

Przedstawiona dokumentacja opisuje algorytm służący symulacji rozkładu bogactwa z określonymi założeniami zdefiniowanymi przez użytkownika.

I. Opis działania algorytmu:

Przez rozpoczęciem symulacji użytkownik definiuje założenia podstawowe symulacji:

- liczbę osób biorących udział w symulacji,
- liczbę rund symulacji,
- wartość majątku początkowego, który dla każdej osoby jest identyczny,
- część dochodu uboższej osoby, która służy do wyznaczenia stawki wygranej przekazywanej dla wygranej osoby,
- stawkę podatku od wygranej, która jest pobierana w każdej transakcji.

Spośród osób biorących udział w symulacji losowo wybierane są dwie osoby, które rywalizują między sobą. Następnie jest losowany wygrany. Stawka wygranej to iloczyn majątku uboższej osoby biorącej udział w rozgrywce i odsetka dochodu jaki ma być przekazany wygranej osobie zdefiniowanego przez użytkownika. Stawka wygranej jest odejmowana od majątku przegranego oraz dodawana do majątku wygranego. Jeśli zdefiniowana jest stawka podatku różna od zera to stawka wygranej, jaką otrzymuje wygrany, jest obniżona o stawkę podatku, a wartość kwoty podatku jest akumulowana w polu opisującym sumę pobranych podatków w każdej transakcji. Symulacja trwa określoną przez użytkownika liczbę rund, a po jej zakończeniu wyświetlana jest informacja na temat wyniku końcowego symulacji z podstawowymi statystykami rozgrywki. Użytkownik może także wyświetlić przebieg symulacji runda po rundzie oraz zapisać wartość majątku każdej osoby w każdej rundzie w pliku w formacie csv.

II. Opis poszczególnych elementów wykorzystanych w algorytmie.

Algorytm składa się z klasy `Rozklad_bogactwa`, w której to definiowane są wszystkie pola oraz metody.

1. Zawartość modyfikatora dostępu private:

- int Osoba_A - indeks osoby biorącej w symulacji (różnej od Osoba_B) w danej rundzie,
- int Osoba_B - indeks osoby biorącej w symulacji (różnej od Osoba_A) w danej rundzie,
- int Wygrany - indeks wygranej osoby w danej rundzie,
- int Przegrany - indeks przegranej osoby w danej rundzie,
- int Liczba_osob – liczba osób biorących udział w symulacji,
- int Liczba_rund - liczba rund w symulacji,
- int Dochod_poczatkowy - dochód każdej osoby przed rozpoczęciem symulacji. Każda osoba ma jednakowy dochód początkowy,
- int Osoba_max - indeks osoby, która uzyskała maksymalny majątek w całej symulacji,
- int* Liczba_wygranych - tablica, w której znajduje się informacja o liczbie wygranych rund przez każdą osobę,
- int* Liczba_przegranych - tablica, w której znajduje się informacja o liczbie przegranych rund przez każdą osobę,
- double Podatek - pole określające stawkę podatku,
- double Suma_podatkow – pole zawierające informację o sumie zebranych podatków w trakcie trwania symulacji,
- double Maksimum - pole zawierające informację o maksymalnej wartości majątku uzyskanej w danej symulacji,
- double Odsetek_majatku - część dochodu uboższego gracza, która służy do wyznaczenia stawki wygranej przekazywanej dla wygranego gracza,
- double** Majatek – macierz zawierająca informację o majątku każdej osoby w każdej rundzie.

