Marta Załuska

INFORMATYKA PRAKTYCZNA, III SEMESTR, I STOPIEŃ, STACJONARNE

PARADOKS URODZIN

Opis projektu

W klasycznym ujęciu, rozwiązanie problemu urodzin polega na znalezieniu najmniejszej liczby osób takiej, żeby prawdopodobieństwo znalezienia dwóch osób obchodzących urodziny tego samego dnia było równe co najmniej ½. Daty urodzin wybierane są z rozkładem jednostajnym, nie uwzględnia się również lat przestępnych – urodzin 29 lutego. Paradoks polega na tym, że ta liczba jest zaskakująco mała – potrzeba jedynie 23 osób, aby spełniony był ten warunek.

Przygotowany przeze mnie program rozszerza ten problem na wszystkie planety Układu Słonecznego – gdzie liczba dni w roku jest różna, a zatem najmniejsza liczba takich osób także się zmienia.

Problem ma oczywiście dokładne rozwiązanie opisane wzorem, jednak program ma je przetestować metodą eksperymentalną – generuje daty urodzin i sprawdza zgodność z warunkami.

Projekt zrealizowany jest w pełni w języku Java, a interakcja z użytkownikiem odbywa się przez konsolę.

Wejście programu

Przy starcie program prosi użytkownika o wpisanie nazwy planety, dla której ma zostać obliczony problem urodzin. Użytkownik podaje polską nazwę dowolnej planety w Układzie Słonecznym, przy czym wielkość liter nie ma znaczenia.

Wynik programu

Program dla wybranej planety oblicza najmniejszą liczbę osób spełniającą warunek problemu i wypisuje ją na ekran.

Tabela 1: Oczekiwane wyniki problemu dla każdej z planet

Planeta	Długość roku	Oczekiwany wynik
Merkury	88 dni	12
Wenus	225 dni	18
Ziemia	365 dni	23
Mars	687 dni	32
Jowisz	4 333 dni	78
Saturn	10 756 dni	123
Uran	30 708 dni	207
Neptun	60 223 dni	290

Zastosowane algorytmy

Opis algorytmu

Szukamy najmniejszej liczby osób takiej, że prawdopodobieństwo istnienia w tej grupie dwóch osób świętujących urodziny w ten sam dzień wynosi co najmniej 50%. Generujemy losowo grupę n osób, a raczej daty ich urodzin – liczbę w przedziale $[1; d]$, gdzie d to liczba dni w roku na wybranej planecie. Daty urodzin wybierane są z rozkładem jednostajnym, w przypadku Ziemi nie uwzględnia się również lat przestępnych – urodzin 29 lutego.

Dla każdej osoby sprawdzamy, czy istnieje inna osoba z tą samą datą urodzin. Jeśli istnieje ktoś taki, przerywamy test, bo nie potrzebujemy informacji o kolejnej dopasowanej parze. Zapisujemy wynik (para znaleziona lub nie), generujemy na nowo pulę n osób i przeprowadzamy kolejny test.

