

Работа с сайтом <http://informatics.mccme.ru/>

Напоминаю, что предыдущее домашнее задание включает в себя регистрацию на [сайте](#); после регистрации нужно прислать мне выбранный логин или сообщить его на ближайшем занятии.

Сайт позволяет сдавать задачи в автоматическую проверяющую систему. Для решения задач на сайте нужно писать программы, которые читают данные из консоли (позднее, когда мы разберём работу с файлами – из файла), обрабатывают их в соответствии с условием и выводят в консоль (или в файл) результат. Отправленная программа компилируется на проверяющем сервере и запускается на различных тестах, выведенные данные сравниваются с ожидаемыми, и программа получает некоторое количество баллов от 0 до 100, в зависимости от количества успешно пройденных тестов. Каждый тест – это свой набор входных данных. В случае возникновения вопросов полезно прочитать на сайте раздел [FAQ](#), а так же ознакомиться с инструкцией [«Как работать на сайте»](#).

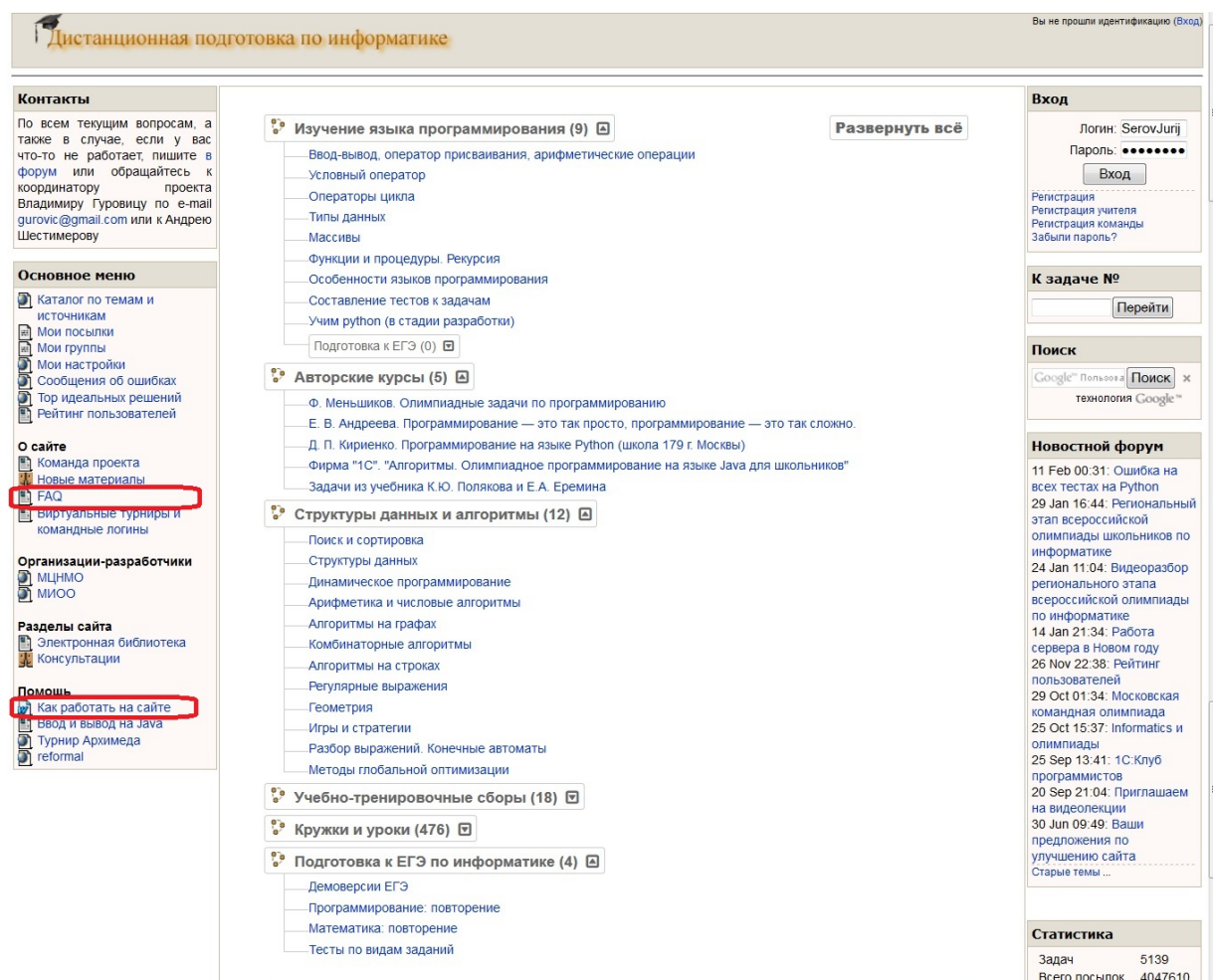


Рис. 1: Главная страница сайта.

