

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«Брестский государственный университет имени А.С. Пушкина»
физико-математический факультет

ОСНОВЫ C++

Электронный практикум
для студентов специальностей

1-31 03 03-01 «Прикладная математика»,
1-31 03 06-01 «Экономическая кибернетика»,
1-31 04 08 «Компьютерная физика»



Брест
БрГУ имени А.С. Пушкина
2018



Кафедра
ПМИ

Начало

Содержание



Страница 1 из 42

Назад

На весь экран

Заккрыть

Авторы:

Белемук Ольга Вячеславовна – старший преподаватель кафедры прикладной математики и информатики Учреждения образования «Брестский государственный университет имени А.С. Пушкина»

Ткач Светлана Николаевна – старший преподаватель кафедры прикладной математики и информатики Учреждения образования «Брестский государственный университет имени А.С. Пушкина»

Рецензенты:

Сендер Александр Николаевич – заведующий кафедрой алгебры, геометрии и математического моделирования Учреждения образования «Брестский государственный университет имени А.С. Пушкина», кандидат физико-математических наук, доцент»

Петрукович Дмитрий Алексеевич – заведующий кафедрой экономики и управления Учреждения образования «Брестский государственный университет имени А.С. Пушкина», кандидат педагогических наук, доцент

Электронный практикум «Основы C++» предназначен для студентов, получающих высшее образование по специальности 1-31 03 03 «Прикладная математика», 1-31 03 06 «Экономическая кибернетика», 1-31 04 08 «Компьютерная физика» при изучении языка программирования C++ в рамках дисциплины «Программирование», а также для самостоятельного изучения основ языка программирования C++ и составлен в соответствии с действующей учебной программой по дисциплине «Программирование».



*Кафедра
ПМИ*

Начало

Содержание



Страница 2 из 42

Назад

На весь экран

Заккрыть

СОДЕРЖАНИЕ

Предисловие	5
ТЕМА 1 ОПЕРАТОРЫ ВЕТВЛЕНИЯ	6
Лабораторная работа 1. Программирование разветвляющихся алгоритмов	6
ТЕМА 2 ОПЕРАТОРЫ ЦИКЛА	11
Лабораторная работа 2. Арифметические циклы	11
Лабораторная работа 3. Циклы с контролем за монотонной величиной	17
Лабораторная работа 4. Итерационные циклы	22
ТЕМА 3 ФУНКЦИИ	37
Лабораторная работа 5. Описание и определение функций	37
Лабораторная работа 6. Перегрузка функций, шаблоны функций	38
ТЕМА 4 ТЕКСТОВЫЕ ФАЙЛЫ	39
Лабораторная работа 7. Обработка текстовых файлов	39
ТЕМА 5 СТРУКТУРЫ И ОБЪЕДИНЕНИЯ	40
Лабораторная работа 8. Структуры	40



*Кафедра
ПМИ*

Начало

Содержание



Страница 3 из 42

Назад

На весь экран

Заккрыть

ТЕМА 6 КЛАССЫ

41

Лабораторная работа 9-11. Классы. Конструкторы и деструкторы класса. Компоненты класса. Перегрузка операций 41

ТЕМА 7 НАСЛЕДОВАНИЕ

42

Лабораторная работа 12-13. Наследование классов, виртуальные функции, абстрактные классы 42

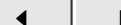
Тест для самоконтроля



*Кафедра
ПМИ*

Начало

Содержание



Страница 4 из 42

Назад

На весь экран

Закрыть

ПРЕДИСЛОВИЕ

Настоящий электронный практикум предназначен для студентов специальностей 1-31 03 03-01 «Прикладная математика», 1-31 03 06-01 «Экономическая кибернетика», 1-31 04 08 «Компьютерная физика» физико-математического факультета, изучающих язык программирования C++ в рамках дисциплины «Программирование». Цикл предложенных лабораторных работ соответствует действующим типовым учебным программам дисциплины «Программирование» для каждой из указанных специальностей.

В электронном издании помимо лабораторных работ излагается теоретический материал, необходимый для их выполнения, а также примеры реализации алгоритмов в виде программного кода на языке C++.

Электронный практикум направлен на облегчение самостоятельной работы студентов при изучении си-подобных языков программирования, в частности, особенностей синтаксиса и семантики языка C++, может использоваться при подготовке к лекциям, лабораторным занятиям и экзамену.

Содержит тест для самопроверки по теоретическому материалу.



*Кафедра
ПМИ*

Начало

Содержание



Страница 5 из 42

Назад

На весь экран

Заккрыть

ТЕМА 1

ОПЕРАТОРЫ ВЕТВЛЕНИЯ

. Лабораторная работа 1. Программирование разветвляющихся алгоритмов

Цель: получать практические умения реализации консольных приложений на языке C++ с использованием операторов ветвления **if** и **switch**.

Оборудование и ПО: ПК, среда разработки приложений для языков программирования C/C++ (DEV-Cpp или MS Visual Studio).

Содержание работы: написать программу, в которой реализовано решение поставленной задачи с использованием классов потокового ввода и вывода **iostream**.

Порядок выполнения работы:

1. Изучить синтаксис и семантику операторов **if** и **switch**.
2. Изучить теоретический материал о линейных операторах и операторах выбора в языке программирования C++, пример первоисточника: Подбельский В.В., *Язык C++: Учеб.пособие.* - 5-е изд. - М: Финансы и статистика, 2003. С. 90-96.
3. Познакомиться с **примером программного кода**.
4. Выполнить задание, соответствующее собственному варианту.



