Язык программирования С++. Ветвление. If – else.

Цель урока:

1. Повторение “cin”, “cout”, создание и использование переменных.
2. Изучение трех основных алгоритмов.
3. Изучение ветки “if-else”.
4. Создание теста для пользователя с помощью “if-else”.

Оборудование:

|  |  |  |
| --- | --- | --- |
| № | Комплектующие | Количество |
| 1 | Платы Arduino Uno R3 или WAVGAT Uno R3 | 12 |
| 2 | Пьезоэлементы | 12+5 |
| 3 | Кнопки | 12+5 |
| 4 | Провода типа вилка-вилка | 12\*5 |

План урока:

1. Учитель объявляет тему урока и проводит опрос: cout, cin, что такое переменные.
2. Ученики собирают схему на Ардуино.
3. Объясняется, какие основные три алгоритма существует в программировании.
4. На прошлом уроке дети проходили линейный алгоритм.
5. Объясняется, что такое “алгоритм ветвления”.
6. Детям дается задание дать математический пример пользователю.
7. На примере введенных пользователем данных объясняется ветка “if-else”.
8. Далее дети создают свой тест из нескольких вопросов, в котором на экране будет выводиться: правильно ли ответил пользователь на тот или иной вопрос.
9. \* Дети создают переменную, которая показывает количество правильных ответов.
10. В конце урока делается анонс следующего урока по созданию интерфейса языка программирования на узбекском языке.

Здравствуйте, ребята. Сегодня мы продолжим свое путешествие по миру программирования компьютерных программ. Прошлый раз мы с вами проходили основной ввод-вывод информации и создание переменных в языке программирования С++. Кстати, у русскоговорящих программистов сложилась традиция говорить “ЯП” вместо “Язык программирования”.

Давайте, вместе вспомним:

1. Что такое основной вывод информации “cout”? Как он расшифровывается?
2. Что такое переменная? Как с ней работать?
3. Что такое основной ввод информации “cin”? Как он расшифровывается?
4. Можно ли использовать “cin” не создавая переменные?

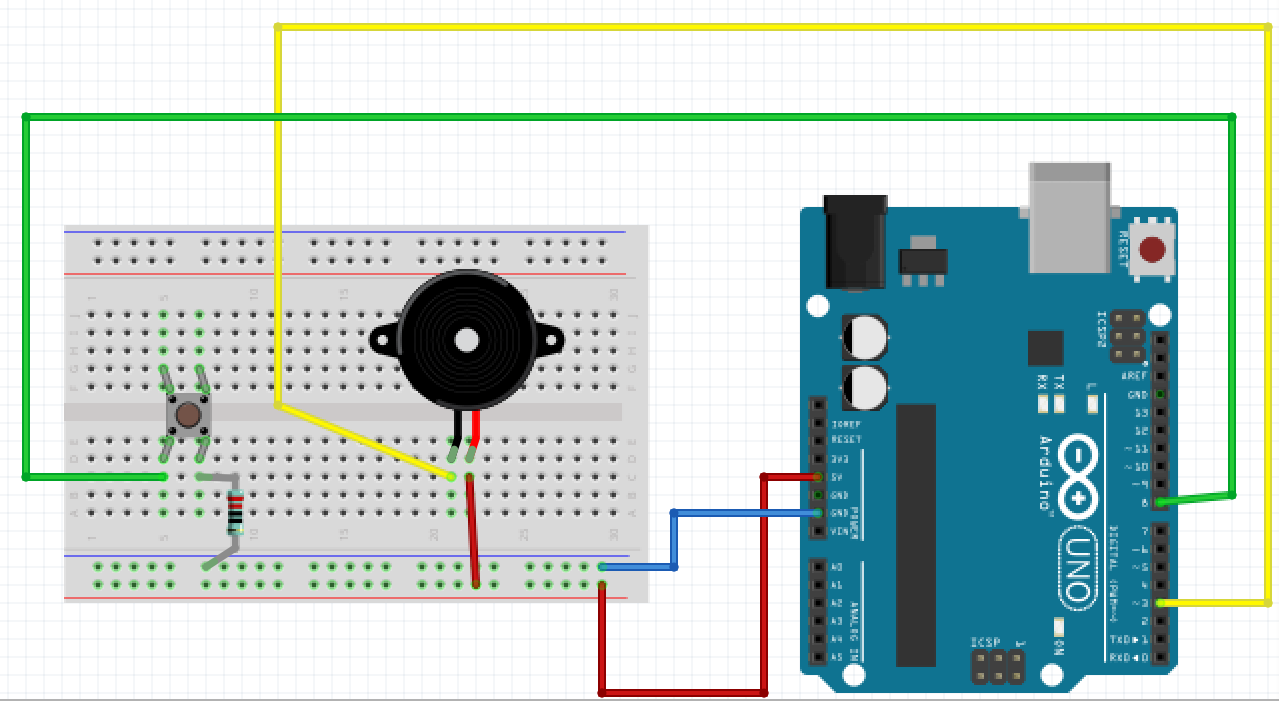
Сегодня же мы с вами разберемся с тем, что такое основные алгоритмы и пройдем один из них.

