

Cubico

Coding play AR

ПРОГРАММИРОВАНИЕ ДЛЯ ДЕТЕЙ

Наглядные и осязаемые средства обучения
программированию для детей дошкольного возраста



Дополненная
РЕАЛЬНОСТЬ



Смартфон
Планшет



Решение задач
Вычислительное
мышление

Навыки для успеха в XXI веке!

Креативность и
воображение

Критическое мышление
и решение задач

Сотрудничество и
работа в команде

Школьные учителя, эксперты в области образования и лидеры бизнеса совместно разработали программу обучения навыкам, которые понадобятся нашим детям, чтобы преуспеть в работе и в социальной и личной жизни в XXI веке.

Начните с Cubico – программы обучения программированию на платформе дополненной реальности! Позвольте Вашим детям получить важные навыки, которые пригодятся им в будущем.

Вычислительное мышление

«Вычислительное мышление – это фундаментальный навык для каждого, не только для специалистов по информатике.

К навыкам чтения, письма и арифметических действий следует добавить вычислительное мышление, чтобы развить аналитические способности каждого ребенка.»

– Жанет Винг, вице-президент Microsoft Research

Вычислительное мышление стало ключевым понятием цифровой эпохи. Оно означает мыслительные процессы, связанные с формулированием проблемы и выражением ее решения в виде алгоритма, который может быть выполнен вычислителем – человеком или машиной.

Программа Cubico создана для удовлетворения современных потребностей в области образования. Это пособие для легкого и увлекательного обучения детей младшего возраста посредством игры. Cubico поможет детям усвоить важные понятия алгоритмизации и развить вычислительное мышление.

Программирование, четвертый вид грамотности

Программирование – новый вид грамотности, наряду с умением читать, писать и считать. Как и при обучении иностранным языкам, занятия программированием в раннем детстве способствуют развитию алгоритмического мышления. Cubico позволяет детям получить подлинный творческий опыт программирования.

В ходе увлекательных занятий дети могут изучить основные понятия программирования и алгоритмизации. Вот некоторые ключевые понятия:

Последовательность

Цикл повторений

Условный оператор

Процедуры
Функции

Класс
Метод
наследования

События

Отладка

Упрощение
кода

Физическое программирование

Дополнительный раздел Cubico под названием «Inventor» («Изобретатель») содержит онлайн-среду, в которой ученик может управлять физическими устройствами путем программирования. Физические устройства включают переключатель, звуковые чипы, динамики, светодиоды, сервомотор и другие базовые компоненты. Эти занятия могут быть расширены до «обучения в процессе создания» («maker education»), которое является новейшим трендом в образовании. Физическое программирование поможет Вам стать настоящим творцом.

Парное программирование: общение и работа в команде

С помощью Cubico можно развивать умение работать в команде. Cubico подходит для парного и группового программирования, при котором над решением задачи работают двое или более учеников. Важно, чтобы дети учились ясно выражать свои идеи и обмениваться ими с другими учениками в процессе совместной работы. Cubico развивает у детей навыки коммуникации и сотрудничества.

Специально для детей младшего возраста: осязаемое и наглядное обучение

Cubico – это не просто онлайн-игра на основе приложения. И это не одна из обычных настольных игр, которые мы знаем. Это комплексное учебное пособие, специально разработанное для детей 4-6 лет с учетом их когнитивного, эмоционального и поведенческого развития.

Для детей этого возраста важны занятия с физическими объектами и наглядными пособиями. Начав с красочных иллюстрированных картонных карт на этапе планирования, дети затем создают программу, размещая блоки команд в определенной последовательности, которая может быть визуализирована на экране. Этот трехэтапный процесс позволяет детям получить интересный и полезный обучающий опыт.

Система обратной связи

После выполнения (или невыполнения) очередного задания ученики получают отзыв, который позволяет им улучшить свои результаты, и мотивирует продолжать занятия. Кроме того, Cubico регулярно информирует о результатах занятий родителей, чтобы они знали о прогрессе своего ребенка.