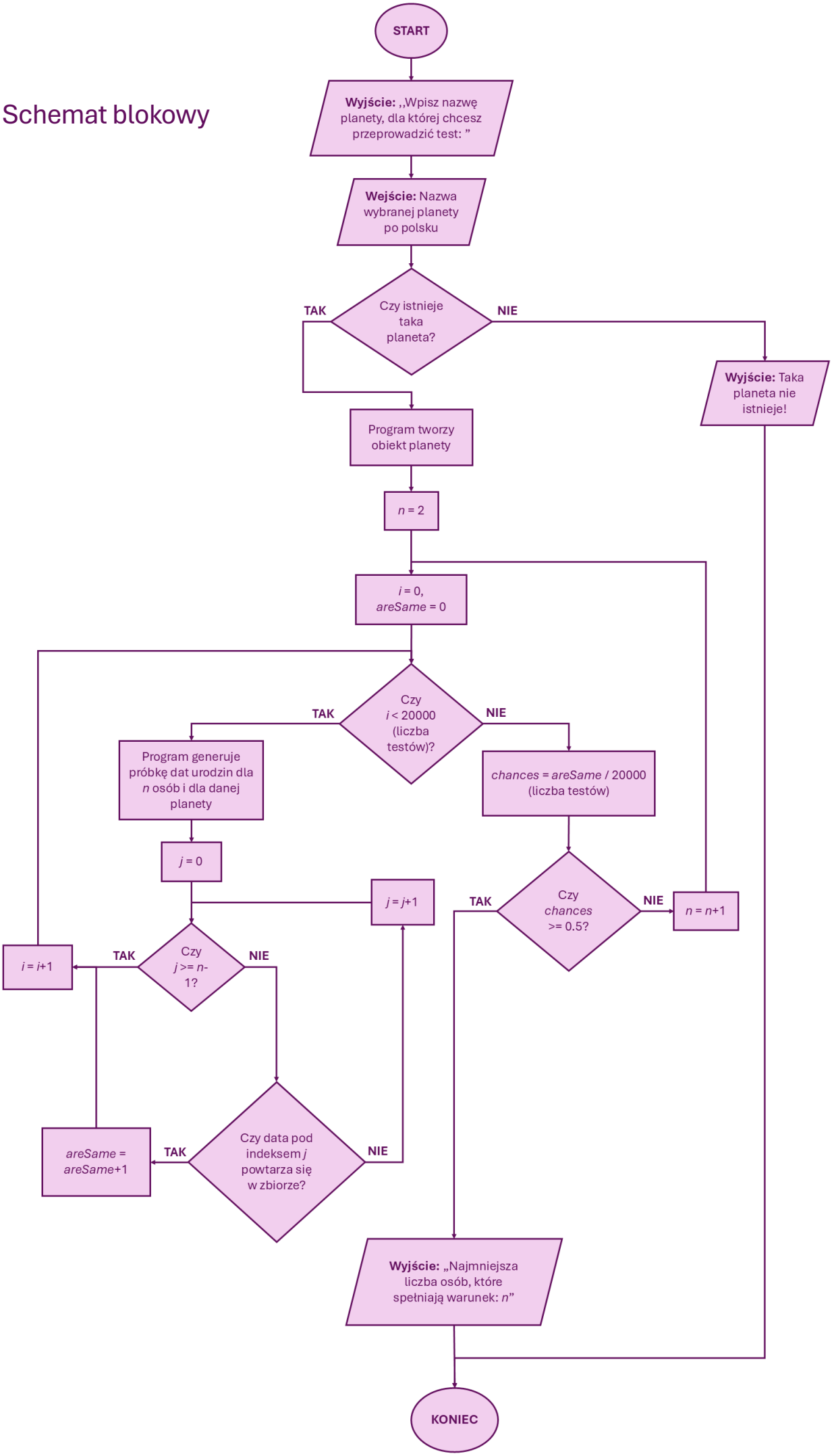
Po ustalonej liczbie prób (domyślnie 20000) zliczamy, ile z nich było pomyślnych (para została znaleziona), a ile nie. Na tej podstawie obliczamy szanse:

$$p_n = \frac{ilość\ prób\ pomyślnych}{ilość\ testów}$$

Jeśli $p_n \geq \frac{1}{2}$, to warunek problemu został spełniony, a n jest naszym ostatecznym wynikiem – szukaną najmniejszą liczbą osób. Jeśli $p_n < \frac{1}{2}$, to szukamy dalej – zwiększamy grupę o jedną osobę (zwiększamy n o 1) i powtarzamy operacje.

Program nie wpadnie nigdy w nieskończoną pętlę, ponieważ dla $d + 1$ osób zasada szufladkowa Dirichleta gwarantuje, że dwie z nich będą dzielić urodziny.

Schemat blokowy



Zastosowane klasy i metody

Main

Zajmuje się komunikacją z użytkownikiem poprzez konsolę oraz obsługą błędów. Tworzy obiekty klas SolarSystem i Planet na podstawie wejścia użytkownika. Tworzy obiekt klasy Birthdays i wywołuje na nim funkcję birthdayParadox. Wyświetla wynik na konsolę.