В разделах сайта с условиями задач слева находится список, «пронумерованный» буквами (А, В, С, D, E, F, ... – см. рис. 2). Для каждой задачи есть условие («Прежде чем решать задачу, полезно ознакомиться с ее условием», ©), под которым находится панель отправки решений (см. рис. 2). В ней нужно выбрать язык, на котором написана программа, выбрать файл программы и нажать кнопку «Отправить».

К сожалению, на сайте нельзя сдавать задачи написанные на Lazarus-e, однако это можно обойти. Во-первых, можно писать на Delphi (код программы будет идентичен коду для Lazarus-a) либо на любом другом языке из списка. Во-вторых, можно в три действия изменить программу, написанную на Lazarus-e, чтобы она компилировалась в Delphi, что позволит отправить её, как программу для «Borland Delphi 6 - 14.5» (см. рис. 3):

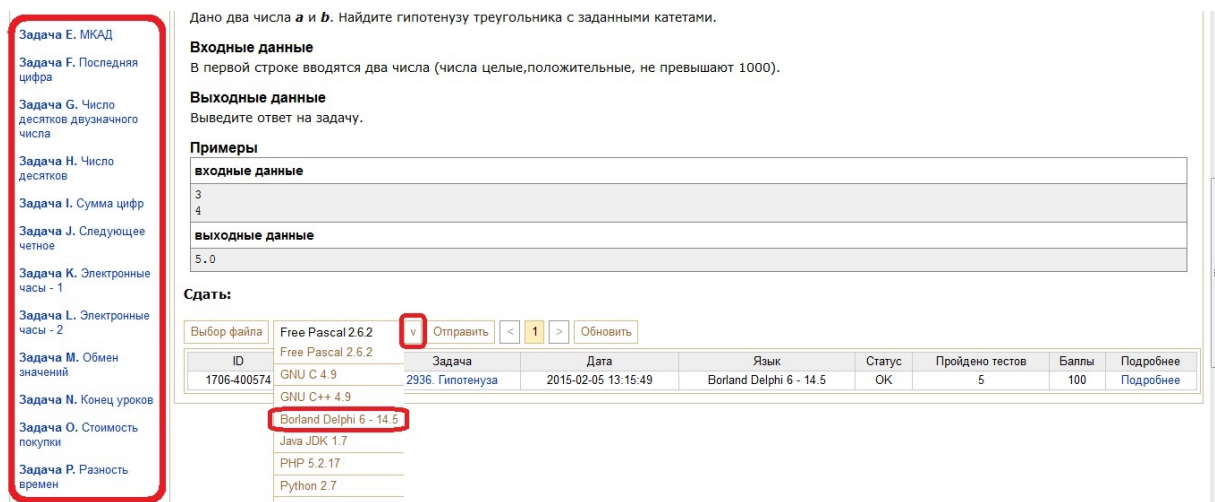


Рис. 2: Страница с задачей и формой отправки.

1. Сохранить программу написанную на Lazarus-е в какую-либо папку и найти в этой папке её код (файл с расширением ***.lpr**). На всякий случай создать его копию.
2. Переименовать файл, изменив расширение с ***.lpr** (Lazarus PProject) на ***.dpr** (Delphi PProject).
3. Открыть получившийся **dpr**-файл любым текстовым редактором (например блокнотом), и после объявления начала программы (командой **program ...;**) добавить строку **{\$APPTYPE CONSOLE}**.

