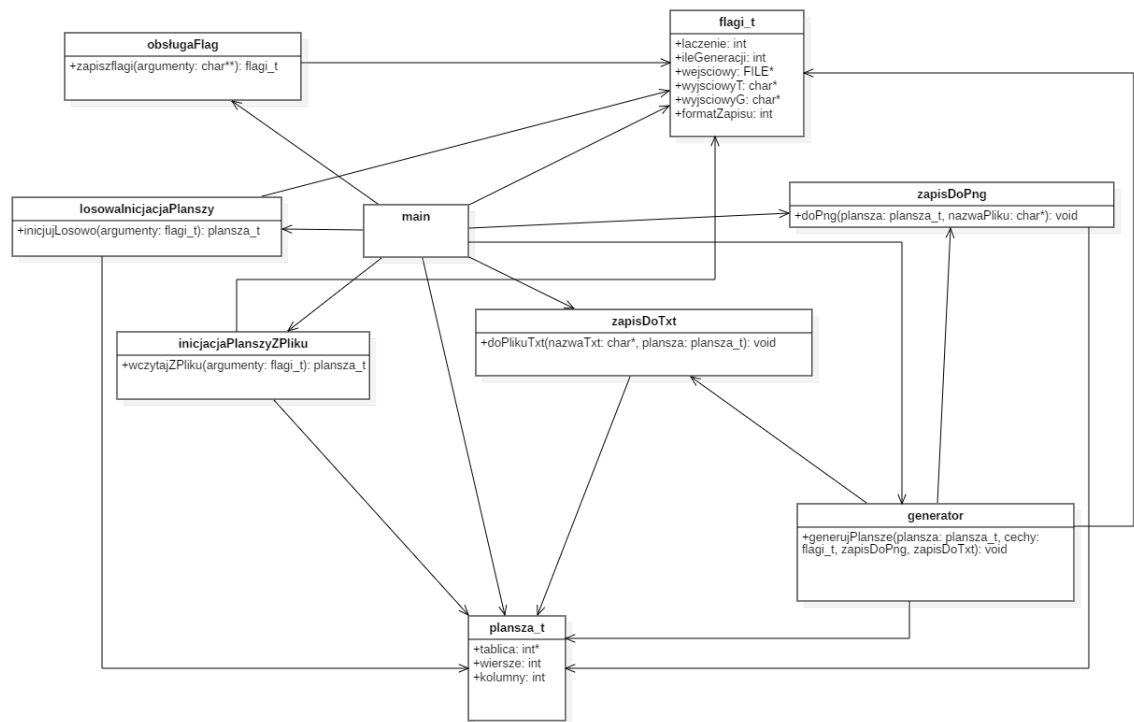


# Specyfikacja implementacyjna programu gra

Katarzyna Stankiewicz, Karolina Czachorska

24 marca 2019

## Moduły programu



## Opis modułów

### **zapisDoTxt**

Zapisuje planszę w formie tekstowej do pliku o nazwie wybranej przez użytkownika. Odpowiednio komórki martwe i żywe są reprezentowane przez 0 i 1.

### **zapisDoPng**

Zapisuje planszę w formie graficznej do pliku o nazwie wybranej przez użytkownika. Odpowiednio komórki martwe i żywe są reprezentowane przez czarne i białe kwadraty.

### **inicjacjaPlanszyZPliku**

W przypadku podania przez użytkownika pliku z danymi do odtworzenia planszy generuje planszę odpowiadającą tym danym.

### **losowaInicjacjaPlanszy**

W przypadku, gdy użytkownik nie poda pliku z danymi dotyczącymi początkowej planszy zostanie ona wygenerowana losowo lub z danymi wywołania jeśli zostały podane.

### **Generator**

Generuje nową planszę na podstawie aktualnych danych zgodnie z zasadami gry za pomocą algorytmu opisanego bardziej szczegółowo w kolejnej części.

### **obsługaFlag**

Zapisuje flagi podane przy wywołaniu programu wpisując je do struktury `flagi_t`.

### **Plansza\_t**

Zawiera definicję struktury z danymi dotyczącymi planszy.

- Tablica zawiera tablicę jednowymiarową wypełnioną 0 i 1, w zależności od stanu komórki
- Wiersze-wysokość planszy wyrażona liczbą komórek w pionie

- Kolumny- szerokość planszy wyrażona liczbą komórek w poziomie

## Flagi\_t

Zawiera definicję struktury z danymi dotyczącymi argumentów wywołania.

