego,
* **fitness** – zwraca sumaryczny stopień przystosowania planów zajęć nauczycieli i grup,
* **crossover** – przeprowadza proces krzyżowania,
* **mutate** – przeprowadza proces mutacji,
* **isCorrect** – sprawdza czy wszystkie lekcje zostały wstawione do planu i nie ma żadnych powtórzeń.

Klasa **Slot** przechowuje informacje o slocie czasowym takie jak blokada slotu oraz lista lekcji występujących w danym slocie czasowym.

Klasa **STGCfg** przechowuje informacje o konfiguracji procesu generowania plany zajęć takie jak rozmiar populacji, ilość epok, ilość lekcji do jednoczesnego wstawienia, dolna i górna granica przedziału godzin do którego lekcje mają być wstawiane w pierwszej kolejności oraz prawdopodobieństwo do zajścia mutacji.

Klasa **Subject** przechowuje informacje o przedmiocie szkolnym takie jak nazwa oraz typ przedmiotu szkolnego.

Klasa **SubjectType** przechowuje informacje o typie przedmiotów szkolnych takie jak nazwa oraz priorytet.

Klasa **Teacher** przechowuje informacje o nauczycielu takie jak nazwa oraz odnośnik do planu zajęć.

Klasa **Timetable** przechowuje informacje o planie zajęć takie jak lista dni, odnośnik do grupy, nauczyciela bądź sali lekcyjnej do której dany plan należy.

Opis najważniejszych funkcji:

* **fitness** – zwraca sumaryczny stopień przystosowania planu zajęć,
* **fitnessFreeSlots** – zwraca stopień przystosowania pod względem ilości wolnych slotów pomiędzy lekcjami,
* **fitnessType** – zwraca stopień przystosowania pod względem rozkładu rodzajów zajęć w dniu,
* **fitnessSubject** – zwraca stopień przystosowania pod względem ilości wystąpień tej samej lekcji w jednym dniu,
* **getFreeSlotsToLesson -** odpowiada za wygenerowanie list slotów czasowych w których jest możliwość wstawienia przekazywanej lekcji.
  + 1. Interfejs użytkownika
  1. Opis najważniejszych funkcjonalności

1. Testy
2. Podsumowanie
3. Literatura

1. Defekt –więcej niż jednej lekcji w jednym slocie czasowym, braku lekcji w planie, lekcji w nieodpowiedniej sali lekcyjnej, … [↑](#footnote-ref-2)