

Введение в программирование

Часть 2

Смена локали

- `setlocale(LC_ALL, "RUSSIAN");` // Смена локали
- `setlocale(0, "");` // Смена локали

Другой способ вывода специальных СИМВОЛОВ

- В стандарте языка C++ 11 программистам предоставили новую возможность для вывода специальных СИМВОЛОВ на экран.
- Для этого необходимо использовать «raw» строки.

```
cout<<R"(hello\nworld)";  
cout<<R>("Test 'string'\t");  
cout<<R"((Such brackets))";
```



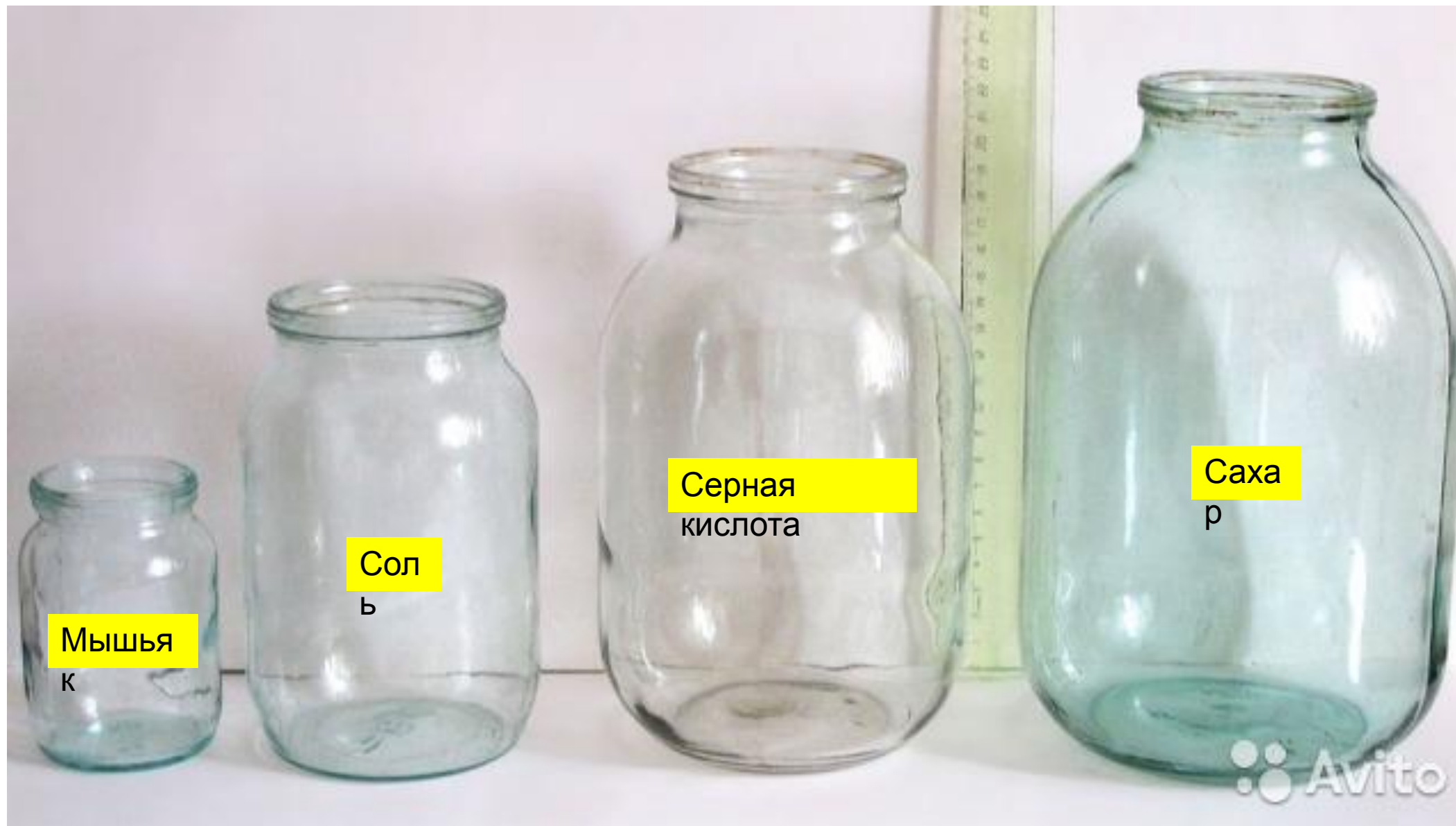


Переменные

ТАРА (ёмкость) для данных

Переменная

- **Переменная** — ИМЕНОВАННАЯ область оперативной памяти, предназначенная для хранения данных, значение которых **может быть изменено**.