Кафедра
ПМИ

Начало

Содержание



Страница 6 из 42

Назад

На весь экран

Заккрыть

Задание лабораторной работы:

Вычислить значение выражения **s** (**x**, **y** ввести с клавиатуры, используя объекты **cout**, **cin** класса **iostream**). Предусмотреть выбор функции $f(x)$: $\sin(x)$, x^2 , e^x с помощью оператора **switch**.

Варианты заданий:

$$1. s = \begin{cases} (f(x) + y)^2 - \sqrt[3]{|f(x)|}, & xy > 0 \\ (f(x) + y)^2 + \sin(x), & xy < 0 \\ (f(x) + y)^2 + y^3, & xy = 0 \end{cases}$$

$$2. s = \begin{cases} \ln(f(x)) + \sqrt[3]{|f(x)|}, & \frac{x}{y} > 0 \\ \ln\left|\frac{f(x)}{y}\right| (x + y)^3, & \frac{x}{y} < 0 \\ (f(x)^2 + y)^3, & \text{в остальных случаях} \end{cases}$$

$$3. s = \begin{cases} f(x)^2 + \sqrt[3]{y} + \sin(y), & x - y = 0 \\ (f(x) - y)^2 + \ln(x), & x - y > 0 \\ (y - f(x))^2 + \operatorname{tg}(y), & x - y < 0 \end{cases}$$

$$4. s = \begin{cases} \sqrt[3]{|f(x) - y|} + \operatorname{tg}(f(x)), & x > y \\ (y - f(x))^3 + \cos(f(x)), & y > x \\ (y + f(x))^2 + x^3, & y = x \end{cases}$$



Кафедра
ПМИ

Начало

Содержание



Страница 7 из 42

Назад

На весь экран

Заккрыть

$$5. s = \begin{cases} y\sqrt{f(x)} + 3\sin(x), & x > y \\ x\sqrt{|f(x)|}, & x < y \\ \sqrt[3]{|f(x)|} + \frac{x^3}{y}, & \text{в остальных случаях} \end{cases}$$

$$6. s = \begin{cases} e^{f(x)-|y|}, & 0.5 < xy < 10 \\ \sqrt[3]{|f(x) + y|}, & 0.1 < xy < 0.5 \\ 2f(x)^2, & \text{в остальных случаях} \end{cases}$$

$$7. s = \begin{cases} e^{f(x)}, & 1 < xy < 10 \\ \sqrt[3]{|f(x) + 4y|}, & 12 < xy < 40 \\ yf(x)^2, & \text{в остальных случаях} \end{cases}$$

$$8. s = \begin{cases} \left(f(x)^2 + y\right)^3, & \frac{x}{y} < 0 \\ \ln\left|\frac{f(x)}{y}\right| + \frac{x}{y}, & \frac{x}{y} > 0 \\ \sqrt[3]{|\sin(y)|}, & \text{в остальных случаях} \end{cases}$$

$$9. s = \begin{cases} 2f(x)^3 + 3y^2, & x > |y| \\ |f(x) - y|, & 3 < x < |y| \\ \sqrt[3]{|f(x) - y|}, & \text{в остальных случаях} \end{cases}$$

$$10. s = \begin{cases} \ln(|f(x)| + |y|), & |xy| > 10 \\ e^{f(x)+y}, & |xy| < 10 \\ \sqrt[3]{|f(x)|} + y, & |xy| = 10 \end{cases}$$



Кафедра
ПМИ

Начало

Содержание



Страница 8 из 42

Назад

На весь экран

Заккрыть

Краткие теоретические сведения:

1. Оператор условной передачи управления **if**

Формат оператора выбора:

```
if (выражение) оператор 1;  
else оператор 2;
```

Выражение должно быть скалярным и может иметь арифметический тип или тип указателя. Если выражение не равно нулю, то выполняется оператор 1, иначе – оператор 2.

2. Оператор множественного выбора **switch**

Общая форма оператора следующая:

```
switch (переменная выбора) {  
  case const1: операторы 1; break;  
  ...  
  case constN: операторы N; break;  
  default: операторы N+1;  
}
```

При использовании оператора **switch** сначала анализируется переменная выбора и проверяется, совпадает ли её значение со значением



*Кафедра
ПМИ*

Начало

Содержание



Страница 9 из 42

Назад

На весь экран

Заккрыть

одной из констант. При совпадении выполняются операторы этого **case**. Конструкция **default** (может отсутствовать) выполняется, если результат выражения не совпал ни с одной из констант.

Пример: Выведите значение функции в точке x , выбрав при этом одну из функций $tg(x)$, e^x или x^2 .



Кафедра
ПМИ

Начало

Содержание



Страница 10 из 42

Назад

На весь экран

Заккрыть

ТЕМА 2

ОПЕРАТОРЫ ЦИКЛА

. Лабораторная работа 2. Арифметические циклы

Цель: получать практические умения реализации консольных приложений на языке C++ с использованием оператора цикла **for**.

Оборудование и ПО: ПК, среда разработки приложений для языков программирования C/C++ (DEV-Cpp или MS Visual Studio).

Содержание работы: написать программу, в которой реализовано решение поставленной задачи с использованием классов потокового ввода и вывода **iostream**.

Порядок выполнения работы:

1. Изучить синтаксис и семантику оператора **for**.
2. Изучить теоретический материал об операторах циклов в языке программирования C++, пример первоисточника:
Подбельский В.В., Язык C++: Учеб.пособие. - 5-е изд. - М: Финансы и статистика, 2003. С. 97-106.
3. Изучить материал об **арифметических циклах**.
4. Познакомиться с **примером программного кода**.
5. Выполнить **задание**, соответствующее собственному варианту.



