ОГЛАВЛЕНИЕ

[ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 6](#_Toc158277313)

[ЭТАПЫ РЕШЕНИЯ ЗАДАЧ НА КОМПЬЮТЕРЕ. 8](#_Toc158277314)

[КОНСПЕКТ ОТКРЫТОГО УРОКА ПО УЧЕБНОЙ ДИСЦИПЛИНЕ «ИНФОРМАТИКА»: 14](#_Toc158277316)

[ЗАКЛЮЧЕНИЕ 23](#_Toc158277317)

[ИСТОЧНИКИ 25](#_Toc158277318)

[ПРИЛОЖЕНИЯ 26](#_Toc158277319)

# ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой информации главным образом с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности. Информатику в узком смысле можно представить как состоящую из трех взаимосвязанных частей — технических средств (hardware), программных средств (software), алгоритмических средств (brainware). Возможности компьютера как технической основы обработки данных связаны с используемым программным обеспечением (программами).

Программирование можно рассматривать как искусство, науку, ремесло. Программирование – это искусство получения ответов от машины. Для этого в узком смысле нужно составить специальный код для технического устройства, в широком – разработать программы на языках программирования, т.е. не просто составить код, а выполнить интеллектуальную работу по составлению высокоразумных программ для решения различных задач во всех сферах человеческой деятельности.

Курс учебной дисциплины Информатика  в образовательных учреждениях СПО содержит один из разделов «Алгоритмы и программирование». В данном разделе рассматривается содержание основных понятий программирования, история его развития, основные элементы и языки программирования. Принципы структурного программирования и начала программирования в среде Basic. Дается материал о программном обеспечении для вычислительной техники и автоматизированных систем.

Тема «Алгоритмы и программирование» направлена на понимание сути алгоритмов, их свойств, способов описания, так как эта тема развивает: ясность и четкость мышления; способность предельно уточнять предмет мысли; внимательность, аккуратность, обстоятельность, убедительность в суждениях; умение абстрагироваться от конкретного содержания и сосредоточиться на структуре своей мысли.

Ввиду важности темы в курсе информатики, мною были разработаны данные методические рекомендации.

# ЭТАПЫ РЕШЕНИЯ ЗАДАЧ НА КОМПЬЮТЕРЕ.

# ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ.

Компьютер предназначен для решения разнообразных задач: научно-технических, инженерных, разработки системного программного обеспечения, обучения, управления производственными процессами и т.д. В процессе подготовки и решения на компьютере разного рода задач можно выделить следующие этапы:

1. *Постановка задачи –* формулируется цель решения задачи, подробно описывается ее содержание; проводится анализ условий, при которых решается поставленная задача, выявляется область определения входных параметров задачи.

2. *Формальное построение модели задачи –* предполагает построение модели с характеристиками, адекватными оригиналу, на основе какого-либо его физического или информационного принципа; анализируется характер и сущность величин, используемых в задаче.

3. *Построение математической модели задачи —* характеризуется математической формализацией задачи, при которой существующие взаимосвязи между величинами выражаются с помощью математических соотношений. Как правило, математическая модель строится с определенной точностью, допущениями и ограничениями.

4. *Выбор и обоснование метода решения —* модель решения задачи реализуется на основе конкретных приемов и методов решения. В большинстве случаев математическое описание задачи трудно перевести на машинный язык. Выбор и использование метода решения позволяет свести решение задачи к конкретному набору машинных команд. При обосновании метода решения рассматриваются вопросы влияния различных факторов и условий на конечный результат, в том числе на точность вычислений, время решения задачи на компьютере, требуемый объем памяти и др.

5. *Построение алгоритма —* на данном этапе составляется алгоритм решения задачи, в соответствии с выбранным методом решения. Процесс обработки данных разбивается на отдельные относительно самостоятельные блоки, определяется последовательность выполнения этих блоков.

6. *Составление программы —* алгоритм решения переводится на конкретный язык программирования.

