

Министерство науки и высшего образования Российской Федерации
Казанский национальный исследовательский технический университет –
КАИ им. А.Н. Туполева

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ «Колледж информационных технологий»

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ

Методические указания к лабораторным работам

Казань 2023

Составитель преподаватель СПО ИКТЗИ Мингалиев Заид Зульфатович

Методические указания к лабораторным работам по дисциплине «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ» предназначены для студентов направления подготовки 09.02.07 «Информационные системы и программирование»

ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ

По итогам выполнения работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретённые знания и навыки отвечает на пару небольших вопросов преподавателя по составленной программе, возможностям её доработки;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Шаблон оформления отчета представлен в приложении 1. Требования к формированию отчета представлены в приложении 2.

ЛАБОРАТОРНАЯ РАБОТА №2.

Программирование алгоритмов разветвляющейся структуры.

ЦЕЛЬ РАБОТЫ

Приобрести умения и практические навыки для программирования алгоритмов разветвляющейся структуры.

ХОД РАБОТЫ

1) Условные конструкции при построении алгоритмов.

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, в результате проверки которого выполняется либо одна, либо другая последовательность действий.

Выбор направления выполнения алгоритма зависит от исходных или промежуточных данных.

Самыми простыми условными ветвлениями в языке C++ являются операторы if/else.

Чтобы указать условие, при выполнении которого будет выполняться определенный код, потребуется прибегнуть к следующей структуре:

1. набрать ключевое слово **if** в своей программе;
2. после него, в круглых скобках, вы должны указать логическое выражение:

if (<выражение>)

Здесь вместо <выражение> нужно указать требуемое условие, при котором будет выполняться требуемый нам код.

3. инструкции, которые будут выполняться, мы оборачиваем в фигурные скобки сразу после круглых скобок с условием:

```
if (выражение)  
{  
<выполняемый код>  
}
```

Также можно опустить фигурные скобки. Это можно сделать только в том случае, если в качестве исполняемого кода будет указана лишь одна инструкция (или один оператор, в том числе другой оператор ветвления, который уже может содержать любое количество инструкций)

Оператор **else** выполняется в том случае, если условие указанное в **if** ложно. В таком случае будет проигнорирован код в блоке **if**, и программа сразу перейдет к выполнению кода в блоке **else**.

Данный оператор нужно прописывать после закрывающей фигурной скобки блока кода **if** или после единственной инструкции оператора **if**.

Общая структура условной конструкции **if/else** выглядит следующим образом:

```

if (выражение)
    {
        <выполняемый код>
    }
    else
    {
        <оператор2 (операторы2)>
    }

```

Если результатом выражения является **true** (истина, любое ненулевое значение), то выполняться будет **оператор1**. Если же результатом выражения является **false** (ложь, 0), то выполняться будет **оператор2**.

Например:

1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	<code>int main()</code>
4	<code>{</code>
5	<code> cout << "Enter a number: ";</code>
6	<code> int a;</code>

7	<code>cin >> a;</code>
8	<code>if (a > 15)</code>
9	<code>cout << "Число " << a << " больше чем 15\n";</code>
10	<code>else</code>
11	<code>cout << "Число " << a << " меньше чем 15\n";</code>
12	<code>return 0;</code>
13	<code>}</code>

Оператор `if` выполняет только одну операцию, если выражение является `true`, и также только одну операцию `else`, если выражение — `false`. Для того, чтобы выполнить несколько операций подряд, используется блок операторов, которые ограничены фигурными скобками `{...}`:

1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	<code>int main()</code>
4	<code>{</code>
5	<code>cout << "Enter a number: ";</code>
6	<code>int a;</code>
7	<code>cin >> a;</code>
8	<code>if (a > 15)</code>
9	<code>{</code>
10	<code>// Блок операторов ограничивается фигурными скобками {...}</code>
11	<code>// Обе операции будут выполнены, если a > 15</code>
12	<code>cout << "Было введено число " << a << "\n";</code>
13	<code>cout << "Число " << a << " больше чем 15\n";</code>
14	<code>}</code>
15	<code>else</code>
16	<code>{</code>

17	// Блок операторов ограничивается фигурными скобками {...}
18	// Обе операции будут выполнены, если $a \leq 15$
19	cout << "Было введено число " << a << "\n";
20	cout << "Число " << a << " меньше чем 15\n";
21	}
22	return 0;
23	}

Если программист не указал скобки для блока операторов `if` или `else`, то компилятор неявно сделает это за него.

