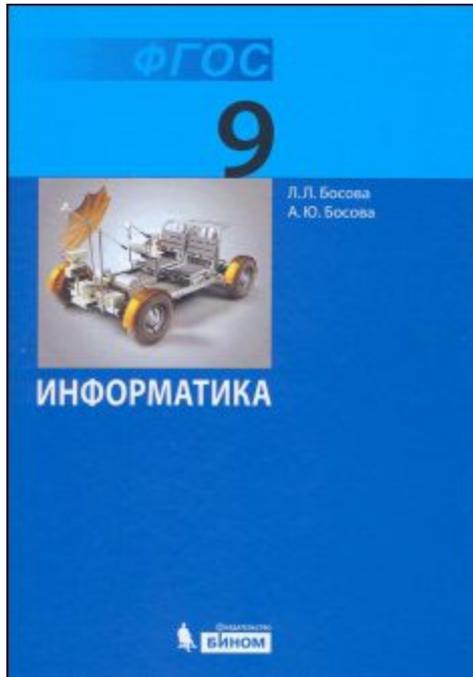


Классная работа

# Этапы решения задач с помощью компьютера

**Урок 41**

# Домашнее задание



## §2.1

Вопросы 2 – 11 устно (стр. 62-63)

Нарисовать граф «Этапы решения задач с помощью компьютера»

**БЫТЬ ГОТОВЫМ К ТЕСТУ**

# Этапы решения задач с помощью компьютера

Этап

Результат



# I. Постановка задачи

- сбор информации о задаче;
- формулировка условия задачи;
- определение исходных данных
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т.п. ).



**Ошибки при постановке задачи приводят к наиболее тяжелым последствиям!**

# II. Формализация

- анализ существующих идей и методов решения задачи
- анализ среды технических и программных средств
- разработка **математической** модели
- разработка структур данных.

# Математическая модель

**Математическая модель** – представление объекта с помощью математического описания (формулы, графики, таблицы)

# Выбор метода решения

Для одной задачи возможно **несколько** методов решения.

Основной критерий при выборе метода – **эффективность**

**Эффективность** – использование наименьших ресурсов для решения задачи

Ресурсы для программы:

- Время выполнения работы
- Объем памяти (оперативной)
- Объем внешней памяти
- Другие требования к Hardware и Software

# III. Построение алгоритма

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схема, алгоритмический язык, псевдокод и др.);
- подготовка тестов;
- проектирование алгоритма;
- прокрутка алгоритма;

# Тестирование алгоритма (прокрутка)

**Тестирование алгоритма** - это проверка алгоритма на тестах (контрольных примерах).

**Тест** (контрольный пример) – это набор исходных данных и соответствующих им результатов.

Тестирование алгоритма ручным выполнением (на бумаге, в уме) называется **прокруткой**

Добросовестная прокрутка предусматривает выполнение алгоритма с записью каждого шага в таблицу прокрутки



**Алгоритм прошел тестирование.  
Гарантирует ли это его правильность?**

# Завершение построения алгоритма

- **Анализ результатов** прокрутки
- **Уточнение** в случае необходимости постановки задачи, математической модели или метода решения, то есть возможно придется вернуться назад к этапу II или I.

# IV. Программирование

- выбор языка программирования
- уточнение способов организации данных
- запись алгоритма на выбранном языке программирования
- тестирование и отладка

# Тестирование программы (прогон)

**Тестирование** (англ. **test** — испытание) – это проверка программы на тестах (контрольных примерах).

**Тест** (контрольный пример) – это набор исходных данных и соответствующих им результатов.

Тестирование программы производится ее выполнением (прогон) в реальных условиях.



**Программа прошла тестирование.  
Гарантирует ли это её правильность?**

# Тестирование программы

Следует предусмотреть три вида тестов и процесс **тестирования** разделить на **три** этапа.

1. **Проверка в нормальных условиях.** Предполагает тестирование на основе данных, которые характерны для **реальных** условий функционирования программы.
2. **Проверка в экстремальных условиях.** Тестовые данные включают **граничные** значения области изменения входных переменных, которые должны восприниматься программой как правильные данные. Типичными примерами таких значений являются **очень маленькие** или **очень большие** числа, **предельные** объемы данных, когда массивы состоят из **слишком малого** или **слишком большого** числа элементов, и, если возможно, **отсутствие** данных.
3. **Проверка в исключительных ситуациях.** Проводится с использованием данных, значения которых лежат за **пределами допустимой** области изменений. Наихудшая ситуация складывается тогда, когда программа воспринимает **неверные данные** как правильные и выдает неверный, но правдоподобный результат.