Кафедра
ПМИ

Начало

Содержание



Страница 11 из 42

Назад

На весь экран

Заккрыть

Задание лабораторной работы:

Вычислить сумму **n** первых слагаемых ряда (**n**, **x** – параметры функций ввести с клавиатуры в функции **main()** или задать как константу). Вычисления реализовать в отдельных функциях, возвращающих значение суммы ряда, имеющих вид:

double A1Cycle(**int**,**double**) – для функции, в которой реализовано вычисление суммы посредством оператора **for** в общем виде;

double A2Cycle(**int**,**double**) – для функции, в которой реализовано вычисление суммы посредством оператора **for** без тела цикла (пустой оператор в теле цикла);

double A3Cycle(**int**,**double**) – для функции, в которой реализовано вычисление суммы посредством оператора **for**, в заголовке которого отсутствует выражение **_модификатор**;

double A4Cycle(**int**,**double**) – для функции, в которой реализовано вычисление суммы посредством оператора **for**, в заголовке которого отсутствует выражение **_условие** окончания цикла (в теле цикла реализовать проверку, выход из цикла осуществляется с помощью оператора **break**).

Сравнить полученные значения (вывести на консоль).



*Кафедра
ПМИ*

Начало

Содержание



Страница 12 из 42

Назад

На весь экран

Заккрыть

№ вар	Задание
1	$S = \frac{x-1}{x+1} + \frac{1}{3} \cdot \left(\frac{x-1}{x+1}\right)^3 + \dots + \frac{1}{2n+1} \left(\frac{x-1}{x+1}\right)^{2n+1} + \dots$ х изменяется в диапазоне [0.2;1]
2	$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$ х изменяется в диапазоне [0.1;1]
3	$S = \frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2-1} + \dots$ х изменяется в диапазоне [0,1;1]
4	$S = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$ х изменяется в диапазоне [0.1;1]
5	$S = 1 - \frac{3}{2!}x^2 + \frac{9}{4!}x^4 - \dots + (-1)^n \frac{2n^2+1}{(2n)!}x^{2n} + \dots$ х изменяется в диапазоне [0.1;1]
6	$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$ х изменяется в диапазоне [0.1;1]
7	$S = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \frac{(2x)^6}{720} + \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!} + \dots$ х изменяется в диапазоне [0.1;1]
8	$S = x + \frac{x^3}{3!} + \frac{x^5}{4!} + \frac{x^7}{7!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$ х изменяется в диапазоне [0.1;1]
9	$S = 1 + 3x^2 + \frac{5x^4}{2!} + \frac{7x^6}{3!} + \dots + \frac{(2n+1)x^{2n}}{n!} + \dots$ х изменяется в диапазоне [0.1;1]
10	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln 3^2}{2!}x^2 + \frac{\ln 3^3}{3!} + \dots + \frac{\ln 3^n}{n!}x^n + \dots$ х изменяется в диапазоне [0.1;1]



Кафедра
ПМИ

Начало

Содержание



Страница 13 из 42

Назад

На весь экран

Заккрыть

Краткие теоретические сведения:

1. Арифметические циклы

Циклы с заранее известным количеством повторений называются арифметическими, они используются при вычислении конечных сумм (произведений).

С понятием арифметического цикла тесно связано понятие рекуррентного соотношения. Рекуррентное соотношение – это формула, устанавливающая связь между следующим и предыдущим значением последовательности (ряда).

Пример 1. Дана последовательность

$$x^2, \frac{x^4}{2!}, \frac{x^6}{4!}, \dots$$

Запишем формулу общего члена последовательности $a_i = \frac{x^{2i}}{(2i-2)!}$.

$$\text{Тогда } a_n = \frac{x^{2n}}{(2n-2)!} \quad \& \quad a_{n+1} = \frac{x^{2(n+1)}}{(2(n+1)-2)!}.$$

Разделим a_{n+1} на a_n .

$$\begin{aligned} \frac{a_{n+1}}{a_n} &= \frac{x^{2n+2}}{(2n+2-2)!} \cdot \frac{(2n-2)!}{x^{2n}} = \frac{x^{2n} \cdot x^2 \cdot (2n-2)!}{(2n-2)! \cdot (2n-1) \cdot 2n \cdot x^{2n}} = \\ &= \frac{x^2}{(2n-1) \cdot 2n} \end{aligned}$$



Кафедра
ПМИ

Начало

Содержание



Страница 14 из 42

Назад

На весь экран

Заккрыть

В результате получим рекуррентную формулу

$$a_{n+1} = a_n \cdot \frac{x^2}{(2n - 1) \cdot 2n}.$$

2. Оператор циклов **for**

В программировании на языке C++ арифметические циклы реализуются с помощью оператора **for**, основная форма цикла **for** имеет следующий вид:

for (инициализация ; выражение_условие ; выражения_модификации)
тело_цикла;

Инициализация - последовательность определений и выражений, разделенных запятыми. Все они выполняются только 1 раз при входе в цикл. В простейшей форме инициализация используется для присвоения начального значения параметру цикла.

Выражение_условие - обычно выражение, которое определяет, когда цикл должен быть завершен: завершение цикла происходит при нарушении истинности этого выражения (нулевое значение).

Выражения_модификации – состоит из выражений, разделенных запятыми. Эти выражения вычисляются на каждой итерации после выполнения тела цикла и до следующей проверки выражения условия.



Кафедра
ПМИ

Начало

Содержание



Страница 15 из 42

Назад

На весь экран

Заккрыть

Эти три раздела **заголовка цикла** должны быть разделены точкой с запятой.

Выполнение цикла происходит до тех пор, пока условное выражение истинно. Как только условие становится ложным, начинает выполняться следующий за циклом **for** оператор.