Однако нередко надо обработать не два возможных альтернативных варианта, а гораздо больше. Например, в случае выше можно насчитать три условия: переменная x может быть больше 5, меньше 5 и равна 5. Для проверки альтернативных условий мы можем вводить выражения `else if`:

1	<code>if(x > 5)</code>
2	{
3	cout << "x больше чем 5 \n";
4	}
5	<code>else if (x < 5)</code>
6	{
7	cout << "x меньше чем 5 \n";
8	}
9	<code>else</code>
10	{
11	cout << "x равно 5 \n";
12	}

То есть в данном случае мы получаем три ветки развития событий в программе.

`switch` – это множественный условный оператор. Он работает также как несколько условий `if`, но может проверять лишь одну переменную в своих условиях. Если одно условие является правильным происходит выход.

Для его использования не нужно подключать какие-то дополнительные библиотеки, кроме стандартной `<iostream>`.

1	<code>switch (имя переменной) {...}</code>
---	--

Для проверки условия нужно использовать конструкцию `case`:

1	<code>switch (имя переменной)</code>
2	<code>{</code>
3	<code>case <проверяемое значение>:</code>
4	<code>//Блок кода</code>
5	<code>break;</code>
6	<code>}</code>

После слова `case` используются `:` (двоеточие), а в самом ее конце должен присутствовать оператор `break`.

Если мы хотим также обработать ситуацию, когда совпадения не будет найдено, то можно добавить блок `default`.

1	<code>int x = 9;</code>
2	<code>switch (x) {</code>
3	<code>case 5:</code>
4	<code>cout << "Значение равно 5";</code>
5	<code>break;</code>
6	<code>case 6:</code>
7	<code>cout << "Значение равно 6";</code>
8	<code>break;</code>
9	<code>case 0:</code>
10	<code>cout << "Значение равно zero";</code>
11	<code>break;</code>

12	default:
13	cout << "Значение не равно 5, 6 и даже 0";
14	}

Если ни одно `case` и `default` не правильны, то программа выйдет из `switch` и продолжит работать дальше.

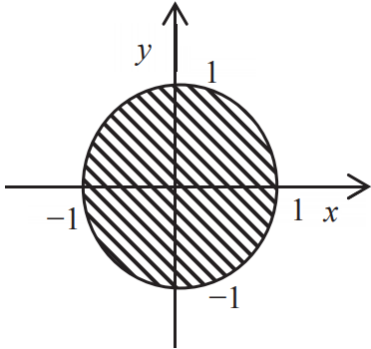
2) Задание на лабораторную работу.

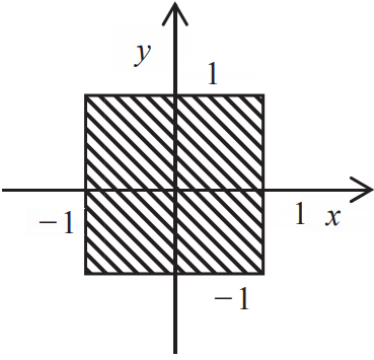
Обе части лабораторной работы должны быть выполнены либо на языке программирования C++, либо на языке программирования C#.

Часть 1

В соответствии с вариантом необходимо написать консольную программу.

