Sito Eratostenesa w języku Java

Źródło: congerdesign, domena publiczna.

Sito Eratostenesa jest algorytmem wyznaczającym liczby pierwsze z zadanego przedziału <2, n>. Liczby pierwsze znajdują zastosowanie między innymi w kryptografii, o czym przeczytasz m.in. w e-materiałach:

- Zastosowanie liczb pierwszych,
- Szyfr RSA.

Ten e-materiał poświęcony jest implementacji algorytmu sita Eratostenesa w języku Java.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- Sito Eratostenesa w języku C++,
- Sito Eratostenesa w języku Python.

Więcej zadań? Sięgnij do: Sito Eratostenesa – zadania maturalne.

> **Twoje cele**
>
> - Zaimplementujesz algorytm Eratostenesa w języku Java.
> - Przeanalizujesz, w jaki sposób można zmodyfikować działanie sita Eratostenesa.
> - Rozwiążesz kilka problemów wymagających użycia sita Eratostenesa.

Animacja

Polecenie 1

Napisz program wyszukujący w zadanym przedziale <2, n> liczby pierwsze.

Swoje rozwiązanie przetestuj dla *n* wynoszącego 99.

Specyfikacja problemu:

Dane:

- n – liczba naturalna dodatnia; górna granica przedziału

Wynik:

Program na standardowym wyjściu wypisuje kolejne liczby pierwsze z zakresu <2, n>.

```
public class Main {
    public static void main(String[] args){
      // Tutaj dodaj własny kod. Użyj funkcji
      // System.out.print();
     }
}
```

```
1
```

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w animacji.

Film dostępny pod adresem /preview/resource/R1DNmXme7ZylQ

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału: Sito Eratostenesa

Polecenie 3

Zastanów się, jak jeszcze można modyfikować założenia omawianego algorytmu.

Przeczytaj

Przykład implementacji sita Eratostenesa w języku Java

1

Implementację sita Eratostenesa rozpoczniemy od inicjacji tablicy, która będzie przechowywać informację, czy dana liczba została skreślona. Przydatna będzie również zmienna n, która będzie nas informować o górnej granicy zakresu liczb [2, n].

2

Do tablicy dodajemy też elementy odpowiadające liczbom 0 oraz 1. Chcemy bowiem, aby indeks komórki odpowiadał liczbie, którą sprawdzamy. Indeks 1 będzie zatem oznaczał liczbę 1, indeks 2 liczbę 2 itd. Dlatego też do długości tablicy musimy dodać 1.

3

Zostaje domyślnie zainicjowana wartościami `false`. W naszym przypadku chcemy, aby na początku żadna z liczb nie była wykreślona, dlatego też musimy nadać wszystkim elementom tablicy `tablica` wartość `true`. Wykorzystamy do tego pętlę `for`.

Ponieważ elementy z indeksami 0 oraz 1 wstawiamy do tablicy tylko po to, aby indeksy pokrywały się z liczbami, możemy pozostawić je z wartością `false`. Nie będziemy ich brać pod uwagę w naszym programie.

4

Mając już przygotowane wszystkie potrzebne elementy, możemy przejść do wykreślania liczb. Musimy przeiterować tylko przez te elementy, których indeks jest nie większy niż pierwiastek z górnej granicy zakresu liczb – zmiennej `n`.

Aby otrzymać pierwiastek z liczby, używamy funkcji `Math.sqrt()`, gdzie jako argument podajemy długość naszej tablicy. Ponownie pomijamy pierwsze dwie liczby (0 oraz 1), ponieważ są one poza zakresem [2, n].

5

Następnie sprawdzamy, czy liczba, którą badamy, została wcześniej skreślona. Jeżeli tak, przechodzimy do kolejnej, jeżeli nie, usuwamy wszystkie jej wielokrotności.

6

Aby wykreślić wszystkie wielokrotności, potrzebujemy kolejnej pętli `for`.

7

W drugiej pętli `for` zmienną sterującą `j` inicjujemy kolejną wielokrotnością zmiennej `i`. Dopóki zmienna `j` jest w zasięgu tablicy, po każdej iteracji dodajemy do niej wartość

zmiennej `i`, przesuwając się przy tym do kolejnej jej wielokrotności.

8

Aby wykreślić liczbę, zmieniamy wartość komórki w tablicy z indeksem jej odpowiadającym na wartość `false`.