7. *Отладка программы —* процесс устранения синтаксических и логических ошибок в программе. В процессе трансляции программы с помощью синтаксического и семантического контроля выявляются недопустимые конструкции и символы (или сочетания символов) для данного языка программирования. Компьютер выдает сообщение об ошибках в форме, соответствующей этому языку. Затем проверяется логика работы программы в процессе ее выполнения с конкретными исходными данными. Для этого используются специальные методы. Например, в программе выбираются контрольные точки, для них подбираются тестирующие примеры и вручную находятся значения в этих точках, которые затем и сверяются со значениями, получаемыми компьютером на этапе отладки. Кроме того, используются отладчики, выполняющие специальные действия на этапе отладки, такие как удаление, замена или вставка отдельных операторов или целых фрагментов программы, вывод промежуточных результатов, изменение значений заданных переменных и др.

8. *Решение задачи на компьютере и анализ результатов.* Теперь программу можно использовать для решения поставленной задачи. Первоначально выполняется многократное решение задачи на компьютере для различных наборов исходных данных. Получаемые результаты анализируются человеком, поставившим задачу.

Но в некоторых случаях отдельные этапы могут отсутствовать. Например, проектирование программного обеспечения не требует построения математической модели.

Программирование это процесс подготовки задач для решения их на ЭВМ, состоящий из следующих этапов:

* алгоритмическое описание задачи;
* запись алгоритма на языке программирования (составление программы);
* трансляция программы с языка программирования на машинный язык (в виде последовательности команд, реализация которых техническими средствами ЭВМ и есть процесс решения задачи).

Основными технологиями разработки программного обеспечения являются:

1. Структурное программирование;

2. Модульное программирование.

*Структурное программирование.* Сутью структурного программирования является возможность разбиения программы на составляющие элементы.

Идеи структурного программирования появились в начале 70-годов 20 века в компании IBM, в их разработке участвовали известные ученые Э. Дейкстра, Х. Милс, Э. Кнут, С. Хоор.

Распространены две методики (стратегии) разработки программ, относящиеся к структурному программированию: программирование "сверху вниз" и программирование "снизу вверх".

Программирование "сверху вниз", или нисходящее программирование – это методика разработки программ, при которой разработка начинается с определения целей решения проблемы, после чего идет последовательная детализация, заканчивающаяся детальной программой.

При нисходящем проектировании задача анализируется с целью определения возможности разбиения ее на ряд подзадач. Затем каждая из полученных подзадач также анализируется для возможного разбиения на подзадачи. Процесс заканчивается, когда подзадачу невозможно или нецелесообразно далее разбивать на подзадачи.

В данном случае программа конструируется иерархически - сверху вниз: от главной программы к подпрограммам самого нижнего уровня, причем на каждом уровне используются только простые последовательности инструкций, циклы и условные разветвления.

Противоположной является методика программирования «снизу вверх». Программирование «снизу вверх», или восходящее программирование – это методика разработки программ, начинающаяся с разработки подпрограмм (процедур, функций), в то время когда проработка общей схемы не закончилась. Является противоположной методике программирования «сверху вниз».

Такая методика является менее предпочтительной по сравнению с нисходящим программированием, так как часто приводит к нежелательным результатам, переделкам и увеличению времени разработки.

Достоинства структурного программирования:

* повышается надежность программ (благодаря хорошему структурированию при проектировании, программа легко поддается тестированию и не создает проблем при отладке);
* повышается эффективность программ (структурирование программы позволяет легко находить и корректировать ошибки, а отдельные подпрограммы можно переделывать (модифицировать) независимо от других);
* уменьшается время и стоимость программной разработки;
* улучшается читабельность программ.

*Модульное программирование* – это организация программы как совокупности небольших независимых блоков (модулей), структура и поведение которых подчиняется определенным заранее правилам.