Мышь

к

Сол

ь

Серная
кислота

Саха
р



Здравствуй, а я
злой
компилятор
КОМП, я очень
нудный.

A cartoon illustration of a man and a woman shaking hands. The man on the left has yellow hair and is wearing a blue polo shirt with a name tag that says 'КОМП'. The woman on the right has black hair and is wearing a red long-sleeved shirt with a name tag that says 'Anita'. She is also holding a book with 'BOOK' written on its spine. Both characters have speech bubbles above them containing Russian text.

Привет, я
переменная
anita, я вся такая
разная

Объявления переменных

- Для того, чтобы использовать в коде программы переменные, их необходимо предварительно **ОБЪЯВИТЬ**, т. е. объяснить компилятору, что какой тип данных мы хотим хранить и под каким именем.

Объявления переменных

Тип_данных имяПеременной;

int myAge;

- При объявлении переменную можно сразу инициализировать, т.е. присвоить ей первоначальное значение:

int myAge = 20;

Показ значений переменной на экран

Общий синтаксис:

```
cout << имяПеременной;
```

Пример вывода значения переменной на экран:

```
cout << "Значение переменной myAge = " << myAge;  
// переменная пишется БЕЗ КАВЫЧЕК!!!
```

Обратите внимание, имя переменной **пишется без кавычек.**

Изменение значения переменной

Итак, раз она переменная, значит её значение может изменяться в процессе работы программы.

```
int myAge = 20;
```

```
cout << «Первое значение переменной myAge = "<< myAge << endl;
```

```
myAge = 22;
```

```
cout << «Второе значение переменной myAge = "<< myAge << endl;
```

```
myAge = 55 + 25;
```

```
cout << «Третье значение переменной myAge = "<< myAge << endl;
```

```
myAge = myAge + 25;
```

```
cout << «Четвёртое значение переменной myAge = "<< myAge << endl;
```



А что значит
переменная:
«something» ???

Правила хорошего тона

- Имена переменных **обязательно** должны быть **осмысленными** и максимально точно характеризовать назначение (их смысл) хранимых в переменной данных.
- Имена переменным, следует давать **СТРОГО** на **английском языке**.
- Все переменные, по возможности, необходимо объявлять в начале блока (т.е. в самом начале блока выделенного символами фигурных скобок { });

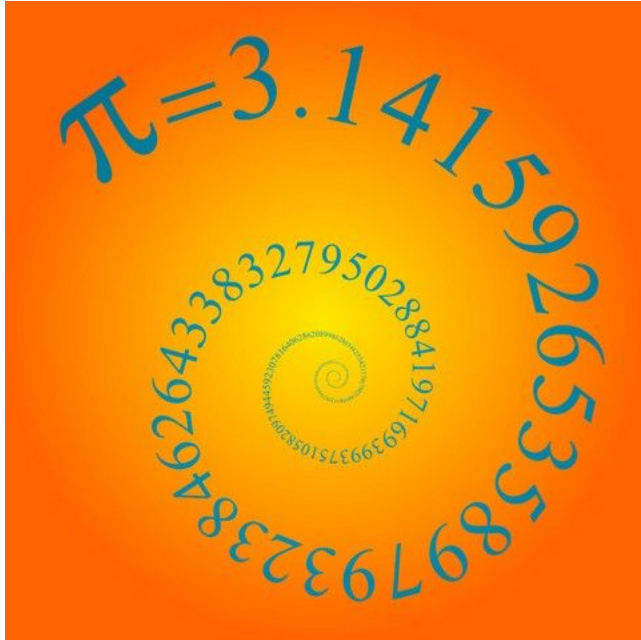
Что такое верблюжий стиль?



Правила хорошего тона

- Имена переменных должны быть записаны в **смешанном Регистре**, начиная с нижнего, например:

```
int myAge;
```