9

Po przejściu przez obie pętle, wszystkie indeksy komórek, w których zapisana jest wartość `true`, są liczbami pierwszymi. Wypiszmy je na ekran.

10

Poniżej program w całości:

Słownik

liczba pierwsza

liczba naturalna większa od 1, która dzieli się tylko przez jeden i przez samą siebie

sito Eratostenesa

algorytm, który przesiewa liczby z określonego zakresu w taki sposób, że zostają w nim tylko liczby pierwsze; służy on do znajdywania wszystkich liczb pierwszych w podanym przedziale <2, n>

wielokrotność liczby

> wielokrotność liczby a to taka liczba b, która powstaje przez pomnożenie liczby a przez liczbę naturalną n

Sprawdź się

Pokaż ćwiczenia: 

Ćwiczenie 1

Piotr Bajtocki prowadzi firmę budowlaną. Zgodnie z prawem ma obowiązek zaopatrzyć swoich pracowników w wodę. Butelki wody o pojemności 1,5 litra są pakowane w zgrzewkach. W każdej z nich zapakowano pewną liczbę pierwszą butelek. Najmniejsza zgrzewka opakowuje `m` butelek; największa `n`. Litr wody kosztuje `x` złotych. Budżet Piotra wynosi `y` złotych. Jak dużą zgrzewkę wody może kupić?

Jeśli liczba butelek, które w zgrzewce może kupić pan Bajtocki, jest większa od `n`, oznacza to, że jego budżet pozwala na zakup największej zgrzewki.

W swoim rozwiązaniu nie wykorzystuj metody `Math.sqrt`.

Swoje rozwiązanie przetestuj dla `x` wynoszącego 0,80; `y` równego 70; `n` równego 97.

Specyfikacja problemu:

Dane:

- `x` – cena litra wody; liczba zmiennoprzecinkowa dodatnia
- `y` – budżet Piotra; liczba zmiennoprzecinkowa dodatnia
- `n` – maksymalna liczba butelek w zgrzewce; liczba pierwsza; $n > 11$
- `m` – liczba butelek w najmniejszej zgrzewce; liczba pierwsza; $m \geq 5$

Wynik:

Na standardowym wyjściu program wypisuje największą liczbę butelek w zgrzewce, która mieści się w budżecie Piotra.

Twoje zadania

1. Program wypisuje największą liczbę butelek w zgrzewce, która mieści się w budżecie Piotra.

```java
public class Main {
    public static void main (String [] args) {
        int n = 100;
        int m = 5;
        double x = 0.8;
        double y = 70.00;
        // Tutaj dodaj kod.
        // Do wypisywania, użyj funkcji: System.out.println();

    }
}
```

```
1
```

Ćwiczenie 2

Zygmunt Bajtek otrzymał od rodziców prezent – zegarek analogowy. Zafascynowany podarkiem postanowił policzyć, ile razy wskazówka minutowa wskaże liczbę pierwszą. Uzyskawszy tę wiedzę, policzył kąty ϕ między osią biegnącą od środka tarczy do minutowego 0, a wskazówką minutową wskazującą liczby pierwsze (przykład na rysunku). Napisz program, który w kolejnych liniach wypisze kolejne wartości ϕ zaczynając od trzeciej minuty.

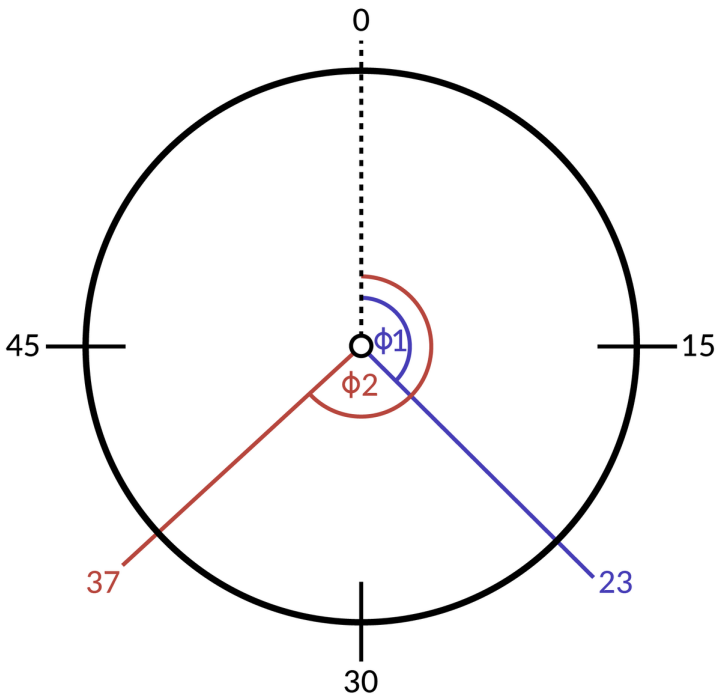
Specyfikacja problemu:

Dane:

- `sito` – sito Eratostenesa; 61-elementowa tablica wartości logicznych początkowo wypełniona wartościami `true`

Wynik:

Na standardowym wyjściu program wypisuje w kolejnych wierszach wartości kąta ϕ.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Twoje zadania

1. Program wypisuje kolejne wartości kąta fi.

```java
public class Main {
    public static void main (String [] args) {
        boolean[] sito = new boolean[61];
        for (int i = 0; i < sito.length; i++) {
            sito[i] = true;
        }

        // Tutaj dodaj kod. Program wypisuje kolejne wartości kąta fi.
        // Do wypisywania, użyj funkcji: System.out.println();

    }
}
```

```
1
```

Ćwiczenie 3

Hipoteza Goldbacha zakłada, że każda liczba naturalna parzysta większa od 2 jest sumą dwóch liczb pierwszych. Napisz program weryfikujący hipotezę Goldbacha dla liczb parzystych nie większych od n.

Swoje rozwiązanie zweryfikuj dla n wynoszącego 1000.

Specyfikacja problemu:

Dane:

- n – parzysta liczba naturalna dodatnia

Wynik:

Na standardowym wyjściu program wypisuje równania potwierdzające hipotezę Goldbacha, zgodnie z poniższym schematem; *NIE* w przeciwnym przypadku.

Przykład wyjścia:

Lewy składnik dodawania jest mniejszy od prawego; kolejne sumy wypisywane są rosnąco:

```
2 + 2 = 4
3 + 3 = 6
3 + 5 = 8
...
3 + 997 = 1000
```

Twoje zadania

1. Program wypisuje kolejne równania potwierdzające hipotezę Goldbacha; słowo NIE w przeciwnym przypadku.

```
public class Rozwiazanie{
    public static void main(String[] args) {
```

```
        int n = 1000;
        //Tutaj dopisz swój kod.
    }
}
```

```
```

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Sito Eratostenesa w języku Java

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

c) generowania liczb pierwszych metodą sita Eratostenesa,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Zaimplementujesz algorytm Eratostenesa w języku Java.
- Przeanalizujesz, w jaki sposób można zmodyfikować działanie sita Eratostenesa.
- Rozwiążesz kilka problemów wymagających użycia sita Eratostenesa.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał „Sito Eratostenesa w języku Java". Uczniowie zapoznają się z treściami w sekcji „Przeczytaj" w kontekście programowania.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć. Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali kryteria sukcesu.
2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Animacja". Uczniowie zapoznają się z poleceniem 1. W parach tworzą program. Chętne lub wybrane osoby prezentują swój kod. Nauczyciel go omawia.
2. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że będą rozwiązywać ćwiczenie nr 1 z sekcji „Sprawdź się". Uczniowie wykonują je w parach. Po ustalonym czasie następuje porównanie napisanych kodów podczas wspólnego omówienia rozwiązań.
3. Liga zadaniowa – uczniowie pracując w parach, wykonują ćwiczenie nr 2 z sekcji „Sprawdź się", a następnie dzielą się swoimi wynikami przez porównywanie napisanego kodu z inną grupą, która również zakończyła zadanie.
4. Nauczyciel dzieli uczniów na grupy czteroosobowe. Uczniowie wykonują ćwiczenie nr 3 z sekcji „Sprawdź się", a następnie każda grupa wyznacza jedną osobę, którą wymienia się z inną grupą. W nowych zespołach uczniowie porównują swój kod i wybierają najbardziej efektywne rozwiązanie.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie". W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy w zakresie programowania w języku Java.

Praca domowa:

1. Uczniowie wykonują polecenie 3 z sekcji „Animacja".

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).

Wskazówki metodyczne:

- Treści w sekcji „Animacja" można wykorzystać jako przykład zwiększenia efektywności algorytmu.