Модулем (в модульном программировании) называется множество взаимосвязанных подпрограмм (процедур) вместе с данными, которые эти подпрограммы обрабатывают. Единственное различие между маленькой модульной программой и большой модульной программой заключается в количестве модулей, из которых они состоят. Поскольку модули не зависят друг от друга, создание большой модульной программы не очень отличается от написания многих маленьких независимых модульных программ, хотя взаимодействие между модулями может быть весьма сложным. Работа над большой цельной программой напоминает одновременную работу со множеством маленьких взаимосвязанных программ. Кроме того, модульность позволяет применить командный способ программирования, при котором несколько программистов работают независимо друг от друга, а затем объединяют свои модули в одну программу.

Модульное программирование предназначено для разработки больших программ. Разработкой больших программ занимается коллектив программистов. Каждому программисту поручается разработка некоторой самостоятельной части программы. И он в таком случае отвечает за конструирование всех необходимых процедур и данных для этих процедур. Сокрытие данных (запрет доступа к данным из-за пределов модуля) предотвращает их случайное изменение и соответственно нарушение работы программы. Для взаимодействия отдельных частей (модулей) программы коллективу программистов необходимо продумать только интерфейс (взаимодействие) сконструированных модулей в основной программе.

Отладка большой программы может оказаться практически невыполнимой задачей.

Большое преимущество модульного подхода заключается в том, что задача отладки большой программы сводится к отладке множества маленьких подпрограмм. Начиная кодировать модуль, вы должны быть уверены, что все остальные модули закодированы правильно. Следовательно, закончив программирование модуля, вы должны внимательно проверить его, как отдельно, так и вместе с другими модулями, вызывая его с фактическими аргументами, тщательно подобранными для выявления всех возможных недостатков. Если это тестирование проведено подобающим образом, можно быть уверенным, что любая обнаруженная ошибка содержится только в модуле, который кодировался последним. Модульность позволяет изолировать ошибки.

Модульный подход не только позволяет программистам справляться со сложностями, возникающими при решении задачи, но и помогает читателям программы понять, как она работает. Модульную программу легко отследить, поскольку читатель хорошо представляет себе, что происходит, не вдаваясь в детали кода. Для того чтобы разобраться в хорошо написанной функции, достаточно лишь прочитать ее имя, начальные комментарии и имена функций, которые вызываются внутри нее.

Другое преимущество модульного подхода проявляется в выявлении вычислений, которые в программе выполняются несколько раз. Такие вычисления следует реализовывать в виде функций. В этом случае код, предназначенный для таких вычислений, в программе встречается только один раз, повышая ее читабельность и модифицируемость.

# КОНСПЕКТ ОТКРЫТОГО УРОКА ПО УЧЕБНОЙ ДИСЦИПЛИНЕ «ИНФОРМАТИКА»:

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 17. РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ РЕШЕНИЯ ТИПОВЫХ ЗАДАЧ БАЗОВОГО УРОВНЯ**

**ПЛАН УРОКА №*14***

|  |  |
| --- | --- |
| Группа | Дата |
| СД-11 | 5.02.24 |

Дисциплина *Информатика*

Преподаватель *Мирохина Е.С.*

Тема урока *Практическое занятие № 17. Разработка и программная реализация алгоритмов решения типовых задач базового уровня.*

Тип урока *Совершенствование знаний, умений, навыков*

Вид урока *Практическое занятие*

Учебные: *формирование знаний об алгоритмах, формирование умения разработки и программной реализации алгоритмов;*

*формирование знаний о среде программирования QBasic, основных операторах и функциях языка программирования, умения решать задачи с использованием языка программирования.*

Воспитательные: *формирование интереса к информатике, личной ответственности за результаты своей работы на компьютере.*

Развивающие: *развитие критического и логико-алгоритмического мышления, которое проявляется в умении строить логические утверждения о свойствах данных, в умении мыслить индуктивно и дедуктивно, в умении формализовывать задачу, вплоть до её записи на алгоритмическом языке.*

**Межпредметные связи**

Обеспечивающие *Русский язык, математика, основы безопасности жизнедеятельности, иностранный язык*