# Тесты

Итак, **тестирование** проводится на **трёх** видах тестов.

- **Нормальные** (обычные)
- **Экстремальные** (крайние, предельные)
- **Исключительные**

# Отладка программы

**Отладка программы** – это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере.

---

Тестирование устанавливает факт наличия ошибок,  
а отладка выясняет ее причину.

# Завершение программирования

- **Анализ результатов тестирования**
- **Уточнение** в случае необходимости алгоритма, а возможно и постановки задачи, математической модели или метода решения. Возможно придется вернуться назад к этапу III, II или I.

# V. Работа с программой

Сопровождение программы — это работы, связанные с обслуживанием программы в процессе её использования.

# Задача о пути торможения автомобиля

Водитель автомобиля, движущегося с некоторой постоянной скоростью, увидев красный свет светофора, нажал на тормоз. После этого скорость автомобиля стала уменьшаться каждую секунду на 5 метров. Требуется найти расстояние, которое автомобиль пройдёт до полной остановки.

*Первый этап*

Дано:

$v_{0x}$  - начальная скорость;

$v_x$  - конечная скорость (равна нулю);

$a_x$  - ускорение (равно -5 м/с)

Требуется найти: - расстояние, которое пройдёт автомобиль до полной остановки.

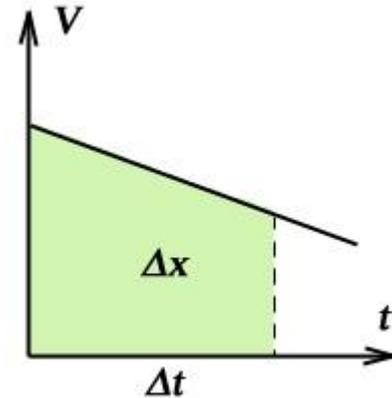


# Задача о пути торможения автомобиля

## Второй этап

В данной ситуации мы имеем дело с прямолинейным равноускоренным движением тела. Формула для перемещения при этом имеет вид:

$$s_x = \frac{v_{0x}(v_x - v_{0x})}{a_x} + \frac{a_x}{2} \left( \frac{v_x - v_{0x}}{a_x} \right)^2$$



Упростим эту формулу с учётом того, что конечная скорость равна нулю:

$$s_x = \frac{v_{0x}^2}{2a_x}$$

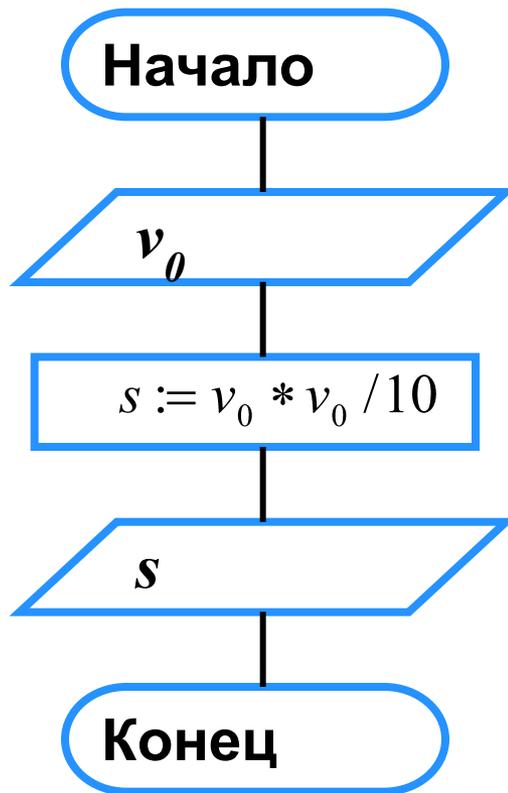
При  $a_x = -5 \text{ м/с}^2$   
получим:

$$s_x = \frac{v_{0x}^2}{10}$$

# Задача о пути торможения автомобиля

## Третий этап

Представим алгоритм решения задачи в виде блок-схемы:



# Задача о пути торможения автомобиля

## Четвёртый этап

Запишем данный алгоритм на языке программирования Паскаль:

```
program n_2;  
  var v0, s: real;  
begin  
  writeln('Вычисление длины пути торможения автомобиля');  
  write('Введите начальную скорость (м/с)> ');  
  readln (v0);  
  s:=v0*v0/10;  
  writeln ('До полной остановки автомобиль пройдет', s:8:4,  
м.)  
end.
```

# Задача о пути торможения автомобиля

## *Испытание программы*

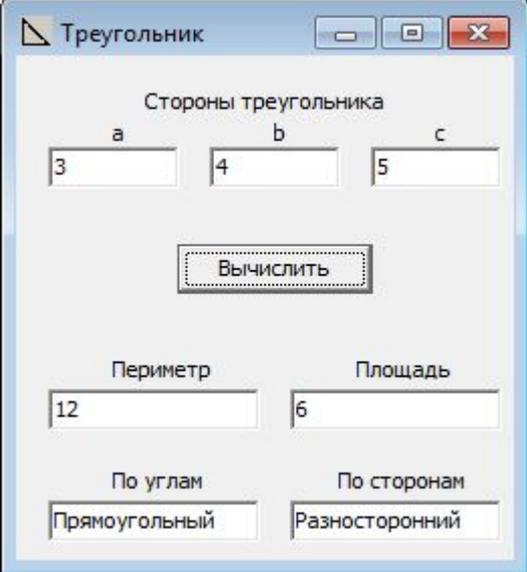
Протестировать составленную программу можно, используя ту информацию, что при скорости 72 км/ч с начала торможения до полной остановки автомобиль проходит 40 метров.

Выполнив программу несколько раз при различных исходных данных, можно сделать вывод: чем больше начальная скорость автомобиля, тем большее расстояние он пройдет с начала торможения до полной остановки.



# Работаем с компьютером

- На рабочем столе имеется программа **Треугольник**, разработанная учеником нашей школы.
- Не используя программу, придумать **6 тестов** к данной задаче и записать **в тетрадь**.
- **Открыть** программу.
- **Протестировать** программу на придуманных тестах. Результаты тестирования зафиксировать в тетради.
- **Заккрыть** программу.



Треугольник

Стороны треугольника

a	b	c
3	4	5

Вычислить

Периметр	Площадь
12	6

По углам	По сторонам
Прямоугольный	Разносторонний

**Закон Вирта** — это полушутливое высказывание, популяризированное **Никлаусом Виртом** в 1995 году. Звучит оно так:

***Программы становятся медленнее более быстро, чем компьютеры становятся быстрее.***

*Оригинальный текст (англ.):*

***Software is getting slower more rapidly than hardware becomes faster.***

**Закон Гейтса**

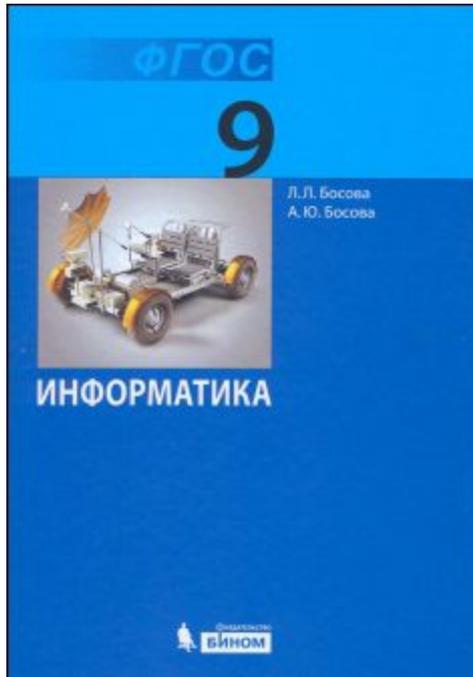
***Программы становятся в два раза медленнее каждые полтора года.***

*Оригинальный текст (англ.):*

***The speed of software halves every 18 months.***

Это может происходить по нескольким причинам: добавление избыточных ненужных функций, плохой код, нежелание программистов дорабатывать программы и плохой менеджмент или частая смена команды.

# Домашнее задание



## §2.1

Вопросы 2 – 11 устно (стр. 62-63)

Нарисовать граф «Этапы решения задач с помощью компьютера»

**БЫТЬ ГОТОВЫМ К ТЕСТУ**