Фантастические приключения: контекстное обучение

Одна из уникальных особенностей Cubico – история фантастических приключений. История начинается в агентстве путешествий во времени. Озорное существо по имени «Баг» («Ошибка») ломает несколько пикселей в машине времени и убегает со сломанной машиной. Главный герой проходит через пространственно-временной портал, чтобы вернуть Бага и починить машину времени. Приключение начинается! Играя в Cubico, ученики путешествуют во времени вместе с главным героем и решают проблемы, связанные с поиском пути.

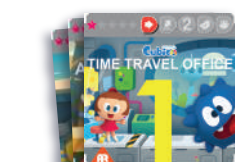
Cubico: программирование для детей

Cubico – это средство обучения, специально разработанное для детей младшего возраста. Cubico привлекает дошкольников осязаемыми физическими объектами и наглядными иллюстрациями.

Компоненты Cubico

Подготовка

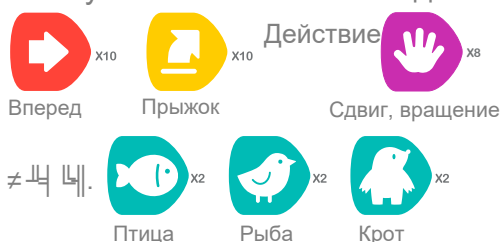
Карты этапов 20х



Маркер



Бумажные блоки команд



Кодирование

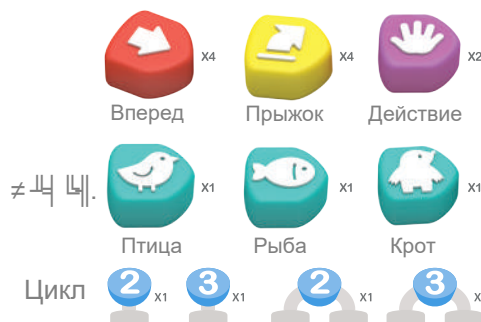
Панель для программирования



Чип кастомизации героев



Блоки команд



Как играть в Cubico – вкратце

Псевдокодирование: анализ задачи, абстракция

Во-первых, проанализируйте задачу: найти выход для главного героя. Ученики создают псевдокод, имитируя решение на карте этапа с помощью фигурок героев и карточек кода.

Кодирование: алгоритмическое мышление

Затем ученики помещают блоки команд (согласно псевдокоду) на панель с Bluetooth. С помощью блоков цикла код можно упростить.

Выполнение и отладка

Ученики выполняют созданный код на экране мобильного устройства. На устройствах с Android или iOS с установленным приложением Cubico они могут визуально проверить выполнение своего кода и выявить ошибки.

Отзыв о пройденном этапе

В конце каждого этапа ученики получают отзыв о своей работе. Интерактивные отзывы помогают им улучшать навыки программирования и переходить к более сложным этапам. Родители учеников также получают сведения о прогрессе своих детей.

Блоки команд

Каждый магнитный блок – это команда, которая управляет движениями главного героя. Располагая блоки команд в определенном порядке и выполняя их, дети приводят главного героя к его цели.

Каждый блок можно рассматривать как процедуру или функцию. Блоки цикла могут действовать как параметры для функции. Также в Cubico можно постепенно познакомить детей с такими понятиями программирования, как последовательность, цикл, условный оператор, наследование метода, событие и отладка.

Блоки команд управляют движениями главного героя. Он будет идти вперед, прыгать, плавать как рыба, летать как птица, двигаться под землей, как крот, вращать колесо лебедки или сдвигать препятствие, чтобы освободить себе путь.