c = 300000 км/с

$$C = 300000$$

km/c

π

=

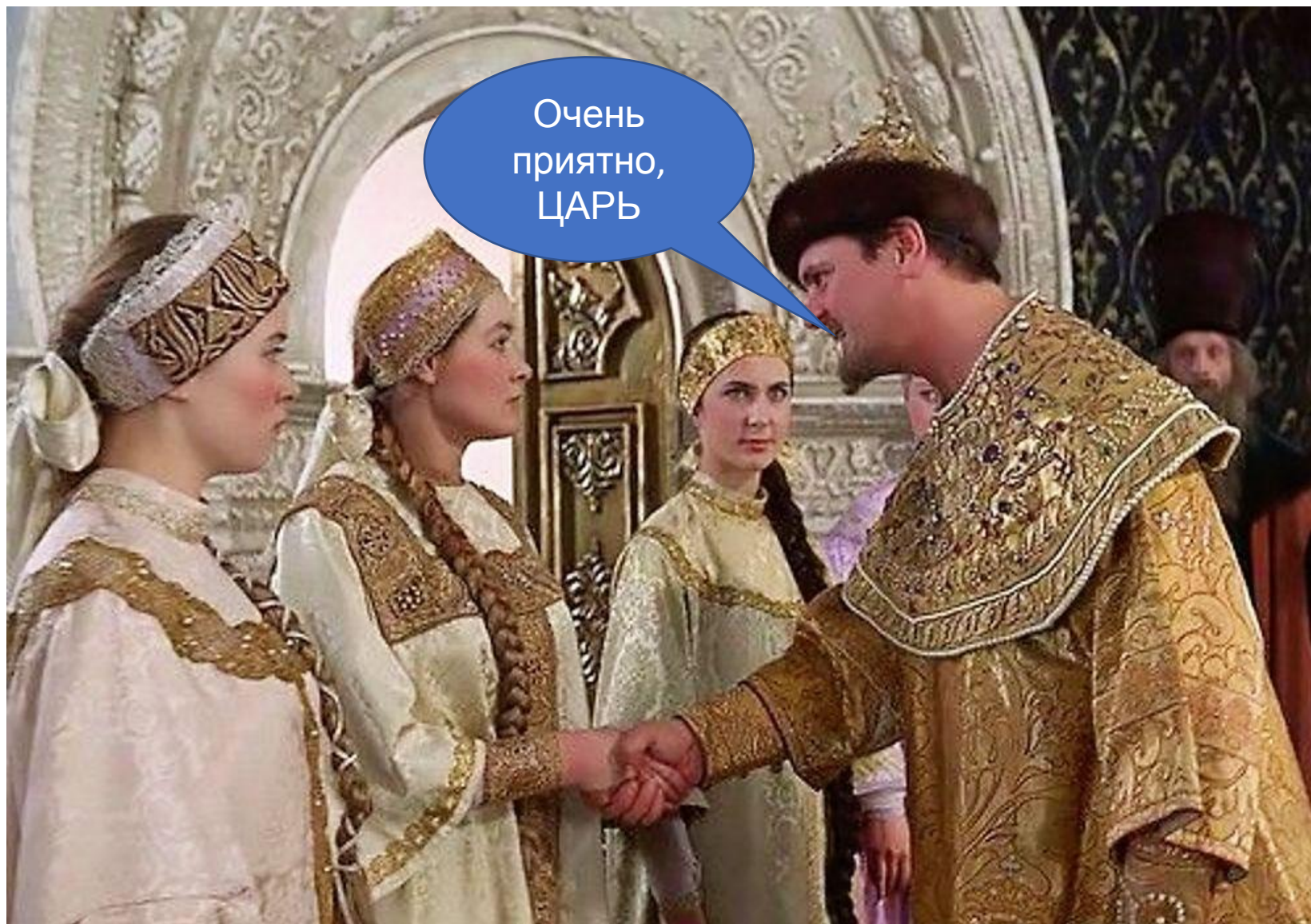
3,14159265358....

Константы

Данности, с которыми придётся смириться.

Константа

- **Константа** — ИМЕНОВАННАЯ область оперативной памяти, предназначенная для хранения данных, значение которых НЕ может быть изменено.



Объявления переменных

- Для того, чтобы использовать в коде программы **константы**, как и переменные, необходимо предварительно **ОБЪЯВИТЬ**, т.е. объяснить компилятору, что какой тип данных мы хотим хранить, под каким именем и присвоить значение этой константе.

Объявления констант

const тип_данных ИМЯ_КОНСТАНТЫ = значение;

const int COUNT_DAYS_IN_WEEK = 7;

При объявлении констант, необходимо **СРАЗУ присвоить ей значение**, иначе компилятор выдаст ошибку.