Пример 2

Написать фрагмент программы вычисления суммы квадратов k натуральных чисел, используя оператор **for** а) в общем виде; б) тело цикла – пустой оператор; в) отсутствует выражение _модификатор.

а) **for** (int j=1, s=0; j<=k; j++) s+=j*j;

б) **for** (int j=1,s=0; j<=k; s+=j*j++);

в) **for** (int j=0,s=0; j<=k;)s+=++j*j;



*Кафедра
ПМИ*

Начало

Содержание



Страница 16 из 42

Назад

На весь экран

Закрыть

Лабораторная работа 3. Циклы с контролем за монотонной величиной

Цель: получать практические умения реализации консольных приложений на языке C++ с использованием операторов цикла **while**, **do...while**.

Оборудование и ПО: ПК, среда разработки приложений для языков программирования C/C++ (DEV-Cpp или MS Visual Studio).

Содержание работы: написать программу, в которой реализовано решение поставленной задачи с использованием классов потокового ввода и вывода **iostream**.

Порядок выполнения работы:

1. Изучить синтаксис и семантику операторов **while** и **do...while**.
2. Изучить теоретический материал об операторах циклов в языке программирования C++, пример первоисточника:

Подбельский В.В., Язык C++: Учеб.пособие. - 5-е изд. - М: Финансы и статистика, 2003. С. 97-106.

3. Изучить материал о **циклах с контролем за монотонной величиной**.
4. Познакомиться с **примером программного кода**.
4. Выполнить **задание**, соответствующее собственному варианту.



Кафедра
ПМИ

Начало

Содержание



Страница 17 из 42

Назад

На весь экран

Заккрыть

Задание лабораторной работы:

Вычислить сумму ряда с точностью **eps** (**eps**, **x** ввести с клавиатуры или задать как константу). Вычисления реализовать в отдельных функциях, возвращающих значение суммы ряда, имеющих вид:

double CMV1Cycle(**double, double, double, int**) – для функции, в которой контроль ведется по модулю разности точного и приближенного значения ряда (функции) (1-ый параметр в функции CMV1Cycle – значение **x**, 2-ой – точное значение функции **y**, 3-ий – значение **eps**, 4-ый – количество итераций);

double CMV2Cycle(**double, double, int**) – для функции, в которой контроль ведется по модулю разности следующего и предыдущего члена последовательности (ряда) (1-ый параметр в функции CMV2Cycle – значение **x**, 2-ой – значение **eps**, 3-ий – количество итераций);

double CMV3Cycle(**double, double, int**) – для функции, в которой контроль ведется по модулю «отбрасываемого» слагаемого (1-ый параметр в функции CMV3Cycle – значение **x**, 2-ой – значение **eps**, 3-ий – количество итераций);

Сравнить полученные значения суммы ряда (вывести на консоль количество итераций каждого цикла, полученную сумму ряда и точное значение функции).



Кафедра
ПМИ

Начало

Содержание



Страница 18 из 42

Назад

На весь экран

Заккрыть

№ вар.	Задание
1	$S = 1 + 3x^2 + \frac{5x^4}{2!} + \frac{7x^6}{3!} + \dots + \frac{(2n+1)x^{2n}}{n!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = (1 + 2x^2)e^{x^2}$</p>
2	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln 3^2}{2!}x^2 + \frac{\ln 3^3}{3!} + \dots + \frac{\ln 3^n}{n!}x^n + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = 3^x$</p>
3	$S = \frac{x-1}{x+1} + \frac{1}{3} \cdot \left(\frac{x-1}{x+1}\right)^3 + \dots + \frac{1}{2n+1} \left(\frac{x-1}{x+1}\right)^{2n+1} + \dots$ <p>х изменяется в диапазоне $[0.2;1]$, $y = \frac{1}{2} \ln x$</p>
4	$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = \sin x$</p>
5	$S = \frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2-1} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = \frac{1+x^2}{2} \arctg x - \frac{x}{2}$</p>
6	$S = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = \frac{e^x + e^{-x}}{2}$</p>
7	$S = 1 - \frac{3}{2!}x^2 + \frac{9}{4!}x^4 - \dots + (-1)^n \frac{2n^2+1}{(2n)!}x^{2n} + \dots$ <p>х – в диапазоне $[0.1;1]$, $y = \left(1 - \frac{x^2}{2}\right) \cos x - \frac{x}{2} \sin x$</p>
8	$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = \cos x$</p>
9	$S = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \frac{(2x)^6}{720} + \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = 2(\cos^2 x - 1)$</p>
10	$S = x + \frac{x^3}{3!} + \frac{x^5}{4!} + \frac{x^7}{7!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$ <p>х изменяется в диапазоне $[0.1;1]$, $y = \frac{e^x - e^{-x}}{2}$</p>



Кафедра ПМИ

Начало

Содержание



Страница 19 из 42

Назад

На весь экран

Заккрыть

Краткие теоретические сведения:

КМВ циклы

Циклы с контролем за монотонной величиной относятся к итерационным циклам, в которых заранее неизвестно количество повторений, но имеется некоторое условие, при выполнении которого вычисления можно прекратить.

В качестве монотонной величины, за которой может вестись контроль окончания цикла при вычислении сумм членов бесконечной последовательности (ряда), может выступать: а) модуль разности точного и приближенного значения ряда (функции); б) модуль разности следующего и предыдущего члена последовательности (ряда); в) модуль «отбрасываемого» слагаемого (вычисленного на данном или предыдущем шаге итерации) и др.

В программировании на языке C++ КМВ циклы реализуются с помощью оператора **while**, **do...while**.