В программировании существует три основных алгоритма, без которых невозможно создать ни одну даже самую простую программу.

Вот эти три основных алгоритма:

1. Линейный .
2. Разветвляющийся (условный).
3. Циклический.

Для начала, чтобы не забывать об Arduino, мы с вами соберем схему из пьезопищалки и кнопки. Пьезопищалка - это просто маленькая колонка.

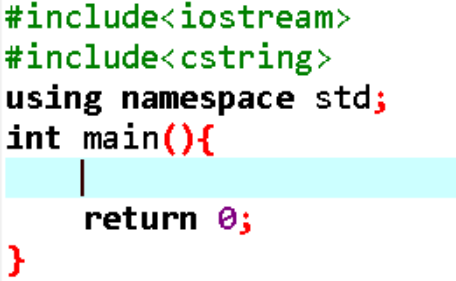
Вот схема, по которой вы сможете собрать свою машинку для азбуки Морзе:

Я вам сразу скажу, что на прошлом уроке вы узнали один из трех основных алгоритмах – это линейный. Линейный алгоритм – это когда действия происходят друг за другом, вне зависимости от чего либо. Вспомните, после написания кода, происходило одно и то же: вы вводили имя и возраст, а компьютер их выводил.

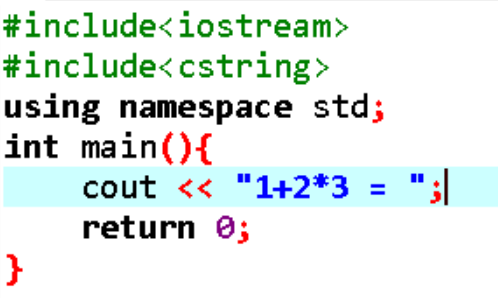
Сегодня мы пройдем алгоритм ветвления, его еще называют веткой if-else.

Дело в том, что только линейный алгоритм не всегда подходит под многие задачи. Например, если вы пытаетесь зайти в аккаунт, то вы зайдете только в том случае, если вы правильно ввели и логин и пароль. Вот так же и со всеми программами на свете. Разным пользователям нужны разные функции приложения. Программист дает возможность пользователю делать, то что захочет именно пользователь, а не все то, что создал программист, как говорится, “время - деньги”.

Для начала, давайте, напишем шаблон нашей программы:

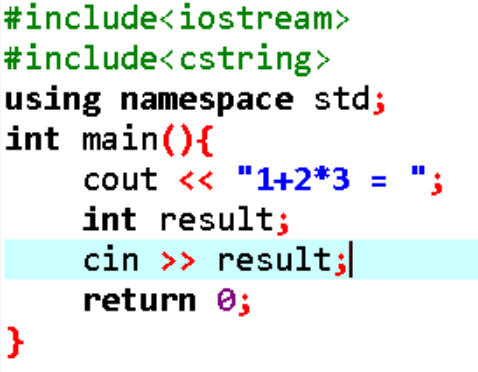


Теперь зададим пользователю пример по математике, только сделайте простой, я же сделаю вот так.