2. Zawartość modyfikatora dostępu public:

- Rozklad_bogactwa(int Liczbaosob, int Liczbarund, int Dochodpoczatkowy=10, double Odsetekmajatku=0.01, double Stawkapodatku=0) – konstruktor klasy Rozklad_bogactwa.
 - int Liczbaosob – liczba osób biorących udział w symulacji,
 - int Liczbarund – liczba rund w symulacji,

- `int Dochodpoczatkowy` - dochód każdej osoby przed rozpoczęciem symulacji. Każda osoba ma jednakowy dochód początkowy (wartość domyślna 10),
- `double Odsetekmajatku` - część dochodu uboższego gracza, która służy do wyznaczenia stawki wygranej przekazywanej dla wygranego gracza (wartość domyślna 0.01),
- `double Stawkapodatku` - pole określające stawkę podatku (wartość domyślna 0),

Poza powyżej opisanymi polami, definiowane są także:

- `int Osoba_max` – wartość domyślna -1
 - `int* Liczba_wygranych` - o długości `Liczba_osob` z wartościami domyślnymi 0,
 - `int* Liczba_przegranych` - o długości `Liczba_osob` z wartościami domyślnymi 0,
 - `double Suma_podatkow` – wartość domyślna 0
 - `double Maksimum` – wartość domyślna 0
 - `double** Majatek` – tablica o długości `Liczba_rund+1` zawierająca tablice o długości `Liczba_osob`
- `int Losowanie_osob()` – metoda losująca dwie osoby biorące udział w danej rundzie. Zwraca pola `Osoba_A`, `Osoba_B`,
- `int Wynik_gry()` – metoda losująca wygraną osobę spośród `Osoba_A` i `Osoba_B`. Zwraca pola `Wygrany`, `Przegrany` oraz uzupełnia wartości w tablicach `Liczba_wygranych` i `Liczba_przegranych`,
- `void Zmiana_majatku(int Numer_rundy)` – metoda przypisująca wartości do macierzy `Majatek` dla rundy określonej wartością `Numer_rundy`. Dla osoby `Wygrany/Przegrany` przypisywana jest wartość z poprzedniej rundy plus/minus kwota wygranej, a dla pozostałych osób jest to wartość majątku z poprzedniej rundy. Jeśli występuje podatek większy od zera to kwota dla osoby `Wygrany` jest pomniejszona o podatek, a wartość podatku jest dodawana do pola `Suma_podatkow`. Wartości pól `Przegrany` i `Wygrany` są przypisywane w metodzie `int Wynik_gry()` i dlatego metoda `void Zmiana_majatku()` powinna być uruchamiana bezpośrednio po metodzie `int Wynik_gry()` w celu zapisania majątku osób biorących udział w danej rundzie symulacji,

- `int Numer_rundy` – pole określające numer rundy dla której w macierzy Majatek mają zostać zapisane wyniki z ostatniego wywołania metody `Wynik_gry()`,
- `void Uruchom_symulacje(bool przebieg_symulacji = false, bool dla_wszystkich = false)` – metoda, która dla każdej rundy uruchamia metody `Losowanie_osob()`, `Wynik_gry()`, `Zmiana_majatku(int Numer_rundy)`.
 - `bool przebieg_symulacji` – jeśli wartość jest równa `true` to uruchamiana jest metoda `Drukuj_przebieg_ostatniej_rundy()` dla każdej rundy i wyświetlane są wyniki dla osób biorących udział w danej rundzie. Wartość domyślna `false` powoduje nieuruchomienie metody `Drukuj_ustawienia_symulacji()`,
 - `bool dla_wszystkich` – jeśli wartość jest równa `true` to uruchamiana jest metoda `Drukuj_przebieg_ostatniej_rundy()` dla każdej rundy i wyświetlane są wyniki dla wszystkich osób biorących udział w symulacji. Wartość domyślna `false` powoduje uruchomienie `Drukuj_ustawienia_symulacji()` dla osób biorących udział w danej rundzie, pod warunkiem, że `przebieg_symulacji = true`, w innym razie metoda `Drukuj_ustawienia_symulacji()` nie zostanie uruchomiona,
- `void Drukuj_przebieg_ostatniej_rundy(int numer_rundy=-1, int liczba_osob=2)` – metoda wyświetla informację na temat przebiegu ostatniej rundy. Wyświetlane są informacje kto wygrał, kto przegrał oraz majątki osób,
 - `int numer_rundy` – numer rundy, z której będą wyświetlone wyniki. Wartość domyślna równa -1 powoduje wyświetlenie wyników z ostatniego wywołania metody `Wynik_gry()` lub `Uruchom_symulacje()`. Wykorzystywanie tej metody dla innych rund niż ostatnia runda w symulacji jest zalecane tylko poprzez metodę `void Uruchom_symulacje()`. Wykorzystanie metody `void Drukuj_przebieg_ostatniej_rundy()` z określonym polem `numer_rundy` innym niż liczba rund w symulacji spowoduje wyświetlenie prawidłowego stanu majątku, ale informacje na temat osoby wygranej i przegranej będą pochodzić z ostatniej rundy, ponieważ nie jest prowadzona historyzacja wartości pól `Wygrany` i `Przegrany`,
 - `liczba_osob` – pole określające dla ilu osób ma zostać wyświetlony stan majątku w danej rundzie. Wartość domyślna równa 2 powoduje wyświetlenie stanu majątku tylko dla osób biorących udział w danej rundzie. Wartość inna niż 2 spowoduje wyświetlenie wyników od osoby z numerem 0 do osoby z numerem wskazanym przez użytkownika,