Обеспечиваемые *Информационные технологии в профессиональной деятельности*

**Обеспечение занятия**

Наглядные пособия *Мультимедийная презентация*

Раздаточные материалы – *Таблица основных операторов и функций QBasic, инструкционные карты*

Технические средства обучения *ЭВМ*

Литература основная *Трофимов В.В., Информатика*

Дополнительная *Гаврилов М.В., Информатика*

**Содержание занятия**

1. Организационный момент *Приветствие, перекличка*
2. Проверка домашнего задания
3. *Фронтальный опрос*

* *понятие алгоритма*
* *свойства алгоритмов*
* *способы записи (представления) алгоритмов*
* *виды алгоритмов (с примерами)*
* *линейный алгоритм*
* *алгоритм ветвления*
* *циклический алгоритм*

1. *Изобразить на доске основные блоки блок-схемы*
2. Краткое объяснение преподавателем нового материала, сообщения обучающихся, знакомство со средой программирования QBasic.
3. Решение простейшей задачи (в качестве примера) с использованием языка программирования Basic
4. Выполнение заданий *практического занятия № 12* по инструкционным картам
5. *Работа в парах (резервное время)*
6. Домашнее задание *отчет о практическом занятии № 17, подготовка к защите*
7. Оценки за урок

ХОД ЗАНЯТИЯ

Программирование – это процесс разработки программы (алгоритма) на определённом языке программирования.

Знакомство пользователя с компьютером начинается с операционной системы. Осваивая ее команды шаг за шагом, узнавая новые и новые программы, пользователь неизменно подходит к той черте, за которой ему хочется не только работать с готовыми программами, но и пробовать свои силы в создании новой, пусть и очень простой, но своей программы. Для таких целей фирма Microsoft разработала очень удобный интерефейс программиста – интерпретатор QBasic.

Язык Basic создан в 60 – х годах прошлого столетия как язык для начального обучения программированию в диалоге с ЭВМ.Этот язык создавался американскими программистами, и поэтому в основу его была положена лексика английского языка. BASIC – Beginner’s All – purpose Symbolic Instruction Code – язык символьных конструкций для начинающих. Программа на Бейсике - это пронумерованная последовательность строк, за номером расположена команда. Команда указывает компьютеру, что по ней надо сделать и над какими данными. Набор команд строго ограничен.

Основные элементы – константы (const), переменные, выражения, функции и массивы.

Имена и значения строятся из символов или литеров. Алфавит Бейсика составляет совокупность литер и около 200 служебных слов.

Чтобы запустить QBasic, откройте папки *Для студентов/Первый курс/ QBasic*. После запуска программы вы увидите приглашение начать работу. Для начала работы нажмите клавишу на клавиатуре ESC, разверните окно программы на весь экран нажатием комбинации клавиш *Alt+Enter*.

**Алфавит языка Basic**.

* буквы латинского алфавита (A-Z)
* арабские цифры (0, 1, 2,…, 9)
* специальные символы (.,:?% $ ” ’ + - / \* ^)
* русские буквы используются только в пояснениях, комментариях.