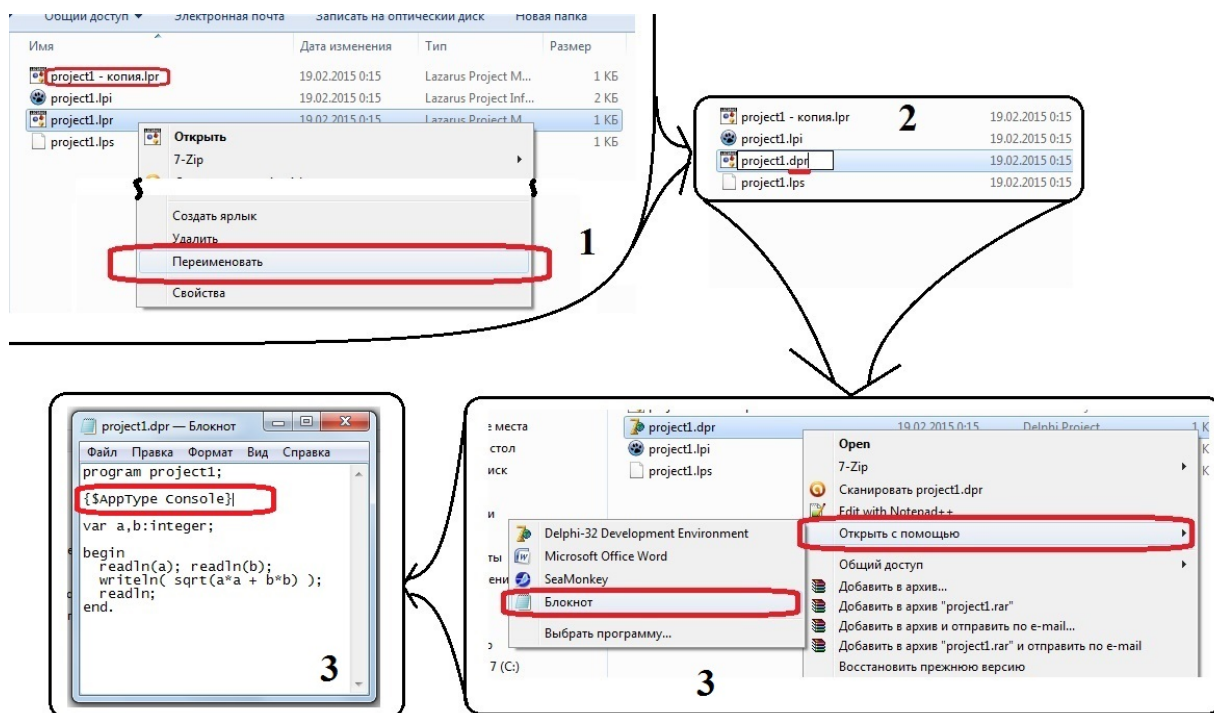


Рис. 3: Изменение программы.

P.S. Если у вас не показывается расширение файлов в проводнике, то нужно поменять настройки отображения папок и файлов. Как это сделать для Windows (XP, 7, 8) написано, например, [здесь](#), или [здесь](#), или в любом другом месте интернета.

Домашнее задание: написать и сдать на сайте задачи **A, B, C, D** и **M** [отсюда](#).

При написании этих и последующих задач вам может пригодиться список процедур и функций, представленный ниже. Там нету уже упоминавшихся операторов **div** и **mod**, которые нужны для некоторых из этих задач.

Математические функции

Наименование функции	Тип аргумента	Тип значения	Результат вычисления
Abs(x)	целый вещественный	целый вещественный	Абсолютное значение "x"
Sin(x)	вещественный	вещественный	синус "x" рад.
Cos(x)	вещественный	вещественный	косинус "x" рад.
Arctan(x)	вещественный	вещественный	арктангенс "x" ($-\pi/2 < y < \pi/2$)
Sqrt(x)	вещественный	вещественный	квадратный корень из "x"
Sqr(x)	целый вещественный	целый вещественный	значение "x" в квадрате (x^2)
Power(a,x)	вещественный	вещественный	значение "a" в степени "x" (a^x)
Exp(x)	вещественный	вещественный	значение "e" в степени "x" (e^x , где $e = 2.718282\dots$)
Ln(x)	вещественный	вещественный	натуральный логарифм "x" ($x > 0$)
Frac(x)	вещественный	вещественный	дробная часть "x"
Int(x)	вещественный	вещественный	целая часть "x"
Random	-	вещественный	случайное число ($0 \leq y < 1$)
Random(x)	Word	Word	случайное число ($0 \leq y < x$)
Succ(c)	порядковый	порядковый	следующий за "c" символ
Pred(c)	порядковый	порядковый	предшествующий "c" символ

Математические процедуры

Наименование функции	Тип аргумента	Тип значения	Результат вычисления
Inc(x)	целый	целый	Увеличивает "x" на 1 ($x := x + 1$;)
Dec(x)	целый	целый	Уменьшает "x" на 1 ($x := x - 1$;)
Inc(x , n)	целый	целый	"x" на n ($x := x + n$;)
Dec(x , n)	целый	целый	"x" на n ($x := x - n$;)

Процедуры преобразования типов переменных

Наименование функции	Тип аргумента	Тип значения	Результат вычисления
Str(x , s)	x-целый или вещественный	s-строковый	Последовательность символов "s" из цифр числа "x"
Val(s , v, cod)	s-строковый	v-целый или вещественный cod- целый	Двоичная форма числа последовательности "s" cod=0 (код ошибки)

Функции преобразования типов переменных

Наименование функции	Тип аргумента	Тип значения	Результат вычисления
Trunc(x)	вещественный	LongInt	целая часть "x"
Round(x)	вещественный	LongInt	округление "x" до целого
Odd(x)	целый	логический	возвращает True если "x" - нечетное число
Chr(x)	Byte	Char	Символ ASCII кода "x"
Ord(x)	Char	Byte	ASCII код символа "x"