



Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Symulacja interaktywna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++

Źródło: Chris Barbalis, domena publiczna.

Rozwiązując niektóre problemy matematyczne i informatyczne, możemy stosować algorytmy numeryczne, wykorzystujące operacje na liczbach. Jedną z opcji jest metoda stycznych, która została omówiona w e-materiale [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych](#). Służy do wyszukiwania wartości miejsca zerowego funkcji z pewną zadaną dokładnością.

W tym e-materiale zaimplementujemy algorytm wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++.

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

- [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Java](#),
- [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Python](#).

Więcej zadań? Sięgnij do [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych – zadania maturalne](#).

Twoje cele

- Zastosujesz wzór na wyznaczanie kolejnych przybliżeń miejsca zerowego metodą stycznych.
- Przeanalizujesz implementację algorytmu wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++.
- Zaimplementujesz programy wykorzystujące metodę stycznych, by obliczyć przybliżoną wartość miejsca zerowego funkcji.

Przeczytaj

Metoda stycznych

Metoda stycznych to algorytm numeryczny wyznaczający przybliżoną wartość miejsca wero funkcji z pewną dokładnością. Innym algorytmem, który można wykorzystać do wyznaczenia przybliżonej wartości miejsca zerowego funkcji jest algorytm bisekcji, który omówiono w e-materiale [Algorytmy numeryczne i przybliżone](#).

1. Działanie metody rozpoczyna się od wskazania punktu startowego x_1 , będącego pierwszym przybliżeniem miejsca zerowego. Punkt ten wybieramy z przedziału $[a, b]$, w którym będzie znajdowało się dokładnie jedno [miejsce zerowe](#) funkcji.
2. Następnym krokiem jest wyznaczenie kolejnego przybliżenia x_2 , które jest współrzędną x punktu przecięcia stycznej do wykresu funkcji $f(x)$ w punkcie x_1 z osią OX . Przybliżenie to wyznaczamy według następującego wzoru:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

gdzie:

- x_1 – punkt startowy; przybliżona wartość miejsca zerowego funkcji $f(x)$ w przedziale $[a, b]$;
- x_2 – przybliżona wartość miejsca zerowego funkcji $f(x)$ w przedziale $[a, b]$; punkt przecięcia stycznej do wykresu funkcji $f(x)$ w punkcie x_1 z osią OX ;
- $f(x_1)$ – wartość funkcji $f(x)$ w punkcie x_1 ;
- $f'(x_1)$ – pochodna funkcji $f(x)$ w punkcie x_1 .

Styczna jest geometryczną interpretacją pochodnej (więcej na ten temat znajdziesz między innymi w e-materiałach [Interpretacja geometryczna pochodnej](#) oraz [Pochodna funkcji w punkcie](#)).

3. Jeżeli:

- różnica pomiędzy kolejnymi przybliżeniami x_1 i x_2 jest mniejsza lub równa wybranej dokładności ϵ lub
- wartość funkcji w punkcie x_2 jest bliska 0,

kończymy algorytm z wynikiem – przybliżoną współrzędną miejsca zerowego funkcji jest x_2 z dokładnością ϵ . W przeciwnym wypadku wartość x_2 staje się nowym punktem startowym (x_1) i wracamy do kroku drugiego.

Implementacja algorytmu w języku C++

Napiszmy program realizujący algorytm wyznaczania miejsca zerowego funkcji $f(x)$ w przedziale $[1, 3]$ z dokładnością do trzech miejsc po przecinku.

$$f(x) = x^3 - 5x$$

Na początek dołączamy niezbędne biblioteki. W trakcie pisania programu będziemy korzystali z funkcji `fabs` umożliwiającej obliczenie [wartości bezwzględnej](#) wyrażenia. Dołączamy bibliotekę `cmath`, dzięki której będziemy mogli użyć tej funkcji.

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
```

Tworzymy definicję funkcji o nazwie `funkcja`. Jej rolą jest obliczenie i zwrócenie wartości analizowanej funkcji w punkcie x zgodnie z wcześniej podanym wzorem.

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double funkcja(double x) {
6     return x * x * x - 5 * x;
7 }
```

W programie zdefiniujemy kolejną funkcję będącą pochodną funkcji, której miejsce zerowe chcemy wyznaczyć. Pochodna funkcji $f(x)$ ma następujący wzór:

$$f'(x) = 3x^2 - 5$$

W programie nadajemy jej nazwę `pochodna`.