**Некоторые операторы языка Basic.**

|  |  |
| --- | --- |
| *Оператор* | *Назначение* |
| **REM** | оператор комментария. |
| **CLS** | очистить экран. Вся информация, которая была на экране, стирается. |
| **PRINT** | оператор вывода |
| **INPUT** | оператор ввода |
| **LET** | оператор присваивания значения переменной |
| **IF…THEN…ELSE** | выполняет блок операторов в зависимости от указанных условий |
| **GOTO строка** | выполняет переход в указанную строку |
| **END** | конец программы |
| **COLOR m, n** | Установка цвета, где m- цвет фона, n-цвет букв. |
| **LOCATE m, n -** | Вывод текста в центре экрана в строке c номером m и столбце с номером n |
| **CIRCLE (X,Y),R,C** | Используется для построения окружности.  (X,Y), - координаты центра, R - радиус,C - цвет |
| **CIRCLE (X,Y),R,C**  **PAINT(X,Y),С1,C2** | PAINT – оператор закрашивания, (X,Y) – координаты точки внутри закрашиваемой области, С1 – цвет закрашивания, С2- цвет линии границы |
| **LINE** | используется для построения линии и прямоугольника |
| **PAINT** | используется для закрашивания замкнутого контура цветом |
| **SCREEN N** | Этот оператор очищает экран и задает графический режим, где N определяет тип графического режима  N=1,7,8,9,12 |
| **WINDOW** | предназначен для задания на экране удобной для задачи на построение графика функций декартовой системы координат |
| **PSET(X,Y),C** | Где X,Y – координаты точки, C – цвет точки |
| **LINE(X1,Y1)-(X2,Y2),C** | (X1,Y1) и (X2,Y2) – координаты начала и конца отрезка, C – цвет |
| **LINE(X1,Y1)-(X2,Y2),C, В** | Рисование прямоугольной рамки , где  (X1,Y1) и (X2,Y2) – координаты концов левой диагонали, C – цвет |
| **LINE(X1,Y1)-(X2,Y2),C, ВF** | Рисование закрашенного прямоугольника. Здесь (X1,Y1) и (X2,Y2) – координаты концов левой диагонали, C – цвет |

**Основные математические функции языка QBasic:**

|  |  |
| --- | --- |
| Корень | SQR(X) |
| Модуль числа | ABS(X) |
| Синус | SIN(X) |
| Косинус | COS(X) |
| Тангенс | TAN(X) |
| Целая часть числа | INT(X) |
| Натуральный логарифм | LOG(X) |

**Арифметические действия**

**+** сложение

**-**  вычитание

**\*** умножение

**/**  деление

**^** возведение в степень

() скобки

**Номера цветов**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Цвет | № | Цвет |
| 0 | черный | 8 | Темно-серый |
| 1 | синий | 9 | ярко-голубой |
| 2 | зеленый | 10 | ярко-зеленый |
| 3 | голубой | 11 | ярко-бирюзовый |
| 4 | красный | 12 | ярко-красный |
| 5 | розовый | 13 | ярко-розовый |
| 6 | коричневый | 14 | желтый |
| 7 | серый | 15 | ярко-белый |

*Решение простейшей задачи (в качестве примера) с использованием языка программирования Basic.*

На первом занятии во время знакомства с алгоритмами, мы начали учиться строить блок-схемы, я говорила вам, что правильная блок-схема – это на 90 процентов составленная программа. В тетради выбираем любую задачу на линейный алгоритм с блок-схемой, составляем к ней программу. Затем обучающиеся набирают ее на компьютере, запускают *Shift+F5*, осуществляют контрольные вычисления.

Выполнение заданий *практического занятия № 17* по инструкционным картам

Практическое занятие № 17.

**Разработка и программная реализация алгоритмов решения типовых задач базового уровня.**

*Цель занятия:*

* формирование знаний об алгоритмах, формирование умения разработки и программной реализации алгоритмов;
* формирование знаний о среде программирования QBasic, основных операторах и функциях языка программирования, умения составлять простейшие программы.

Задание 1. Составьте блок-схему и программу для вычисления площади прямоугольного треугольника со сторонами *а* и *b*.

REM Площадь прямоугольного треугольника  
INPUT "Введите сторону а", а  
INPUT "Введите сторону b", b  
s = 0.5 \*a \* b  
PRINT "Площадь равна: ", s  
END

*Контрольный счет:*

Введите сторону а 5

Введите сторону b 8

Площадь равна: ­­\_\_\_\_\_\_\_

Задание 2. Составьте блок-схему и программу для вычисления длины окружности и площади круга по данному радиусу.

REM Вычисление длины окружности и площади круга  
INPUT "Введите радиус ", r  
PI = 3.14  
l = 2 \* PI \* r  
s= PI \* r \* r  
PRINT "Длина окружности равна: ", l  
PRINT "Площадь равна: ", s  
END

Задание 3. Составить блок-схему и программу для нахождения корней квадратного уравнения.