Оператор цикла с предусловием

while (выражение_условие) тело_цикла;

Организует повторение операторов тела_цикла до тех пор, пока выражение_условие истинно (отлично от нуля).



*Кафедра
ПМИ*

Начало

Содержание



Страница 20 из 42

Назад

На весь экран

Заккрыть

Оператор цикла с постусловием

do тело_цикла **while**(выражение_условие);

Организует повторение операторов тела_цикла до тех пор, пока выражение_условие истинно (отлично от нуля).

Тело_цикла **do...while** выполняется, по крайней мере, один раз.

Пример

Вычислить сумму ряда $S = x + \frac{x^3}{3!} + \frac{x^5}{4!} + \frac{x^7}{7!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$ с точностью $\text{eps}=10^{-5}$, если x изменяется в диапазоне $[0.1;1]$, точное значение функции $y = \frac{e^x - e^{-x}}{2}$.

Программный код примера можно увидеть [здесь](#).



Кафедра
ПМИ

Начало

Содержание



Страница 21 из 42

Назад

На весь экран

Заккрыть

. Лабораторная работа 4. Итерационные циклы

Цель: получать практические умения реализации консольных приложений на языке C++ с использованием операторов цикла **while**, **do...while**.

Оборудование и ПО: ПК, среда разработки приложений для языков программирования C/C++ (DEV-Cpp или MS Visual Studio).

Содержание работы: написать программу, в которой реализовано решение поставленной задачи с использованием классов потокового ввода и вывода **iostream**.

Порядок выполнения работы:

1. Изучить синтаксис и семантику операторов **while** и **do...while**.
2. Изучить теоретический материал об операторах циклов в языке программирования C++, пример первоисточника:

Подбельский В.В., Язык C++: Учеб.пособие. - 5-е изд. - М: Финансы и статистика, 2003. С. 97-106.

3. Изучить материал об **итерационных методах нахождения корня уравнения на отрезке**.
4. Познакомиться с примерами программного кода для методов **дихотомии**, **хорд** и **касательных**.
4. Выполнить **задание**, соответствующее собственному варианту.



Кафедра
ПМИ

Начало

Содержание



Страница 22 из 42

Назад

На весь экран

Заккрыть

Задание лабораторной работы:

Дана функция $f(x)$, известно, что на отрезке $[a;b]$ функция имеет один корень x_k ($f(x_k)=0$). Вычислить корень с точностью **eps** методом дихотомии, хорд и касательных.

Ввести **a**, **b**, **eps** с клавиатуры.

Реализовать поиск корня с помощью функций, возвращающих количество итераций в методе (четвертый параметр функции – значение корня):

int dihotomia (**double, double, double, double**) – функция, реализующая метод дихотомии.

int horda (**double, double, double, double**) – функция, реализующая метод хорд.

int newton (**double, double, double, double**) – функция, реализующая метод касательных.

Все вышеозначенные функции описать в отдельном заголовочном файле проекта – methods.h.

Вывести на консоль информацию о методе, значении корня для каждого из методов, а также количество итераций для получения приближенного значения корня в каждом из методов.



Кафедра
ПМИ

Начало

Содержание



Страница 23 из 42

Назад

На весь экран

Заккрыть

Вар	Уравнение вида $f(x) = 0$, промежуток $[a; b]$, содержащий корень
1	$x^2 - \frac{1}{x} + 3 = 0, [a; b] = [0.1; 1]$
2	$\ln(x^2 - 5) - \frac{1}{x} = 0, [a; b] = [2.3; 3]$
3	$x^2 \cos(2x) + 1 = 0, [a; b] = [0.5; 1.5]$
4	$2x^2 - 2\cos(x^3) = 0, [a; b] = [0.2; 1]$
5	$\operatorname{tg}(x) - x + 1 = 0, [a; b] = [-1.4; -0.4]$
6	$x^5 - \cos(x) + 25 = 0, [a; b] = [-3; -1]$
7	$x^4 + \cos(x) - 3 = 0, [a; b] = [0.1; 2]$
8	$x^4 + \frac{1}{\cos^2(x)} = 0, [a; b] = [0.1; 1.4]$
9	$\ln(x^4 + x^3) + \cos(x) = 0, [a; b] = [0.1; 1.4]$



*Кафедра
ПМИ*

Начало

Содержание



Страница 24 из 42

Назад

На весь экран

Заккрыть

Краткие теоретические сведения:

Итерационные методы

нахождения корня уравнения на отрезке

Основными методами нахождения корня функции $f(x)$ с известной точностью ε на данном интервале $[a; b]$ являются метод **дихотомии** (метод половинного деления), метод **хорд** (секущих) и метод **касательных** (метод Ньютона). Правильное решение уравнения указанными методами возможно лишь в том случае, если известно, что на заданном интервале имеется корень и он единственный. Это не означает, что методы могут использоваться только для решения линейных уравнений. Для нахождения корней уравнений более высокого порядка необходимо сначала отделить корни по отрезкам. Процесс отделения корней осуществляется путем отыскания первой и второй производной от функции и приравнивания их нулю $f'(x)=0$ и $f''(x)=0$. Далее определяются знаки $f(x)$ в критических и граничных точках. Интервал, где функция меняет знак $[a; b]$, где $f(a) \cdot f(b) < 0$.