- laczenie - opisuje, czy wiersze i kolumny mają być łączone, 0-brak łączenia, 1-łączenie pionowe, 2-łączenie poziome, 3-łączenie całkowite
- ileGeneracji- określa liczbę generacji planszy które zostaną stworzone
- wejściowy- plik podany przez użytkownika z danymi dotyczącymi pierwszej planszy
- wyjściowyT- nazwa pliku podana przez użytkownika, do tego pliku zapisane mają być dane wygenerowanej planszy w formie tekstowej
- wyjściowyG- nazwa pliku podana przez użytkownika, do tego pliku zapisane mają być dane wygenerowanej planszy w formie graficznej
- formatZapisu- określa w jakim formacie ma zostać zapisany wynik, 0-tekst, 1-grafika, 2-grafika i tekst

## Opis przepływu sterowania

- 1.Wczytanie argumentów wywołania programu.
- 2.Wczytanie pliku (jeśli podano). Jeśli nie, wygenerowanie losowej planszy.
- 3.Uruchomienie symulacji.
  - 3.1.Wyświetlenie aktualnego stanu.
  - 3.2.Iteracja kolejnych generacji zgodnie z podaną liczbą lub domyślnie 10 razy.
    - 3.2.1.Stworzenie nowej struktury typu plansza, liczba wierszy i kolumn inicjowana jest taką samą wartością jak dla poprzedniej planszy.
    - 3.2.2.Obliczenie liczby żywych sąsiadów dla każdej komórki na podstawie poprzedniej planszy.
    - 3.2.3.Zapisywanie nowej wartości każdej komórki do nowej planszy, kierując się zasadami gry.
    - 3.2.4.Zapis do pliku tekstowego lub graficznego, w zależności od preferencji użytkownika.
- 4.Zakończenie symulacji.

## Opis głównych algorytmów

### Generowanie kolejnej planszy

Funkcja generator otrzymuje wskaźnik na planszę. Tworzona jest nowa plansza, mająca taką samą liczbę kolumn i wierszy co pierwotna. Następnie określany jest stan każdej komórki po kolei na podstawie liczby żywych sąsiadów. Wynik zapisywany jest do nowej planszy, pierwotna plansza przez cały czas pozostaje bez zmian aż do momentu określenia stanu wszystkich komórek. Następuje zapis do pliku graficznego lub tekstowego, w zależności od preferencji użytkownika. W kolejnym kroku nowa plansza staje się pierwotną, a cały algorytm powtarza się tyle razy ile zostało to wcześniej określone przez użytkownika.

## Testy

### Test modułu zapisDoTxt

Tworzymy przykładową strukturę i sprawdzamy poprawność zapisu do plików txt.

Struktura:

```
plansza.tablica[30]={1,0,0,0,1,1,1,0,0,1,1,0,1,1,0,0,1,1,0,1,1,1,0,1,1,0,1,1,0,1,0}
```

```
plansza.kolumny=6
```

```
plansza.wiersze=5
```

Spodziewany wynik w pliku tekstowym:

```
1 0 0 0 1 1
```

```
1 0 0 1 1 0
```

```
1 1 0 0 1 1
```

```
0 1 1 1 0 1
```

```
1 0 1 0 1 0
```

### Test modułu zapisdoPng

Tworzymy przykładową strukturę i sprawdzamy poprawność zapisu do plików png.

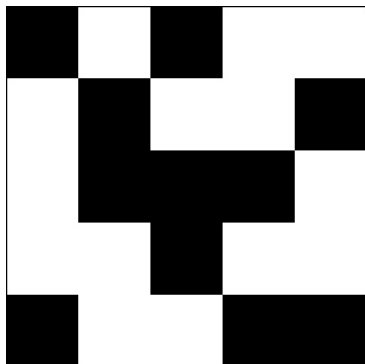
Struktura:

```
plansza.tablica[30]={0,1,0,1,1,1,0,1,1,0,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0}
```

```
plansza.kolumny=5
```

```
plansza.wiersze=5
```

Spodziewany wynik w pliku png:



### Test modułu inicjacji planszy ZPliku

Tworzymy przykładowe pliki. Kilka zawierających poprawne dane i kilka błędnych. Sprawdzamy poprawne działanie i prawidłowe reakcje na błędy. Sprawdzamy zachowanie się funkcji dla różnych rozmiarów planszy.

Pliki prawidłowe:

Plik1:

```
10101010101
01011001111
01001110000
01000101010
```

Plik2:

```
111111
000000
101010
101010
```

Pliki nieprawidłowe:

Plik1:

```
1010000
10101
1010100
0101101010
```

Plik2:

0108010  
1018280  
1009003

Plik3:

7fbd8en  
7dheb8j  
E772bsj  
909e89

### **Test modułu losowaInicjacjaPlanszy**

Sprawdzamy poprawność działania w przypadku gdy użytkownik nie poda pliku wejściowego oraz poda rozmiar planszy oraz w przypadku gdy rozmiar przyjmie wartość domyślną.

### **Test modułu generator**

Sprawdzamy działanie dla przykładowej planszy.

Struktura:

plansza.tablica[30]={1, 0, 1, 1, 0, 1, 0, 1,0, 0, 1, 1, 0, 1, 1, 1}

plansza.kolumny=4

plansza.wiersze=4

Plansza pierwotna(jako plik tekstowy):

```
1 0 1 1
0 1 0 1
0 0 1 1
0 1 1 1
```

Plansza po 1 generacji:

```
0 1 1 1
0 1 0 0
0 1 0 0
0 1 0 1
```