Программные блоки		Описание	Понятия программирования
	Вперед	Блок «Вперед» заставит главного героя продвигнуться вперед на одну единицу пространства.	Процедура
	Прыжок	Блок «Прыжок» позволяет герою прыгнуть вверх или вниз на одну единицу пространства или перепрыгнуть через две единицы пространства.	Условный оператор, процедура
	Рыба	Если главный герой выйдет к океану или озеру, блок рыбы позволит ему проплыть под водой и достичь суши.	Условный оператор, класс, метод, наследование, цикл "while"
	Птица	Если главный герой выйдет к долине, он сможет лететь, как птица, пока не достигнет гнезда.	
	Крот	Главный герой сможет прорыть подземный ход, как крот, если путь будет заблокирован.	
	Действие: сдвиг или вращение	Используя блок действия, можно выбрать одно из двух действий: сдвиг или вращение колеса. Главный герой может сдвинуть препятствие или поднять его, вращая колесо лебедки, чтобы освободить себе путь. В каждом случае программист должен отодвинуть препятствие или покрутить колесо на экране, пока код выполняется.	Объявлять класс, метод, наследование, переменная
	Цикл	Если добавить этот блок, главный герой выполнит команду 2 раза.	Цикл/повторение, параметр
		Если добавить этот блок, главный герой выполнит команду 3 раза.	
		Если добавить этот блок, главный герой выполнит две последовательные команды 2 раза.	
		Если добавить этот блок, главный герой выполнит две последовательные команды 3 раза.	

Раздел	Этап	Тема	Блок	Понятия
Раздел 1	1	Агентство путешествий во времени	 Вперед Прыжок	Команды, последовательность, условный оператор
	2	Машина времени		
	3	Ледник из мороженого		
	4	Замерзший океан		
Раздел 2	5	Таинственные пирамиды	 Вперед Прыжок Цикл	Цикл, упрощение кода
	6	Парк Сфинкса		
	7	Храм Анубиса		
	8	Космическая станция		
	9	Блуждающая планета		
Раздел 3	10	Джунгли Амазонки	 Вперед Прыжок Цикл	Наследование метода
	11	Пещера в джунглях		
	12	Поселение майя		
	13	База пиратов		
	14	Корабль пиратов		
Раздел 4	15	Поселение пещерных людей	 Вперед Прыжок Действие Цикл	Объявление класса, событие
	16	Эра динозавров		
	17	Долина птеранодона		
Раздел 5	18	Мир стимпанка	 Вперед Прыжок Действие Цикл	Отладка
	19	Назад к машине времени		
	20	Цифровой мир		

План урока (пример)

Название	Эра динозавров (раздел 4: этап 16)
Карта задания	6 страниц
Схема этапа и решение	
План действий	1. Пусть ученики попытаются решить проблему, используя только блоки «Вперед», «Прыжок», «Цикл» и блоки животных. Спросите учеников, что потребуется для выполнения задания. 2. Рассмотрите блоки действия. Спросите учеников, помнят ли они, как их использовать. 3. Определите условие, которое требует наличия на карте блока действия. 4. Вместе найдите способ устранить препятствие. 5. Используйте изображение или короткий видеоролик, которые показывают, как использовать лебедку. 6. Найдите все возможные коды, которые приведут главного героя к цели. 7. Упростите код до пяти или менее блоков, используя блоки цикла.

Понятия программирования



Базовые алгоритмические понятия, такие как последовательность команд, условный оператор и оператор цикла, постоянно используются в программе Cubico, а на ее более поздних этапах вводятся более сложные понятия программирования.

Последовательность – важное понятие теории вычислений. Блоки команд выполняются в определенной последовательности, и ученики понимают, что их порядок важен.

Блоки **цикла** повторяют одну или две команды до трех раз. Выявление шаблонов – важный когнитивный навык для детей младшего возраста. Когда они выявляют повторяющиеся команды, они видят, что можно использовать блок цикла и тем самым упростить код.

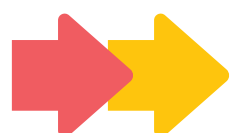
Условный оператор – еще одно важное понятие вычислительного мышления, которое лежит в основе каждого блока команд. Если тропинка немного повышается или понижается, следует использовать блок «Прыжок».

Если главному герою встречается океан, долина или тупик, он наследует **функции** рыбы, птицы или крота и действует, как они – таким же образом, как в **объектно-ориентированных языках** объект **наследует метод** или функцию, связанные с **классом**.

Используя блок действия, можно выбрать одно из двух действий: скольжение или вращение. Главный герой может сдвинуть препятствие или поднять его, вращая колесо лебедки, чтобы освободить себе путь. Это сходно с понятиями **определить переменную** или **объявить класс**. В каждом случае программист должен сдвинуть препятствие или вращать колесо на экране, пока код выполняется. Это можно рассматривать как **события**, инициированные пользователем.