Klasa SolarSystem

Przechowuje tablicę z nazwami planet i tablicę z odpowiadającą każdej planecie długością roku (w dniach).

Klasa Planet

Dziedziczy po klasie SolarSystem. Docelowo przechowuje informację o planecie – jej nazwę, numer i liczbę dni w roku. Jej konstruktor weryfikuje otrzymany numer planety i, jeśli znajduje się ona w zbiorze, przypisuje jej wartości pobrane z klasy SolarSystem.

Klasa Birthdays

Ta klasa wykonuje potrzebne obliczenia i zwraca wynik problemu urodzin dla przekazanej konstruktorowi w argumencie planety.

Klasa Birthdays – metoda generateRandomBirthdays

Przyjmuje jako argument liczbę osób, dla których ma wygenerować daty urodzin. Daty wybierane są z rozkładem jednostajnym, losowanie przebiega przy użyciu funkcji Math.random(). Metoda zwraca tablicę z datami urodzin w formie numeru dnia w roku.

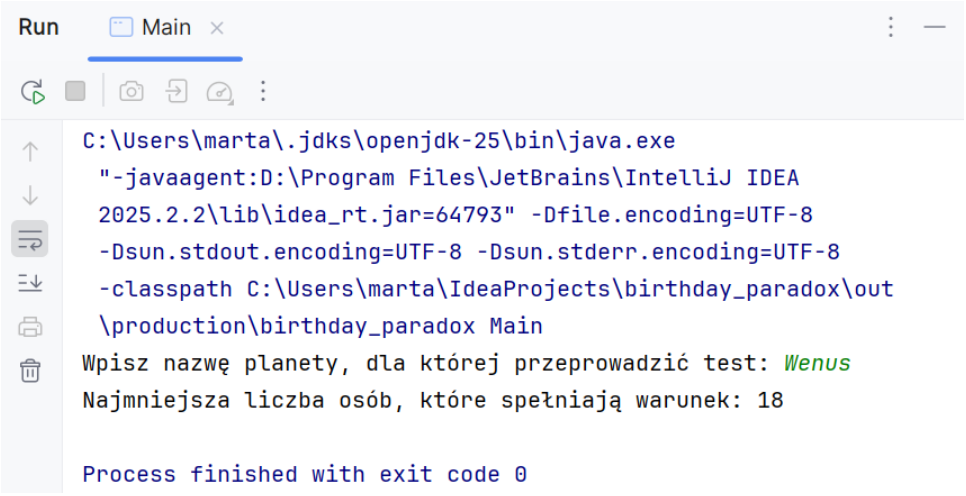
Klasa Birthdays – metoda sameBirthdayChances

Ta metoda oblicza szanse na spełnienie warunków problemu urodzin dla zadanej liczby osób. Przyjmuje domyślną liczbę testów – 20000. W każdym teście wywołuje metodę generateRandomBirthdays() i na otrzymanej tablicy wykonuje obliczenia. Przechodzi przez tablicę i sprawdza, czy dana data urodzin powtarza się. Jeśli tak – zwiększa licznik pozytywnych testów o 1 i przechodzi do kolejnego testu. Po przeprowadzeniu wszystkich testów zwraca prawdopodobieństwo w postaci licznika podzielonego przez liczbę testów.