CLS  
INPUT "A =", А  
INPUT "B=", B  
INPUT "C=", C  
D = B^2 - 4 \* A \* C  
IF D > = 0 THEN

X1 = (-b + SQR\*(D)) / a\*2

X2 = (-b - SQR\*(D)) / a\*2

PRINT "X1=", X1  
PRINT "X2=", X2

ELSE PRINT "deistvitelnix korney net"  
END IF

END

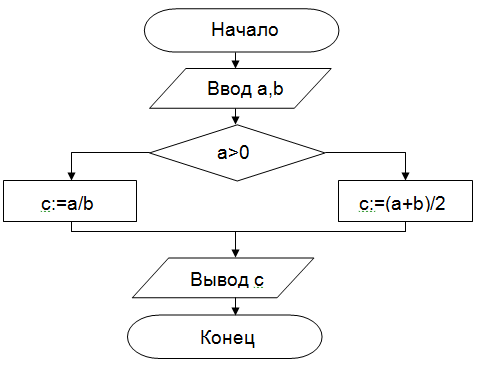
Задание 4. Составьте программу, складывающую два числа.

REM Сумма двух чисел  
a = 5  
b = 6  
c = a + b  
PRINT "Результат: ", c  
END

Задание 5. Вычислить значение выражения  . (Составьте программу, запустите, контрольный счет: a=6, b=8).

Задание 6.

Определить среднее арифметическое двух чисел, если *a* положительное и частное *(a/b)* в противном случае.

 *Блок-схема Программа*

Задание 7. Какой четверти принадлежит точка (х,у).

INPUT X,Y

IF X>0 AND Y>0 THEN ?”1”

IF X<0 AND Y>0 THEN ?”2”

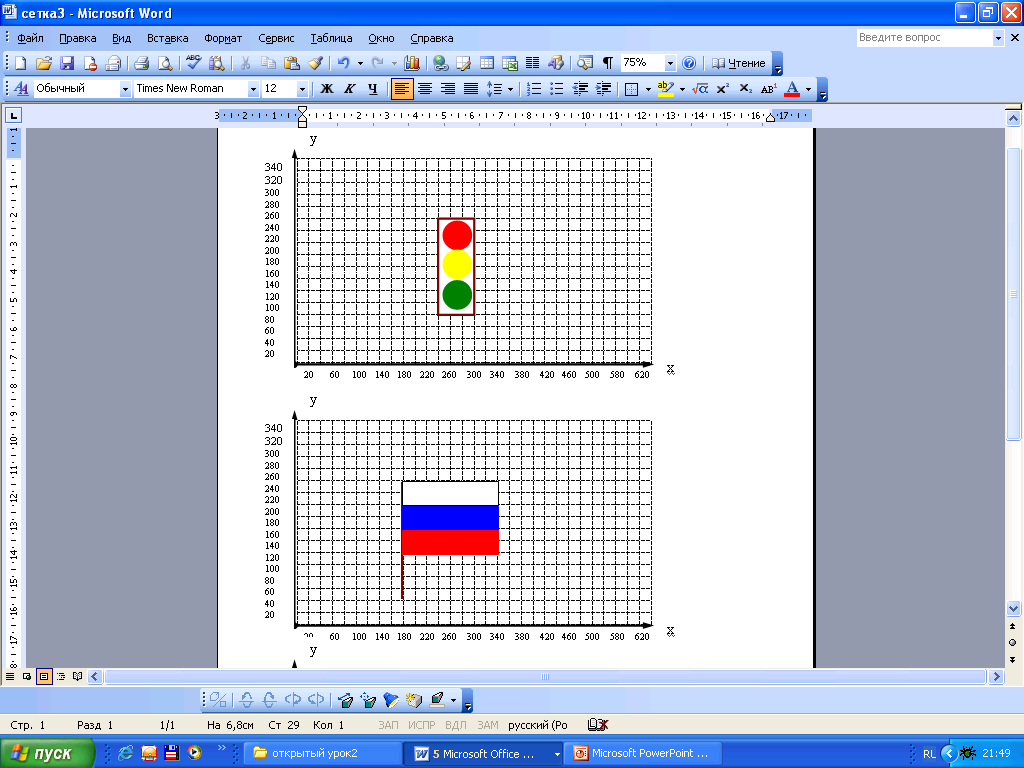
IF X<0 AND Y<0 THEN ?”3”

IF X>0 AND Y<0 THEN ?”4”

Задание 8. Что будет на экране компьютера в результате выполнения программы?