Ученики должны помочь главному герою достичь цели, используя не более пяти блоков команд. Когда уровень сложности возрастает, они должны научиться упрощать код с помощью блоков цикла.

Иногда в код закрадывается ошибка, и тогда ему требуется отладка. Ученики анализируют код, чтобы найти и исправить ошибки. В процессе проб и ошибок ученики развивают свою способность к решению проблем и критическому аналитическому мышлению.



Парное/групповое программирование

Парное или групповое программирование – это часть гибкого метода разработки программного обеспечения. Два или три разработчика (ученика) работают на одном компьютере как одна команда. Один член команды выполняет роль «ведущего», а другой – роль «штурмана»; если в команде три человека, то один из них – «ведущий», а два других – «штурманы». В процессе обучения учителя нередко требуют, чтобы ученики периодически менялись ролями.

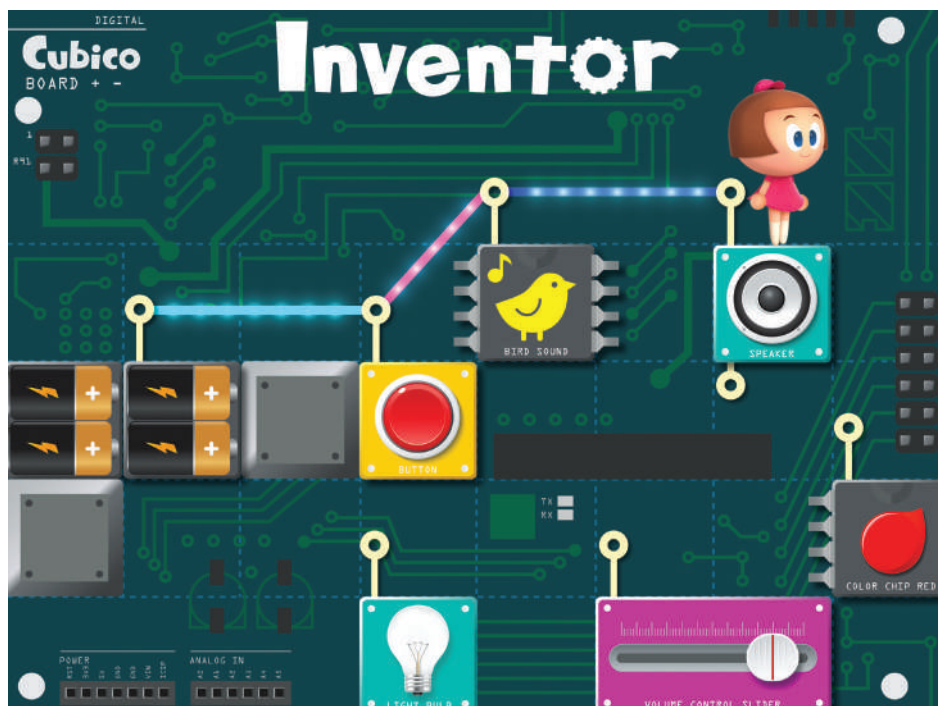
Ведущий пишет код, а штурман просматривает код и дает советы ведущему. Ведущий и штурман должны обсуждать код друг с другом. Иногда между членами команды могут возникать разногласия, и это тоже часть процесса программирования. Штурман должен высказывать свое мнение в конструктивной форме, а ведущий должен внимательно относиться к предложениям штурмана.

В Cubico штурман конструирует псевдокод, используя карту задания и бумажные программные блоки. Ведущий обсуждает кодирование со штурманом. В случае ошибки ведущий и штурман исправляют код вместе. А когда задача успешно решена, члены команды празднуют успех приветствием «дай пять!». На следующем этапе они меняются ролями.

В процессе парного или группового программирования ученики развивают навыки общения и постепенно понимают преимущества работы в команде. Работая вместе с другими учениками, дети учатся выражать свои мысли ясно и эффективно. Они могут делиться своими идеями и изучать проблемы с разных точек зрения.