Откомпилируйте и запустите программу.

Далее дадим возможность пользователю решить этот пример. Как же мы сможем предоставить ему шанс? Правильно: создаем переменную и используем “cin”, чтобы пользователь ввел результат.

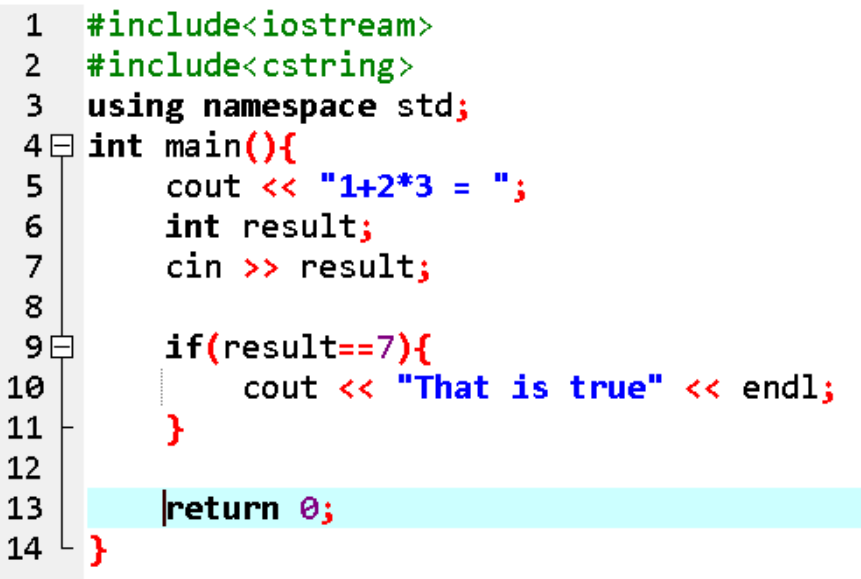


Откомпилируйте и запустите программу.

Теперь логичный вопрос: как же пользователь поймет, правильно ли он ответил? А если он ответит неправильно? Мы же не можем вывести ему надпись, что он ответил и верно и не верно? Представьте, что было бы, если учитель по математике на каждый пример, который вы решаете, говорил “Да, правильно. Нет, неправильно” вне зависимости от настоящего результата.

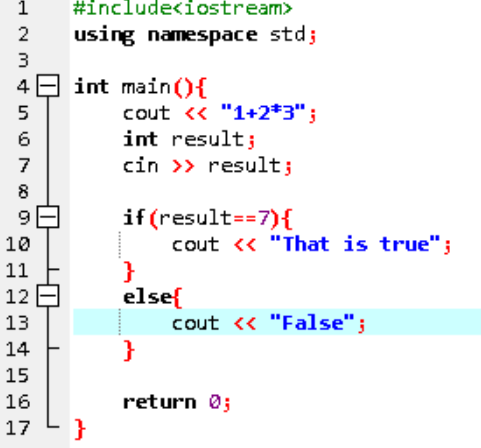
Здесь нам на помощь приходит ветка if-else.

Добавьте следующие строки:



Откомпилируйте и запустите программу.

Попробуйте решить пример сначала правильно и посмотрите, какое сообщение показывает наша программа пользователю, затем дайте неправильный ответ, что происходит на экране?



Запустите программу и проверьте результат ее работы.

Задания:

1. Задавайте пользователю вопросы, на которые они будут отвечать строчками текста “yes” и “no”. После того, как пользователь ответил на один вопрос ему выходит сообщение о правильности ответа, затем выходит следующий вопрос. И так несколько вопросов.
2. Создайте переменную, в которой будет храниться количество правильно решенных пользователем заданий.
3. Создайте переменную, в которой будет храниться количество неверно решенных пользователем заданий.
4. В конце программы выведите количество правильных и неправильных ответов.