- void Drukuj_ustawienia_symulacji() – metoda wyświetlająca ustawienia symulacji, które zostały zdefiniowane w konstruktorze klasy Rozklad_bogactwa,
- void Drukuj_wyniki(int numer_osoby = -1) – metoda wyświetlająca dane dotyczące wartości majątku, liczbie wygranych i przegranych uzyskanych w ciągu trwania symulacji,
 - int numer_osoby – pole wskazujące dla której osoby mają zostać wyświetlone wyniki. Wartość domyślna równa -1 powoduje wyświetlenie wyników dla wszystkich osób biorących udział w symulacji,
- double Wartosc_minimum() – metoda wyświetlająca minimalną wartość majątku w ostatniej rundzie symulacji spośród wszystkich osób,
- double Wartosc_maksimum() - metoda wyświetlająca maksymalną wartość majątku w ostatniej rundzie symulacji spośród wszystkich osób,
- double Wartosc_sredniej() - metoda wyświetlająca średnią wartość majątku wszystkich osób w ostatniej rundzie symulacji,
- double Wartosc_wariancji() - metoda wyświetlająca wariancję wartości majątku wszystkich osób w ostatniej rundzie symulacji,
- double Wartosc_podatkow() – metoda wyświetlająca pole Suma_podatkow zawierające informacje o sumie podatków zebranych w ciągu trwania symulacji,
- int Wygrana_osoba() – metoda wyświetlająca indeks osoby, która wygrała symulację,
- int Numer_rundy_max() – metoda wyświetlająca numer rundy, w której uzyskano wartość majątku równą wynikowi metody Wartosc_maksimum(),
- void Zapisz_wyniki_csv(string nazwa_pliku = "wyniki_symulacji.csv") – metoda zapisująca wartości majątku każdej osoby w każdej rundzie do pliku csv,
 - string nazwa_pliku = "wyniki_symulacji.csv" – pole, które przypisuje nazwę pliku, w którym zapisane zostaną wyniki symulacji. Wartość domyślna równa "wyniki_symulacji.csv" sprawia, że metoda tworzy plik o takiej nazwie.
- int main() – funkcja, w której tworzona jest klasa Rozklad_bogactwa i wywoływane są poszczególne metody tej klasy.

III. Instrukcja użytkowania programu.

Opisane metody w tym rozdziale powinny być wywoływane w funkcji int main(). Symulację należy rozpocząć od utworzenia klasy Rozklad_bogactwa z odpowiednimi polami według

preferencji użytkownika. Następnie należy wywołać metodę `Uruchom_symulacje()`. Wydrukowanie wyników odbywa się poprzez wykorzystanie metody `Drukuj_wyniki()`. Zapisanie wyników do pliku csv wykonuje się dzięki wywołaniu metody `Zapisz_wyniki_csv()`. W celu wyświetlenia dodatkowych informacji można wykorzystać metody opisane w poprzednim rozdziale.