1. SCREEN 9
2. WINDOW (0,0)-(640,350)
3. CIRCLE (300, 200), 120, 10
4. CIRCLE (260,240),40,3,,, 0.5
5. PAINT (270, 250), 3
6. LINE (310, 240) – (390, 240), 3
7. LINE (300,200)-(280,160), 10
8. LINE (280,160)-(320,160), 10
9. LINE (320,160)-(300,200), 10
10. LINE (260,140)-(340,140), 4
11. LINE (260,0) – (340, 100), 10, BF

Задание 9. По рисунку составьте программу сначала в тетради, затем наберите ее на компьютере, сверьте изображения



**Флаг.**

SCREEN 9

WINDOW (0,0)-(640,350)

LINE (180,200)-(340,240), 7, BF

LINE (180,160)-(340,200), 1, BF

LINE (180,120)-(340,160), 4, BF

LINE (180,40)-(180,240), 6

***Работа в парах.*** *Задание с сюрпризом (обучающиеся объединяются в пары и набирают программу, данную преподавателем).*

*I вариант*

SCREEN 12  
CLS  
LINE (200, 200)-(350, 400), , B  
LINE (200, 200)-(275, 100)  
LINE (275, 100)-(350, 200)  
CIRCLE (275, 160), 20  
PAINT (275, 160)  
LINE (250, 250)-(300, 300), , B  
LINE (275, 250)-(275, 300)  
LINE (250, 270)-(300, 270)  
LINE (350, 400)-(355, 210), , B  
LINE (350, 210)-(370, 200), , B  
LINE (350, 400)-(400, 400)  
LINE -(400, 380)  
LINE -(380, 380)  
LINE -(380, 360)  
LINE -(355, 360)  
LINE (325, 165)-(325, 100)  
LINE -(310, 100)  
LINE -(310, 145)

*II вариант*

|  |
| --- |
| Screen12  CLS  CIRCLE(240,240),25  CIRCLE(360,240),25  LINE(270,240)-(330,240)  LINE(180,240)-(180,180)  LINE(420,240)-(420,180)  LINE(180,240)-(210,240)  LINE(390,240)-(420,240)  LINE(180,180)-(240,180)  LINE(380,180)-(420,180)  CIRCLE(300,180),80,,0,3.14  CIRCLE(240,240),30,,0,3.14  CIRCLE(360,240),30,,0,3.14 |

*III вариант*

SCREEN12

LINE(50,100)-(100,50)

LINE(100,50)-(150,100)

LINE(50,100)-(150,100)

LINE(50,100)-(50,150)

LINE(150,100)-(150,150)

LINE(50,150)-(150,150)

LINE(60,110)-(60,140)

LINE(60,110)-(80,110)

LINE(80,110)-(80,140)

LINE(60,140)-(80,140)

LINE(140,110)-(140,140)

LINE(140,140)-(120,140)

LINE(120,140)-(120,110)

LINE(120,110)-(140,110)

LINE(60,120)-(80,120)

LINE(70,110)-(70,140)

LINE(140,120)-(120,120)

LINE(130,110)-(130,140)

CIRCLE(100,80),10

LINE(90,80)-(110,80)

LINE(100,70)-(100,90)

*IV вариант*

SCREEN12

FOR X=0 TO 640 STEP 1

CLS

CIRCLE (X, 240), 25

CIRCLE (X+100, 240), 25

LINE (X+30,240)-(X+70,240)

LINE (X-50,240)-(X-30,240)

LINE (X+150,240)-(X+130,240)

LINE (X+150,240)-(X+150,190)