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double funkcja(double x) {
```

```

6     return x * x * x - 5 * x;
7 }
8
9 double pochodna(double x) {
10     return 3 * x * x - 5;
11 }

```

Możemy przejść do zdefiniowania funkcji realizującej kolejne przybliżenia. Jej przyjmowanymi parametrami są: punkt startowy x_1 wartość błędu bezwzględnego ϵ_x oraz dokładność ϵ_y , czyli maksymalna różnica pomiędzy kolejnymi przybliżeniami, przy której możemy zakończyć algorytm.

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double funkcja(double x) {
6     return x * x * x - 5 * x;
7 }
8
9 double pochodna(double x) {
10     return 3 * x * x - 5;
11 }
12
13 double metodaStycznych(double x1, double ex, double ey) {
14 }

```

Tworzymy zmienną x_2 typu `double`. To w niej będą przechowywane kolejne przybliżenia. Na początek przypisujemy jej wartość x_1 .

W pętli `do...while` przypisujemy zmiennej x_1 wartość x_2 . W tym momencie wartość zapisana w zmiennej x_2 staje się nowym punktem startowym. Następnie zgodnie z przedstawionym wcześniej wzorem obliczamy miejsce zerowe i zapisujemy do zmiennej x_2 . Jeżeli różnica pomiędzy kolejnymi przybliżeniami (wartość bezwzględna, którą obliczamy funkcją `fabs()`) jest większa od dokładności przyjmowanej jako parametr, przechodzimy na początek pętli i ponawiamy obliczenia. W przeciwnym wypadku wychodzimy z pętli i na koniec zwracamy wartość zmiennej x_2 , czyli przybliżoną wartość miejsca zerowego funkcji. Wykonywanie pętli kończymy również w przypadku, gdy wartość funkcji dla podanego argumentu jest mniejsza lub równa dokładności, oznacza to również, że znaleźliśmy miejsce zerowe z zadowalającą dokładnością.

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double funkcja(double x) {
6     return x * x * x - 5 * x;
7 }
8
9 double pochodna(double x) {
10    return 3 * x * x - 5;
11 }
12
13 double metodaStycznych(double x1, double ex, double ey) {
14     double x2 = x1;
15
16     do {
17         x1 = x2;
18         x2 = x1 - funkcja(x1) / pochodna(x1);
19     } while (fabs(x2 - x1) > ex && funkcja(x2) > ey);
20
21     return x2;
22 }

```

Na koniec w głównej funkcji programu `main()` wywołujemy funkcję `metodaStycznych` z punktem startowym 3 i przybliżeniem oraz błędem bezwzględnym równymi 0,001. Wynik tego wywołania, pomnożony przez 1000, zapisujemy w zmiennej typu całkowitego `przyblizenie`. Dzięki temu wynik zostanie zaokrąglony do części całkowitej i w momencie, gdy do zmiennej `wynik` zapiszemy `przyblizenie` podzielone przez 1000, otrzymamy zaokrąglenie do części tysięcznej.

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double funkcja(double x) {
6     return x * x * x - 5 * x;
7 }
8
9 double pochodna(double x) {
10    return 3 * x * x - 5;

```



```

11 }
12
13 double metodaStycznych(double x1, double ex, double ey) {
14     double x2 = x1;
15
16     do {
17         x1 = x2;
18         x2 = x1 - funkcja(x1) / pochodna(x1);
19     } while (fabs(x2 - x1) > ex && funkcja(x2) > ey);
20
21     return x2;
22 }
23
24 int main() {
25     int przyblizenie = metodaStycznych(3, 0.0001, 0.0001) * 1000;
26     double wynik = przyblizenie / 1000.0;
27
28     cout << wynik << endl;
29
30     return 0;
31 }

```

Wynikiem przedstawionego programu i jednocześnie miejscem zerowym funkcji $f(x) = x^3 - 5x$ w przedziale $[1, 3]$ jest $x = 2,236$.

Pamiętaj, że **zbieżność** nie zawsze musi zachodzić. Jeżeli źle dobierzemy punkt startowy, algorytm stycznej nie zwróci wyniku. W takim wypadku program może np. nigdy się nie kończyć, ponieważ w każdej iteracji będzie się oddalał od znalezienia miejsca zerowego.