Кафедра
ПМИ

Начало

Содержание



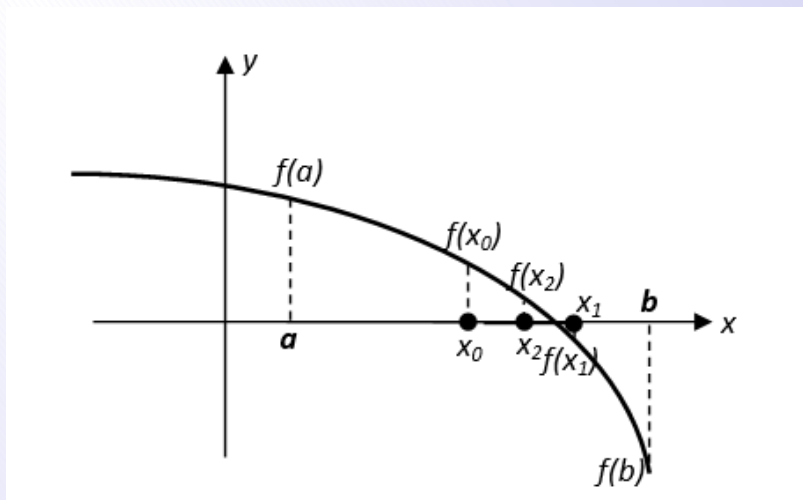
Страница 25 из 42

Назад

На весь экран

Заккрыть

Метод дихотомии



Алгоритм метода дихотомии (половинного деления) состоит в следующем:

Пусть дана функция $f(x)$, известно, что на отрезке $[a; b]$ функция имеет один корень x_k ($f(x_k)=0$). Вычислить корень с точностью ε .

1. На первом этапе вычисляется $x_0 = (a+b)/2$;
2. Далее определяется значение функции в этой точке:

(а) если $f(x_0) \cdot f(a) < 0$, то корень находится на отрезке $[a, x_0]$ ($b = x_0$);



Кафедра
ПМИ

Начало

Содержание



Страница 26 из 42

Назад

На весь экран

Заккрыть



(b) если $f(x_0) \cdot f(a) > 0$, тогда $f(x_0) \cdot f(b) < 0$, тогда корень находится на отрезке $[x_0, b]$ ($a = x_0$), т.о. происходит сужение интервала, в результате формируется последовательность x_i , где i - номер итерации.

3. Повторяем этап 1, 2 до тех пор, пока разность $b - a$ не станет меньше требуемой погрешности ε .

Пример 1

В качестве примера использования метода половинного деления найдем корень на интервале $[0; 1]$ уравнения $x^3 - 3x + 1 = 0$ с точностью 10^{-3} .

Предварительные вычисления: $f(x) = x^3 - 3x + 1$; $a = 0, b = 1$; $\varepsilon = 0.001$.

1. $x_0 = (a + b) / 2 = 0.5$; $f(a) = f(0) = 1$; $f(b) = f(1) = -1$

2. $f(x_0) = f(0.5) = -0.375$, т.к. $f(x_0) \cdot f(a) < 0$, то $b = x_0$;

3. $[a; b] = [0; 0.5]$; $b - a = 0.5 > \varepsilon = 0.001$;

1'. $x_1 = (a + b) / 2 = 0.25$; $f(a) = f(0) = 1$; $f(b) = f(0.5) = -0.375$;

2'. $f(x_1) = f(0.25) = 0.265625$, .. $f(x_1) \cdot f(b) < 0$;

3'. $[a; b] = [0.25; 0.5]$; $b - a = 0.25 > \varepsilon = 0.001$;

$$1''.x_2 = (a + b)/2 = 0.375; f(a) = f(0.25) = 0.265625; f(b) = f(0.5) = -0.375$$

$$2''.f(x_2) = f(0.375) = -0,072265625, ..f(x_2) * f(a) < 0, b = x_2;$$

$$3''.[a; b] = [0.25; 0.375]; b-a=0.125 > \varepsilon = 0.001;$$

$$1'''.x_3 = (a + b)/2 = 0.3125; f(a) = f(0.25) = 0.2656; f(b) = f(0.375) = -0,0722656;$$

...

$$x_n=0.34668. b-a=0.0009766<\varepsilon.$$

x_n – корень $f(x)$ с точностью 10^{-3} .

Фрагмент кода данного примера можно посмотреть [здесь](#).



Кафедра
ПМИ

Начало

Содержание



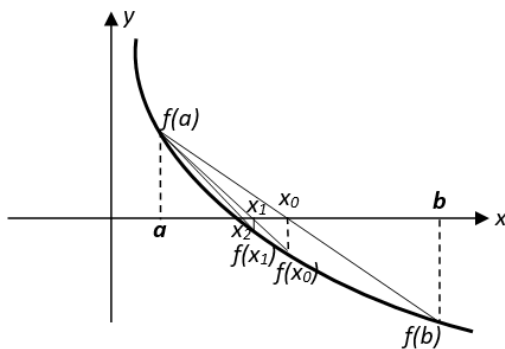
Страница 28 из 42

Назад

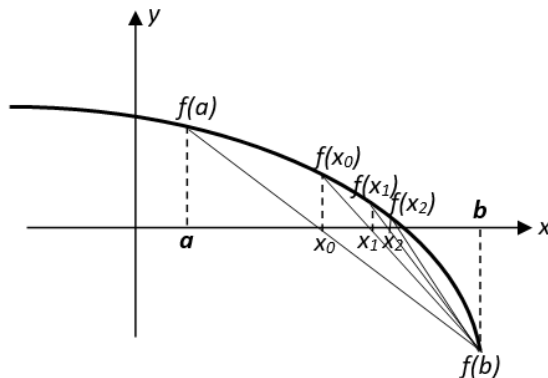
На весь экран

Закрыть

Метод хорд (секущих)



а) a – «неподвижная» точка
 $f(a) \cdot f''(a) > 0$



а) b – «неподвижная» точка
 $f(b) \cdot f''(b) > 0$

Суть алгоритма поиска корня (нуля) функции $f(x)$ (уравнения $f(x)=0$)