Показ значений константы на экран

Общий синтаксис:

`cout<< имяПеременной_или_КОНСТАНТЫ;`

`cout << "Значение Константы = "<< COUNT_DAYS_IN_WEEK << endl;`

Показ значений константы на экран

Общий синтаксис:

```
cout<< имяПеременной_или_КОНСТАНТЫ;
```

```
COUNT_DAYS_IN_WEEK = 20;
```

```
cout << "Значение Константы = "<< COUNT_DAYS_IN_WEEK << endl;
```

Использование констант

Константы можно использовать как и переменные в формулах, при расчётах;

```
const int COUNT_DAYS_IN_WEEK = 7;
```

```
cout << "Значение Константы = "<< COUNT_DAYS_IN_WEEK << endl;
```

```
int myAge = 20;
```

```
cout << "Первое значение переменной myAge = "<< myAge << endl;
```

```
myAge = myAge * COUNT_DAYS_IN_WEEK ;
```

```
cout << "Второе значение переменной myAge = "<< myAge << endl;
```

Правила хорошего тона

- Имена КОНСТАНТ **обязательно** должны быть **осмысленными** и максимально точно характеризовать назначение хранимых данных.
- Имена константам, следует давать **СТРОГО на английском языке**.
- Все КОНСТАНТЫ, по возможности, необходимо объявлять в начале блока (т.е. в самом начале блока выделенного символами фигурных скобок { }), перед переменными;

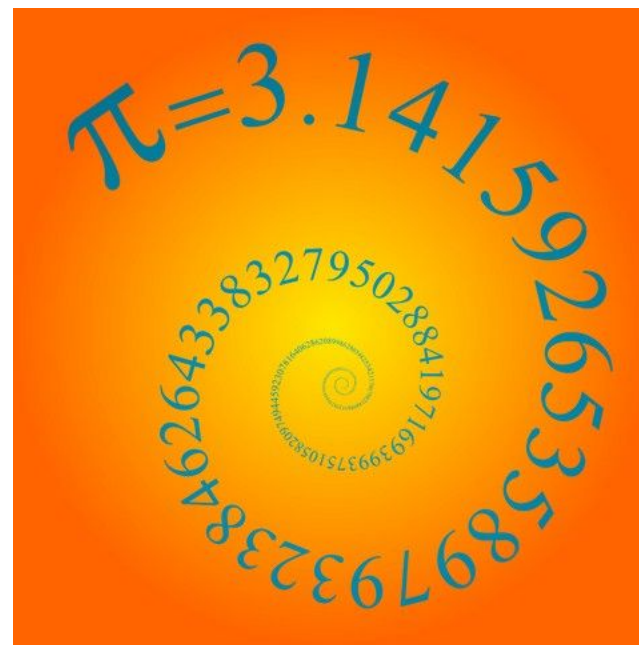
Правила хорошего тона

- Именованные константы (включая значения перечислений) должны быть записаны в **ВЕРХНЕМ_РЕГИСТРЕ_С_НИЖНИМ_ПОДЧЁРКИВАНИЕМ** в качестве разделителя. Например:

```
const int COUNT_DAYS_IN_WEEK = 7;
```

Преимущество использования констант

- Упрощение процесса внесения изменений: значение константы задано в программе всего в одном месте.
- Улучшение читаемости и понятности кода (отсутствие магических цифр).



Переменные и константы



Переменные и константы

- **Переменная** — ИМЕНОВАННАЯ область оперативной памяти, предназначенная для хранения данных, значение которых **может быть изменено**.
- **Константа** — ИМЕНОВАННАЯ область оперативной памяти, предназначенная для хранения данных, значение которых **НЕ может быть изменено**.

Ввод данных

- Для ввода данных в программу используется команда **cin**.

cin>> имяПеременной;

cin>> myAge;

Ввод данных

Пример получения данных (ввода) от пользователя.

```
int myAge;
```

```
cout << "Введите Ваш возраст";
```

```
cin>> myAge;
```

```
cout << "Понял, не дурак, Ваш возраст равен " << myAge << " годам";
```

Ввод данных

- Ввод сразу нескольких переменных, записывают таким образом:

cin>> имяПеременной1 >> имяПеременной2 >>...>>
имяПеременнойN;

cin>> quantity >> price >> discount;

Пример

// Заголовок

#include <iostream>

// определение пространства имен, в котором есть cout<<

using namespace std;

// Главная функция

int main()

{

Пример

```
    setlocale(LC_ALL, "RUSSIAN");  
    //Объявляем переменную Discount  
    const float DISCOUNT = 0.05;  
    //Объявляем переменную Cost  
    float cost=10.50;  
    //Приглашение ввести цену товара  
    cout<<"Укажите цену:\n";  
    //Ввод значения в переменную Cost  
    cin>>cost;  
    //Объявляем переменную Count  
    int count=5;
```

Пример

```
// Приглашение ввести количество  
cout<<"Укажите количество:\n";  
// Ввод значения в переменную Count  
cin>>count;  
// Объявляем переменную Price  
float price;  
// Вычисляем значение переменной Price  
price = count * cost - count * cost * DISCOUNT;  
// Выводим итоговую стоимость товара со скидкой  
cout<<"Заплатите, пожалуйста: "<<price<<" тенге\n";  
    return 0;  
}
```

Спасибо за внимание.