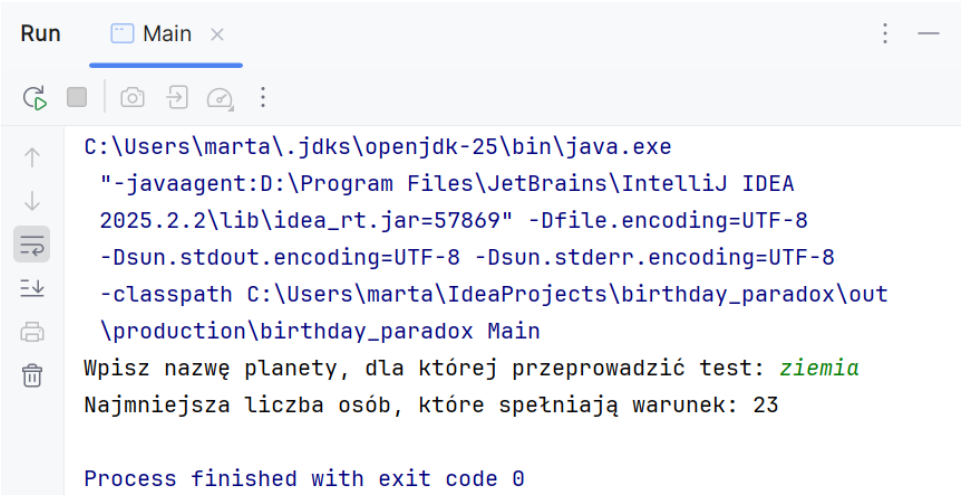
Klasa Birthdays – metoda birthdayParadox

Metoda przyjmuje początkową liczbę osób – 2. Następnie, wywołując metodę sameBirthdayChances() i przekazując jako argument liczbę osób, sprawdza prawdopodobieństwo. Jeśli prawdopodobieństwo wynosi co najmniej 50%, to znaczy, że ta liczba osób jest rozwiązaniem paradoksu urodzin. Wtedy program wypisuje wynik do konsoli i kończy działanie. Jeśli prawdopodobieństwo jest mniejsze, program zwiększa liczność grupy o jeden i ponownie wywołuje metodę.

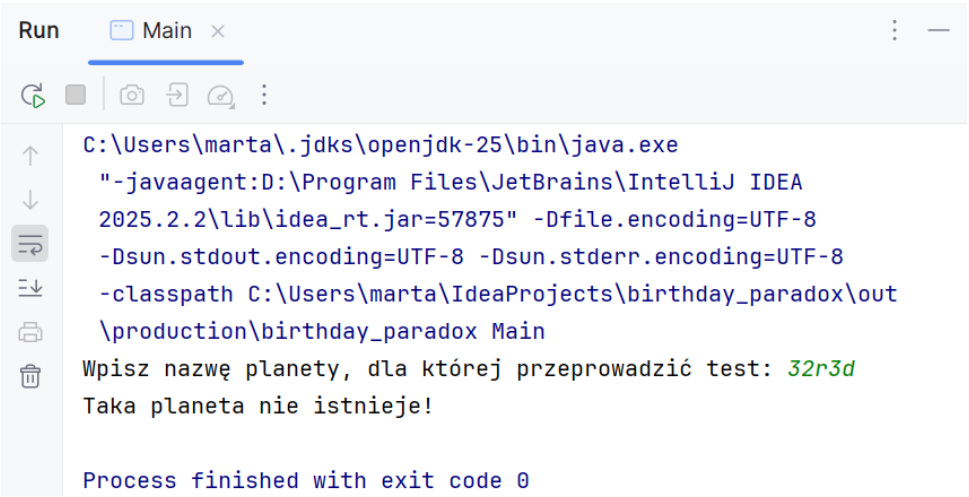
Testy poprawności działania



Zrzut ekranu 1: Pomyślny test - nazwa z wielkiej litery



Zrzut ekranu 2: Pomyślny test - nazwa z małej litery



Zrzut ekranu 3: Błąd - podana planeta nie istnieje

Wnioski

Przy liczbie testów 20000, program prawie zawsze zwraca wynik zgodny z oczekiwanym na planetach o krótszym roku. Jednak wraz ze zwiększaniem dni w roku, zmniejsza się dokładność programu dla tej liczby testów. Możliwością na rozwój programu byłoby dostosowanie liczby testów do liczby dni w roku, chociaż w takim przypadku zwiększenie dokładności odbywałoby się kosztem czasu działania programu.

Dodatkowo, aby komunikacja z użytkownikiem mogłaby być usprawniona i bardziej odporna na błędy, można byłoby przerobić interfejs z konsolowego na graficzny.

W fazie obecnej program jest w pełni odporny na błędy i łatwo skalowalny do innych parametrów, np. innego prawdopodobieństwa powodzenia testów.