**Знакомство с типами int и String**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 1](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=1), Лекция 4

— Привет, Амиго.

— Здравствуйте, Эллеонора Керри.

— Называй меня просто Элли, а то звучит как-то слишком официально.

— ОК, Элли.

— Я думаю, что с моей помощью ты быстро станешь одним из лучших. У меня большой опыт в обучении новичков. Слушай меня, и дело пойдёт как по маслу. Итак, приступим.

— В **Java** есть два основных типа: **String** и **int**. В String мы храним строки/текст, а в int – числа (целые). Чтобы объявить новую переменную, нужно написать её тип и имя. Это имя не должно быть именем другой переменной и/или функции.

|  |  |
| --- | --- |
| **Пример 1, код:** | **Описание** |
| String s; | Объявлена новая переменная s, способная хранить текст/строку. |
| int i; | Объявлена новая переменная i, в которой можно хранить целые числа. |

— При объявлении переменных в них сразу можно заносить их значения.

|  |  |
| --- | --- |
| **Пример 2, код:** | **Описание** |
| String s = "Elly"; | В переменной s хранится строка со значением "Elly". |
| int i = 5; | В переменной i хранится число 5. |

— Для того, чтобы занести новое значение в переменную, используется знак “=”. Его ещё называют **«оператором присваивания»**. Присваивание – это занесение в переменную значения, взятого из другой переменной или вычисленного на основе нескольких переменных.

|  |  |
| --- | --- |
| **Пример 3, код:** | **Описание** |
| int a = 5; | В переменной a хранится значение 5. |
| int b = 6; | В переменной b хранится значение 6. |
| int c = a + b; | В переменной c хранится значение 11. |

— При вычислении нового значения переменной может использоваться её старое значение.

|  |  |
| --- | --- |
| **Пример 4, код:** | **Описание** |
| int a = 2; | Теперь a равно 2 |
| int b = 3; | Теперь b равно 3 |
| a = a + b; | Теперь a равно 5 |
| b = b + 1; | Теперь b равно 4 |

— Строки можно объединять, используя знак плюс:

|  |  |
| --- | --- |
| **Пример 5, код:** | **Описание** |
| String s1 = "Мама";  String s2 = "Мыла";  String s3 = s1 + s2 + "Раму"; | В переменной s3 содержится значение «МамаМылаРаму» |

— Иногда удобно использовать строки, состоящие из одного или нескольких пробелов:

|  |  |
| --- | --- |
| **Пример 6, код:** | **Описание** |
| String s1 = "Мой любимый фильм";  String s2 = "Дорога";  int roadNumber = 66;  String text = s1 + " " + s2 + " " + roadNumber; | text содержит «Мой любимый фильм Дорога 66» |

Сейчас мы рассмотрим, как выводить текст и значение переменных на экран:

[](https://cdn.javarush.ru/images/article/255f0e57-ca1b-4633-a839-89af3d07d469/original.jpeg)

|  |  |
| --- | --- |
|  | **Пример 7, код:** |
| 1 | System.out.println("Ты должен делать то, что должен"); |
| 2 | String s = "Ты должен делать то, что должен";  System.out.println(s); |

— Кстати, Диего просил передать тебе пару задачек:

1

Задача

Java Syntax,  1 уровень,  4 лекция

JavaRush. Learn once - use anywhere

Загадка: «Написана один раз, запускается везде». Ответ: Java-программа. Эта загадка правдива потому, что Java логична, и структура у неё правильная. Перефразируем этот лозунг для студентов: «изучи однократно, используй многократно!». Вот вам задачка на эту тему: напишите программу, которая 10 раз выводит на экран одну полезную фразу.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Ну это… С новым годом!

Представляете, если вы не забросите учёбу и будете продолжать решать задачи и читать лекции, то следующий новый год, который наступит не раньше, чем через три месяца, вы сможете встретить уже настоящим программистом! Это — мечты. А пока — за работу. Убираем ненужные комментарии и выводим на экран поздравление с новым годом.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Вносим изменения в код

Редактировать чужой код порой сложнее, чем свой писать, уж поверьте опыту специалистов секретного центра JavaRush. Вот почему в нашем учебном плане есть задачки на исправление кода. Пока что открываем задачник, вчитываемся в простой код и вносим в него изменения так, чтобы переменная name приняла значение Amigo.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Еще немного исправлений

В статьях типа «100500 советов как стать классным программистом» и им подобных почти всегда присутствует вариация на тему фразы «чужой код учит, как писать свой собственный». Программист редко работает один, так что это — чистая правда и без этого никуда. Приучаемся к командной работе, исправляем чужой код.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Закомментируйте лишнее

Комментарии гораздо полезнее, чем кажутся на первый взгляд! В них можно спрятать сокровенные мысли, и ни один компилятор их не прочитает! Только коллега-программист. Так вот, в нашей программе есть какие-то лишние строки, которые мешают программе работать так, как надо. Исправьте ситуацию, закомментировав лишнее.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Мне снова 15(!)

Перед вами чужой код. Он загадочен и неправилен. Но в ваших руках сосредоточена власть программиста, вы можете все поменять, и да поможет вам компилятор справиться с этой задачей. Что нужно сделать для того, чтобы вернуть хрупкое равновесие и мир? Просто внести изменения в программу так, чтобы переменная age приняла значение 15.

1

Задача

Java Syntax,  1 уровень,  4 лекция

Подойдет только 20

Многие программисты буквально помешаны на головоломках, всяких там кубиках Рубика, «переставь две спички и получи 100», башенках и прочих. Так уж устроены их мозги. Давайте тренироваться решать головоломки в программах. В этой — расставляем знаки плюс и минус так, чтобы значение переменной result равнялось 20.