LINE (X-50,240)-(X-50,190)

LINE (X-50,190)-(X-10,190)

LINE (X+110,190)-(X+150,190)

CIRCLE (X+50, 190), 60, , 0, 3.14

CIRCLE (X, 240), 30, , 0, 3.14

CIRCLE (X+100, 240), 30, , 0, 3.14

NEXT X

*V вариант*

SCREEN 12

CIRCLE (320,240), 150, 14

CIRCLE (250,200), 50, 1, , , .5

PAINT (250,200), 1

LINE (340,200)-(420,200), 1

CIRCLE (320,240), 50, 14, -3.95, -5.5

CIRCLE (320,300), 60, 4, 3.14, 6.28

LINE (300,390)-(340,430), 14, BF

# ЗАКЛЮЧЕНИЕ

При наличии небольшого объема учебного времени, отводимого на тему «Этапы решения задач на компьютере», которая входит в раздел «Алгоритмы и программирование» алгоритмы и программирование мы можем изучать лишь на уровне введения. Основная задача раскрыть понятие программного управления работой компьютера. Изучение происходит на примерах простых программ на Бейсике. Показывается, как организуется простейший диалог компьютера с человеком: компьютер спрашивает, учащийся отвечает, компьютер реагирует на ответ в соответствии с его содержанием. Показывается, как организуются простейшие вычисления, например, вводится числовая последовательность, выводится ее среднее арифметическое значение; или вводятся два числа, выводится их наибольший общий делитель (алгоритм Евклида) и т.п.. Этого вполне достаточно с точки зрения поставленной цели. Изучение программирования с прагматической точки зрения заключается в освоении азов профессионального программирования. Такую цель можно ставить только перед профильным или элективным курсом информатики. Успешность учащихся в освоении этой темы во многом зависит от приобретенных ими общеучебных навыков в предыдущие годы обучения. Без сомнения, навыки, составляющие основу алгоритмического мышления, должны формироваться как можно раньше.

Многие ошибочно полагают, что современные подростки разбираются в компьютерах и при изучении информатики не должно возникать трудностей. Но как практикующий педагог не могу с этим согласиться. Все еще есть такие обучающиеся, у которых нет компьютера дома, либо он имеется, но используется только для игр. Многие после школы приходят с очень посредственными знаниями по учебной дисциплине информатика. Учитывая эти обстоятельства, считаю, что для познания основ программирования, для понимания принципа работы программы, среда программирования Qbasic отлично подходит. На примере Qbasic можно наглядно увидеть те характерные особенности, которые присущи программированию на алгоритмических языках вообще. Поэтому тексты представленных в методических рекомендациях программ отлажены именно в нем.

Мною был проведен открытый урок, где мною было продемонстрировано, как лучше преподнести данную тему обучающимся, чтобы не перегрузить их избыточными знаниями, но оставить глубокие знания о программировании, заинтересовать, и, может быть, вызвать желание заняться программированием дополнительно самостоятельно. Конспект открытого занятия содержится в данной методической разработке.

# ИСТОЧНИКИ

1. Трофимов, В. В.Информатика в 2 т. Том 1 : учебник для вузов / В. В. Трофимов, М. И. Барабанова ; ответственный редактор В. В. Трофимов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 553 с. — (Высшее образование). — ISBN 978-5-534-02613-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/470744.
2. Трофимов, В. В.Информатика в 2 т. Том 2 : учебник для среднего профессионального образования / В. В. Трофимов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 553 с. — (Профессиональное образование). — ISBN 978-5-534-02518-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/491211.
3. http://www.методкабинет.рф/index.php/publications/informatika/1719-kuzheleva.html
4. http://shakht-pedcol.ru/images/stories/metodkabinet/metodichki/№7
5. <http://elibrary.sgu.ru/uch_lit/1037.pdf>
6. https://infourok.ru/metodicheskie-ukazaniya-k-prakticheskim-rabotam-po-informatike-kurs-spo-1251517.html

# 

# ПРИЛОЖЕНИЯ