Варианты	Индивидуальное задание
1	Напишите программу, которая принимает возраст пользователя и выводит сообщение о его статусе: "ребенок", "подросток", "взрослый" или "пожилой".
2	По заданным x, y составить программу вычисления значения z : $z = \frac{\min(x, y) + 0,5}{1 + \max^2(x, y)}$
3	По заданным x, y составить программу вычисления значения z : $z = \begin{cases} \max(x, y), & \text{при } x < 0 \\ \min(x, y), & \text{при } x \geq 0 \end{cases}$
4	Даны действительные числа x, y . определить, принадлежит ли точка с координатами (x, y) заштрихованной части плоскости.

Варианты	Индивидуальное задание
	
5	<p>Даны действительные числа x, y, z. Получить:</p> $L = 2 * \max(x, z) - 3 * \min(x, y, z)$
6	<p>По заданным вещественным числам a, b, c вычислить:</p> $P = \frac{\max(a, b, c) - \min(a, b, c)}{2}$
7	<p>Напишите программу, которая принимает координаты точки (x, y) и определяет, в какой четверти координатной плоскости она находится.</p>
8	<p>Напишите программу, которая принимает длины трех сторон треугольника и определяет его тип: "равносторонний", "равнобедренный" или "разносторонний".</p>
9	<p>Даны действительные числа a, b, c. удвоить эти числа, если $a \geq b \geq c$, и заменить их абсолютными значениями (значение по модулю), если это не так.</p>
10	<p>Напишите программу, которая принимает возраст пользователя и проверяет, может ли он посещать определенные мероприятия.</p>
11	<p>Даны действительные положительные числа x, y, z. Выяснить, существует ли треугольник с длинами x, y, z.</p>
12	<p>Дано действительное число a. Вычислить $f(a)$, если</p> $f(x) = \begin{cases} x^2 + 4x + 5, & \text{при } x \leq 2 \\ \frac{1}{x^2 + 4x + 5}, & \text{в противном случае} \end{cases}$

Варианты	Индивидуальное задание
13	Напишите программу, которая определяет сезон года на основе введенного месяца (1-12).
14	Написать программу определения стоимости разговора по телефону с учетом скидки 20%, предоставляемой по воскресеньям. У пользователя должен запрашиваться день недели, в который производится разговор.
15	Даны три действительные числа. Если все числа положительны, найти среднее арифметическое, иначе произведение.
16	Даны три действительные числа. Возвести в квадрат те из них, значения которых положительны, и в четвертую степень — отрицательные.
17	Даны три действительных числа. выбрать из них те, которые принадлежат полуинтервалу (3, 7].
18	По заданным x, y составить программу вычисления значения z : $z = \begin{cases} \max(x^2, y^2), & \text{при } y < 0 \\ \min(x, y), & \text{при } y \geq 0 \end{cases}$
19	Даны действительные числа x, y . определить, принадлежит ли точка с координатами (x, y) заштрихованной части плоскости. 
20	По четырехзначному номеру трамвайного билета определить, является ли его номер зеркальным.

Варианты	Индивидуальное задание
21	Напишите программу, которая принимает три числа и выводит наибольшее из них.
22	Напишите программу, которая принимает оценку студента по математике и литературе, и выводит сообщение о его успеваемости: "отлично", "хорошо", "удовлетворительно" или "неудовлетворительно", в зависимости от среднего балла.
23	Напишите программу, которая определяет, является ли введенный год високосным или нет.
24	Создайте программу, которая принимает рост и вес пользователя, а затем вычисляет его индекс массы тела (ИМТ) и выводит соответствующую категорию: "недостаточный вес", "нормальный вес", "избыточный вес" или "ожирение".
25	Напишите программу, которая проверяет, является ли введенное число положительным, отрицательным или нулем.

Часть 2.

Разработать простой тест: на экране по очереди появляются вопросы (вопросы выбираются студентом), с вариантами ответов. В конце работы программа выдает количество заработанных баллов по результатам ответа. Минимальное количество вопросов – пять.

3) Контрольные вопросы

1. Что такое тип данных? Какие типы данных существуют для хранения числовых данных?
2. Какие условные операторы используются при реализации алгоритмов разветвляющейся структуры?
3. Какие операторы используются при создании сложных логических выражений?