Кафедра
ПМИ

Начало

Содержание



Страница 29 из 42

Назад

На весь экран

Закреть

на отрезке $[a;b]$ с точностью ε методом хорд (секущих) состоит в следующем:

1. Делаем предварительные вычисления, находим «неподвижную» точку:
 - (a) Находим $f(x)$, $f'(x)$, $f''(x)$, а также $f(a)$, $f'(a)$, $f''(a)$, $f(b)$, $f'(b)$, $f''(b)$;
 - (b) Проверяем, какая из точек a или b соответствует условию «неподвижной» точки $f(a) * f''(a) > 0$ или $f(b) * f''(b) > 0$.
 - (c) В алгоритме для простоты объяснения вычисления будем проводить для неподвижной точки a .
2. Проводим прямую через точки кривой $y=f(x)$ с координатами $(a;f(a))$, $(b;f(b))$, уравнение прямой, проходящей через указанные точки: $\frac{x-a}{b-a} = \frac{y-f(a)}{f(b)-f(a)}$,
3. Находим точку пересечения прямой с осью ОХ ($y=0$), $x_0 = a - \frac{f(a)(b-a)}{f(b)-f(a)}$,
4. Находим значение функции в точке x_0 . Сравниваем $f(x_0)$ с ε .
5. Если $|f(x_0)| < \varepsilon$, то завершаем алгоритм, иначе – повторяем шаги 2-5, сужая промежуток до $[a, x_0]$, т.е. $b=x_0$.



Кафедра
ПМИ

Начало

Содержание



Страница 30 из 42

Назад

На весь экран

Заккрыть

Пример 2

В качестве примера использования метода хорд (секущих) найдем корень на интервале $[0.2; 1]$ уравнения $x^3 - 3x + 1 = 0$ с точностью 10^{-3} .

Предварительные вычисления: $f(x) = x^3 - 3x + 1$; $a = 0.2, b = 1$; $\varepsilon = 0.001$.

$$1. f'(x) = 3x^2 - 3; f''(x) = 6x; f(a) = f(0.2) = 0.408; \\ f''(a) = f''(0.2) = 1.2; f(b) = f(1) = -1; f''(b) = f''(1) = 6.$$

Т.о. a – «неподвижная» точка.

$$2. x_0 = 0.2 - \frac{f(0.2)(1-0.2)}{f(1)-f(0.2)} = 0.2 - \frac{0.408(1-0.2)}{-1-0.408} = 0.4318$$

$$3. f(x_0) = -0.2149.$$

$$4. |f(x_0)| = 0.2149 > 0.001. b = 0.4318.$$

$$2'. x_1 = 0.2 - \frac{f(0.2)(0.4318-0.2)}{f(0.4318)-f(0.2)} = 0.2 - \frac{0.408*0.2318}{-0.2149-0.408} = 0.3518.$$

$$3'. f(x_1) = -0.0119.$$

$$4'. |f(x_1)| = 0.0119 > 0.001. b = 0.3518.$$

$$2'. x_2 = 0.2 - \frac{f(0.2)(0.3518-0.2)}{f(0.3518)-f(0.2)} = 0.2 - \frac{0.408*0.1518}{-0.0119-0.408} = 0.3475.$$



Кафедра
ПМИ

Начало

Содержание



Страница 31 из 42

Назад

На весь экран

Заккрыть

$$3'.f(x_2) = -0.00057.$$

$4'.|f(x_2) = 0, 00057 < 0.001$, следовательно, $x = 0.3475$ – корень уравнения $f(x) = 0$ на промежутке $[0.2;1]$ с точностью 0.001.

Фрагмент кода данного примера можно посмотреть [здесь](#).



Кафедра
ПМИ

Начало

Содержание



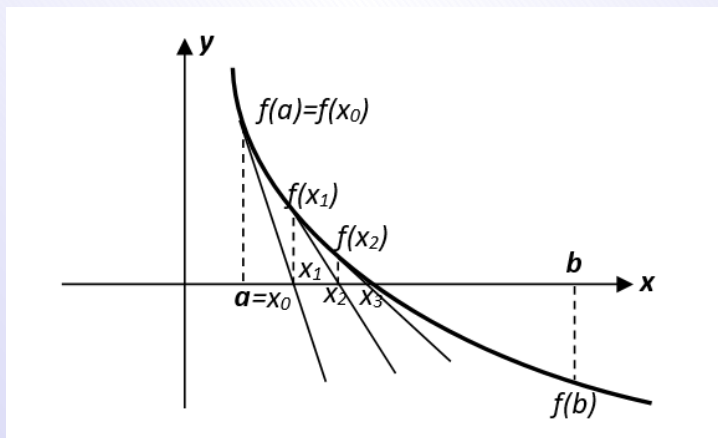
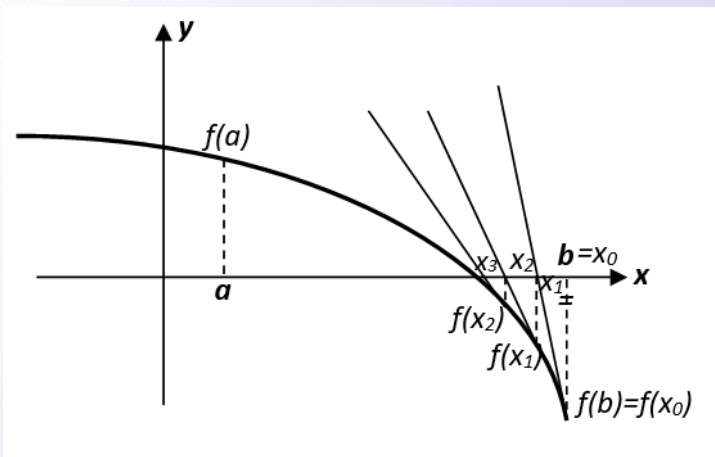
Страница 32 из 42