Игры "Inventor"



Cubico предлагает дополнительную онлайн-игру «Inventor». Эта игра позволяет ученикам получить опыт управления такими устройствами, как переключатель, звуковые чипы, динамики, светодиоды и др., а также опыт физического программирования и опыт обучения в процессе создания («maker education»).

«Физическое программирование» («physical computing») означает построение интерактивных физических систем путем использования программного и аппаратного обеспечения, которое воспринимает реальный мир и реагирует на него. Ученики обучаются физическому программированию, соединяя блоки входных датчиков с блоками исполнительных устройств. Они могут создавать собственные проекты, используя блоки по своему усмотрению.

Игра «Inventor» – это первый шаг к реализации цели Cubico, которая заключается в том, чтобы вовлечь детей в создание новых полезных и творческих решений с использованием технологических методов.

Обзор блоков "Inventor"

	Название	Изображение	Описание
Питание	Батарейный блок		Блок питания. Батарейный блок обеспечивает работу других блоков.
Ввод	Кнопочный переключатель		Положение по умолчанию – «выключено». Когда кнопка нажата, плата включена. Когда кнопка отпущена, плата выключена.
	Ползунковый переключатель		Ползунковый переключатель (регулятор), как и другие переключатели, включает и выключает подключаемые блоки. Также он может регулировать яркость света, громкость звука, скорость пропеллера и серводвигателя. Он занимает до двух единиц пространства.



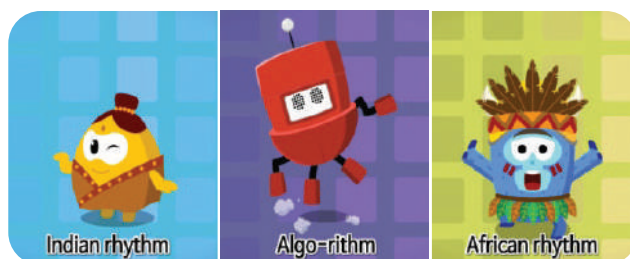
Обзор блоков "Inventor"

	Название	Изображение	Описание
Ввод	Движковый переключатель		Включение/выключение одним кликом.
Ввод	Чип щебета птиц		При соединении чипа с блоком питания и динамиком раздается щебет птиц.
	Чип колокольчика		При соединении чипа с блоком питания и динамиком раздается звон колокольчика.
	Чип рева слона		При соединении чипа с блоком питания и динамиком раздается рев слона.
	Чип красного света		При соединении чипа с блоком питания и светодиодным индикатором, цвет индикатора становится красным.
	Чип зеленого света		При соединении чипа с блоком питания и светодиодным индикатором, цвет индикатора становится зеленым.
	Чип синего света		При соединении чипа с блоком питания и светодиодным индикатором, цвет индикатора становится синим.
Вывод	Динамик		При соединении с блоком питания и звуковыми чипами динамик издает соответствующий звук. Громкость регулируется с помощью ползункового регулятора.
	Светодиод		Красно-зелено-синий светодиодный индикатор меняет свой цвет при соединении с цветовыми чипами. Яркость регулируется с помощью ползункового регулятора.
	Пропеллер		Пропеллер – как двигатель постоянного тока.
	Серводвигатель		Серводвигатель поворачивается на 90 градусов в любом направлении, т.е. в общей сложности на 180 градусов. Его скорость можно изменять с помощью ползункового регулятора.

Обучающие песенки Cubico для детей



Пение – это веселый и эффективный способ обучения. В классе используются песенки, посвященные разным понятиям программирования. Веселые и простые песенки Cubico помогают улучшить знание терминов и понятий компьютерных наук. Дети обучаются через повторение. Повторяющиеся стихотворные строки и живой песенный ритм способствуют запоминанию и делают обучение более увлекательным. Песни Cubico посвящены следующим темам:



О компьютерных науках в целом:

Computers are
Everywhere

(Вездесущность
компьютеров)

Software,
Hardware,
Anywhere,
Everywhere

(Программное и
аппаратное
обеспечение)

Zero
One
Song

(Двоичный код и
кодирование)

Python is
not a reptile

(Языки
программирования)

О программировании:

Coding
Song

(Понятие
кодирования)

Bits
and
Bytes

(Данные, размер
данных)

Rhythm,
Rhythm,
Algorithm

(Алгоритмы)

Do It Again
Song

(Итерация, цикл)

If, Then
Song

(Условные
операторы)

Catch a Bug

(Ошибки, отладка)

Занятия в оффлайне

Занятия в оффлайне – это учебные занятия без устройств с дисплеями, в ходе которых изучаются базовые понятия компьютерных наук и вычислительного мышления. Это могут быть настольные игры, пазлы, сравнение изображений, расшифровка кодов, поиск пути на карте и многое другое.

Эти занятия, весьма разноплановые с педагогической точки зрения, помогают вовлечь учеников в процесс решения проблем и способствуют развитию логического и аналитического мышления. Программа Cubico предлагает увлекательные занятия в оффлайне, специально предназначенные для детей, которые еще не умеют читать.

Словарь терминов Cubico

(Примечание: Данный глоссарий составлен на основе стандартов CSTA K-12 Computer Science Standards [Interim])

абстракция (процесс):

Процесс уменьшения сложности путем фокусирования на основной идее.

Скрытие несущественных деталей и выявление существенных и полезных деталей уменьшает сложность и позволяет сфокусироваться на проблеме. В начальной школе абстракция заключается в том, чтобы скрыть ненужные детали и тем самым облегчить рассмотрение проблемы.

алгоритм:

Поэтапный процесс выполнения задачи.

аппаратное обеспечение:

Физические составляющие вычислительной системы, компьютера или вычислительного устройства. [MA-DLCS]

вводные данные:

Сигналы или инструкции, посылаемые компьютеру. [Techopedia]

выполнение:

Процесс выполнения команды или набора команд.

выполнять:

Выполнять команду или набор команд (программу, приложение и т.д.).

вычислительное мышление:

Мыслительный процесс, обеспечивающий формулирование проблем и представление их решений в форме, которая может быть воспринята средством обработки информации (например, компьютером) [Cuny, Snyder, & Wing, 2010]

данные:

Информация, которую собирают и используют для справки или анализа.

Данные могут быть цифровыми или нецифровыми и могут иметь различную форму, такую как числа, текст, поднятие рук, изображения, звуки или видео. [CAS-Prim, TechTerms]

декомпозиция:

Разложение проблемы или системы на составные части. [MA-DLCS]

исполнимый файл:

Двоичный файл, содержащий готовую к выполнению программу на языке программирования. [FOLDOC]

итеративный:

Включающий повторение процесса с целью приближения к поставленной цели или результату. [MA-DLCS]

класс:

Прообраз объекта.

Можно представить себе класс как концепцию, а объект – как воплощение этой концепции.

код:

Любой набор инструкций, выраженный на языке программирования. [MA-DLCS]

команда:

Указание программе осуществить определенное действие для решения определенной задачи. [Techopedia]

компьютер:

Машина или устройство, которое осуществляет процессы, вычисления и операции на основе инструкций аппаратно и программно реализуемых программ. [Techopedia]

компьютерные науки:

Изучение компьютеров и алгоритмических процессов, их принципов, разработки, реализации и воздействия на общество.

контроль:

(вообще) Способность определять способ действий.

(в программировании) Определение того, какие действия должны быть совершены и в каком порядке, с помощью элементов программного кода.

Словарь терминов Cubico

метод:

(в объектно-ориентированном программировании) Процедура, принадлежащая к какому-то классу. Метод определяет поведение объектов, созданных на основе этого класса. Иными словами, метод – это действие, которое объект способен выполнять. [MA-DLCS, CSTA]

моделирование:

Имитация работы реального процесса или системы во времени. [MA-DLCS]

моделировать:

Имитировать работу реального процесса или системы во времени.

объектно-ориентированное программирование (ООП):

Подход к решению проблем, при котором все вычисления выполняются с использованием объектов.

операция:

Действие, вызванное одной командой, которое изменяет состояние данных. [Dictionary.com]

отладка:

Процесс поиска и исправления ошибок в программах. [MA-DLCS]

ошибка:

Ошибка в коде программы. Ошибка может вызвать непредусмотренное поведение программы или неожиданное прекращение ее работы. [TechTerms] Процесс исправления ошибок называется отладкой.