Słownik

cmath

biblioteka zawierająca deklaracje funkcji matematycznych, takich jak potęgowanie, pierwiastkowanie czy obliczanie wartości bezwzględnej

miejsce zerowe

argument funkcji, dla której przyjmuje ona wartość 0

wartość bezwzględna

odległość danej liczby od zera na osi liczbowej

zbieżność

własność mówiąca o tym, że dany problem posiada rozwiązanie, do którego przybliżamy się wraz ze wzrostem liczby wykonanych iteracji

Symulacja interaktywna

Polecenie 1

Przeanalizuj symulację interaktywną wizualizującą kolejne kroki działania algorytmu obliczającego przybliżone miejsce zerowe metodą stycznych. Na podstawie obserwacji przedstawionej symulacji dla różnych funkcji zastanów się, jak ważny jest dobór odpowiedniego punktu startowego. Na początku parametr jest dobrany niepoprawnie, co powoduje, że nie mamy pewności, czy algorytm kiedykolwiek się zakończy. Przypomnij sobie, jaki warunek powinien spełniać punkt startowy i dobierz go tak, aby znaleźć miejsce zerowe funkcji.

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

Polecenie 2

Przygotuj notatkę ze swoimi spostrzeżeniami dotyczącymi symulacji interaktywnej. Weź pod uwagę to, jak należy dobrać parametry oraz jakie warunki powinien spełniać punkt startowy.

Porównaj działanie algorytmu stycznych z metodą połowienia przedziałów.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Mamy daną funkcję $f(x)$ oraz jej pochodną $f'(x)$. Wiemy, że funkcja $f(x)$ ma jedno miejsce zerowe znajdujące się w przedziale (x_1, x_2) .

Napisz program, który za pomocą metody stycznych wyznaczy przybliżoną wartość miejsca zerowego funkcji $f(x)$ w danym przedziale i wypisze ją z dokładnością do trzeciego miejsca po przecinku (nie zaokrąglaj ostatniej cyfry).

Przetestuj swój program dla następujących danych:

- $f(x) = x \sin(x) + x^3 - 5$
- $f'(x) = x \cos(x) + \sin(x) + 3x^2$
- $x_1 = 1$
- $x_2 = 2$

Specyfikacja problemu:

Dane:

- funkcja – funkcja f , której przybliżoną wartość miejsca zerowego należy obliczyć
- pochodna – funkcja będąca pochodną funkcji f
- x_1 – początek przedziału; liczba rzeczywista
- x_2 – koniec przedziału; liczba rzeczywista

Wynik:

- wynik – przybliżona wartość miejsca zerowego funkcji; liczba rzeczywista

Twoje zadania

1. Program wyznacza przybliżoną wartość miejsca zerowego funkcji $f(x)$ w danym przedziale (x_1, x_2) i wypisuje ją na standardowe wyjście z dokładnością do trzech miejsc po przecinku, bez zaokrąglania.

Ćwiczenie 2



Napisz program, który za pomocą metody stycznych obliczy rozwinięcie dziesiętne liczby π z dokładnością do piątego miejsca po przecinku. Wykorzystaj funkcje `sin()` oraz `cos()` z biblioteki `cmath`.

Pamiętaj, że $\sin(\pi) = 0$.

Rozwiązanie wypisz z dokładnością do pięciu miejsc po przecinku, bez zaokrąglania.

Specyfikacja problemu:

Dane:

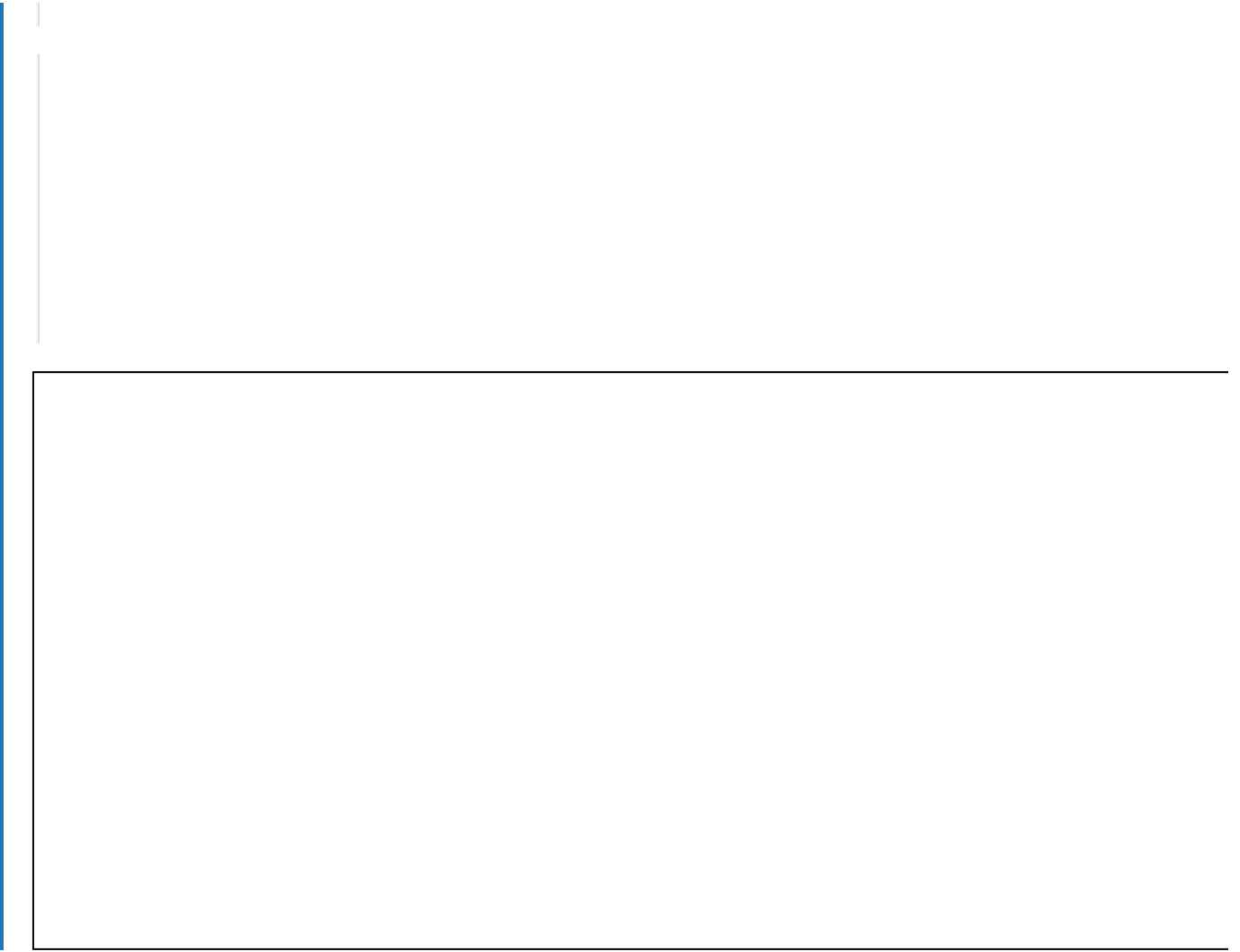
- π – liczba niewymierna

Wynik:

- `wynik` – rozwinięcie dziesiętne liczby π z dokładnością do pięciu miejsc po przecinku; liczba rzeczywista

Twoje zadania

1. Program wypisuje rozwinięcie dziesiętne liczby π .



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++

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

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

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

1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

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

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

1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;

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

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):

- Zastosujesz wzór na wyznaczanie kolejnych przybliżeń miejsca zerowego metodą stycznych.
- Przeanalizujesz implementację algorytmu wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++.
- Zaimplementujesz programy wykorzystujące metodę stycznych, by obliczyć przybliżoną wartość miejsca zerowego funkcji.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- mapa myśli;
- 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 C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć zawarte w sekcji „Wprowadzenie”. Następnie wspólnie z uczniami ustala kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu, np.
 - do czego służą algorytmy numeryczne?
 - jak działa metoda stycznych?
 - co zawiera biblioteka cmath?

Chętni lub wybrani uczniowie udzielają na nie odpowiedzi.

3. Nauczyciel dyskutuje z uczniami na temat ich odpowiedzi.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel sprawdza przygotowanie uczniów do lekcji, prosząc wybraną osobę o przedstawienie najważniejszych informacji dotyczących metody stycznych. Inna osoba zapisuje je na tablicy w formie mapy myśli. Jeśli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi ich o ponowne zapoznanie się z sekcją „Przeczytaj”.
2. Uczniowie powtarzają na swoich komputerach i testują program przedstawiony w sekcji „Przeczytaj”.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1 oraz 2 z sekcji „Sprawdź się”. Chętne lub wybrane osoby omawiają swój kod na forum klasy. Nauczyciel sugeruje ew. poprawki.
4. **Praca z multimediami.** Uczniowie zapoznają się z symulacją interaktywną. Przygotowują jej opis i proponują problem, który można za jej pomocą rozwiązać.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń z programowania w języku C++.

Praca domowa:

1. Uczniowie wykonują polecenia nr 1–2 z sekcji „Symulacja interaktywna”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Wskazówki metodyczne:

- Uczniowie mogą przygotować informacje na temat wykorzystywanych w e-materiale bibliotek.