Назад

На весь экран

Закрыть

Метод касательных (Ньютона)



При нахождении корня нелинейного уравнения $f(x)=0$ с точностью ϵ методом касательных (Ньютона) дается начальное приближение x_0 ,



Кафедра
ПМИ

Начало

Содержание



Страница 33 из 42

Назад

На весь экран

Заккрыть

если же начальное приближение не определено, и поиск корня ведется на промежутке $[a;b]$, то в качестве начального приближения берут одну из точек a или b . Как правило, в качестве начального приближения удобно брать «неподвижную» точку, указанную в методе хорд, т.е. ту точку, для которой выполняется условие $f(x) \cdot f''(x) > 0$.

Суть метода касательных (Ньютона) при нахождении корня нелинейного уравнения $f(x)=0$ с точностью ε на отрезке $[a;b]$ состоит в следующем:

1. Делаем предварительные вычисления, находим «неподвижную» точку – начальное приближение:
 - (a) Находим $f(x)$, $f'(x)$, $f''(x)$, а также $f(a)$, $f'(a)$, $f''(a)$, $f(b)$, $f'(b)$, $f''(b)$;
 - (b) Проверяем, какая из точек a или b соответствует условию «неподвижной» точки $f(a) \cdot f''(a) > 0$ или $f(b) \cdot f''(b) > 0$.
 - (c) В алгоритме для простоты объяснения вычисления будем проводить для неподвижной точки a . Начальное приближение $x = x_0 = a$.
2. Проводим касательную к графику кривой $y=f(x)$ в точке $(x_0; f(x_0))$, уравнение касательной имеет вид: $y = f(x_0) + f'(x_0)(x - x_0)$.
3. Находим точку пересечения прямой (касательной) с осью OX ($y=0$), $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, т.е. очередное приближение.



Кафедра
ПМИ

Начало

Содержание



Страница 34 из 42

Назад

На весь экран

Заккрыть

4. Находим $f(x_1)$ и проверяем условие окончания.

5. Если $|f(x_1)| < \varepsilon$, то завершаем алгоритм (корень равен x_1), иначе – повторяем шаги 2-5, для значения $x=x_1$.

Пример 3

В качестве примера использования метода касательных (Ньютона) найдем корень на интервале $[0.2;1]$ уравнения $x^3-3x+1=0$ с точностью 10^{-3} . Предварительные вычисления: $f(x)=x^3-3x+1$; $a=0.2, b=1$; $\varepsilon=0.001$.

$$1. f'(x) = 3x^2-3; f''(x) = 6x; f(a) = f(0.2) = 0.408; \\ f''(a) = f''(0.2) = 1.2; f(b) = f(1) = -1; f''(b) = f''(1) = 6.$$

Т.о. a – «неподвижная» точка, в качестве начального приближения возьмем $x_0=a=0.2$.

$$2. f(x_0) = 0.408, f'(x_0) = -2.88;$$

$$3. x_1 = 0.2 - 0.408 / (-2.88) = 0.34167;$$

$$4. f(x_1) = 0.01488;$$

$$5. |f(x_1)| = 0.01488 > 0.001;$$

$$2'. f(x_1) = 0.01488, f'(x_1) = -2.64979;$$

$$3'. x_2 = 0.34167 - 0.01488 / (-2.64979) = 0.34728;$$



Кафедра
ПМИ

Начало

Содержание



Страница 35 из 42

Назад

На весь экран

Заккрыть

$$4'. f(x_2) = 0,00003;$$

5'. $|f(x_2)| = 0,00003 < 0.001$, следовательно, $x=0,34728$ – корень уравнения $f(x)=0$ на промежутке $[0.2;1]$ с точностью 0.001.

Фрагмент кода данного примера можно посмотреть [здесь](#).



*Кафедра
ПМИ*

Начало

Содержание



Страница 36 из 42

Назад

На весь экран

Закрыть

ТЕМА 3 ФУНКЦИИ

. Лабораторная работа 5. Описание и определение функций



*Кафедра
ПМИ*

Начало

Содержание



Страница 37 из 42

Назад

На весь экран

Закреть

. Лабораторная работа 6. Перегрузка функций, шаблоны функций



*Кафедра
ПМИ*

Начало

Содержание



Страница 38 из 42

Назад

На весь экран

Закрыть

ТЕМА 4 ТЕКСТОВЫЕ ФАЙЛЫ

. Лабораторная работа 7. Обработка текстовых файлов



*Кафедра
ПМИ*

Начало

Содержание



Страница 39 из 42

Назад

На весь экран

Закрыть

ТЕМА 5

СТРУКТУРЫ И ОБЪЕДИНЕНИЯ

. Лабораторная работа 8. Структуры



*Кафедра
ПМИ*

Начало

Содержание



Страница 40 из 42

Назад

На весь экран

Закрыть

ТЕМА 6 КЛАССЫ

- Лабораторная работа 9-11. Классы. Конструкторы и деструкторы класса. Компоненты класса. Перегрузка операций



*Кафедра
ПМИ*

Начало

Содержание



Страница 41 из 42

Назад

На весь экран

Закрыть

ТЕМА 7 НАСЛЕДОВАНИЕ

- Лабораторная работа 12-13. Наследование классов, виртуальные функции, абстрактные классы



*Кафедра
ПМИ*

Начало

Содержание



Страница 42 из 42

Назад

На весь экран

Закрыть