параметр:

Особый вид переменной, используемый в процедуре, чтобы сослаться на один из элементов данных, предоставленных как входные данные для процедуры. Эти элементы данных называются аргументами. В определении подпрограммы обычно содержится упорядоченный список параметров; таким образом, при каждом вызове подпрограммы ее аргументы для этого вызова могут быть присвоены соответствующим параметрам. [MADLCS]

парное программирование:

Способ программирования, при котором два разработчика (или ученика) вместе работают на одном компьютере. [TechTarget] Один из них выполняет роль «ведущего», а другой – роль «штурмана». В процессе обучения учителя нередко требуют, чтобы ученики периодически менялись ролями.

переменная:

Символическое название, используемое для отслеживания значения, которое может меняться, пока выполняется программа. Переменные используются не только для чисел. Они также могут относиться к тексту, в том числе к целым предложениям («строкам»), или к логическим значениям «true» («истинно») и «false» («ложно»). Переменная содержит определенных тип данных и связана с определенным местом хранения данных; ее значение, как правило, меняется в течение выполнения программы.

Примечание:

Это определение отличается от определения, используемого в математике.

подпрограмма:

Вызываемая единица кода, вид процедуры.

программа:

Набор инструкций, которые компьютер выполняет для достижения определенной цели. [MA-DLCS]

программировать:

Создавать программу путем программирования.

программирование:

Искусство анализа проблем и разработки, написания, тестирования и поддержки программ для их решения. [MA-DLCS]

программное обеспечение:

Программы, которые выполняются в компьютерной системе, компьютере или ином вычислительном устройстве.

Словарь терминов Cubico

процедура:

Независимый модуль кода, который выполняет конкретную задачу и на который имеются ссылки внутри более крупного массива исходного кода. Этот элемент кода также может называться функцией или подпрограммой.

Фундаментальная роль процедуры заключается в том, чтобы обеспечивать единую отправную точку для некоторой небольшой цели или задачи, которую разработчик или программист может запускать, вызывая соответствующую процедуру. Процедура может также называться функцией, программой, подпрограммой или методом. [Techopedia]

псевдокод:

Подробное, но удобочитаемое описание того, что должна сделать компьютерная программа или алгоритм, написанное на формализованном естественном языке, а не на языке программирования.

событие:

Любое различимое событие, которое имеет значение для аппаратного или программного обеспечения системы. События, инициированные пользователем, включают нажатия клавиш и щелчки мышью; события, генерируемые системой, включают загрузку программ и ошибки.

[TechTarget]

условный оператор:

Элемент языка программирования, который выполняет разные вычисления или действия в зависимости от того, выполняется или нет определенное программистом логическое условие.

функция:

Вид процедуры или программы, т.е. набора инструкций, составленного для достижения определенного результата. Обычно функция требует ввода некоторых данных (называемых аргументами) и дает некоторые результаты.

Примечание: Это определение отличается от определения, используемого в математике.

цикл:

Структура программирования, которая повторяет последовательность инструкций, пока выполняется определенное условие. [TechTerms]

циклическое выполнение:

Повтор, использование оператора цикла

CAS-Prim:	Computing At School. Computing in the national curriculum: A guide for primary teachers (http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf)
CSTA:	CSTA K-12 Computer Science Standards (2011) https://csta.acm.org/Curriculum/sub/K12Standards.html
FOLDOC:	Free On-Line Dictionary of Computing. (http://foldoc.org/)
MA-DLCS:	Massachusetts Digital Literacy and Computer Science (DL&CS) Standards, Glossary (Draft, December 2015)
NIST/DADS:	National Institute of Science and Technology Dictionary of Algorithms and Data Structures. (https://xlinux.nist.gov/dads/)
Techopedia:	Techopedia. (https://www.techopedia.com/dictionary)
TechTarget:	TechTarget Network. (http://www.techtarget.com/network)
TechTerms:	Tech Terms Computer Dictionary. (http://www.techterms.com)