Занятие 2: Переменные и типы данных. Ввод и вывод информации

Цели занятия: Знакомство с элементарными типами данных (целые, дробные числа, строка), освоение основных операций с простыми типами. ввод и вывод информации в консоль

Результат: познакомились с элементарными типами данных, изучили основные действия с переменными, научились вводить и выводить данные из программы

Введение

На данном занятии нам необходимо разобраться, что такое переменные, какую информацию они могут хранить и причём тут типы данных. Также мы разберёмся, как вводить и выводить информацию для взаимодействия с программой.

Переменная

В языке C++, как и в любом другом языке программирования информация хранится систематизировано, т.е. и мы, и программа всегда знают, где искать необходимую информацию.

Переменная – та самая единица информации, в которой содержатся не только какие-то необходимые данные, но и информация о том, как взаимодействовать с данной информацией.

Давайте разберём основные типы данных. В данном уроке мы рассматриваем только элементарные типы, о пользовательских говорить будем намного позже.

Тип данных – такая структурная единица языка, которая определяет, как и какие данные могут хранится в памяти компьютера.

Для того, чтобы создать переменную в коде, существуют несколько правил:

<тип данных> <имя переменной> = <какая-то информация>;

Типы данных

Элементарные типы данных делятся на две категории: числовые и символьные. В первой категории мы будем хранить какую-либо численную информацию, во второй будем хранить текст и символы языка.

Числовые типы данных в свою очередь делятся на целочисленные и дробные.

Целочисленный тип у нас один – **int**.

Числа с плавающей запятой хранятся в **float** и **double**.

Также существуют модификаторы типа – **unsigned**, **short** для int, а также **long** и **long long** для всех численных типов.

Символьные типы данных у нас представлены в виде двух типов данных **char** и **string**.

Теперь поговорим конкретно про каждый:

int – целочисленный тип, название происходит от англ. integer – «целый». Отличительная особенность данного типа, как понятно из названия, - хранить только целые числа. 10 мы сохранить можем, 10.21 уже не можем.

int ограничен в значениях – от -2 147 483 648 (минус два миллиарда сто сорок семь миллионов четыреста восемьдесят три тысячи шестьсот сорок восемь) до 2 147 483 647

Немного про модификаторы (тут важно опереться на возраст и опыт учеников, и чтобы не перегрузить информацией пропустить материал про модификаторы, это повышенный уровень сложности)

Если хочется больше значений хранить – можно использовать модификатор long или long long (по факту одно и то же, отличия появляются в разных операционных системах).

Если хочется экономить память, можно использовать модификатор short, который сокращает допустимые значения от -32 768 до 32 767

Также к уже существующим модификаторам мы можем добавить модификатор unsigned, который уберет все отрицательные числа из типа, расширив при этом лимит положительных значений (положительный лимит станет в два раза больше).

С целочисленными типами все математические операции работают привычно, кроме деления. Для деления работает логика деления в столбик – будет результат деления и отдельно остаток.

Результат от деления получается с помощью оператора /, а остаток от деления можно найти с помощью %.

Как пример: 11/2 = 5, 11%2 = 1, 5 - результат деления, 1 – остаток

Можно написать с учениками код, который будет показывать возможности математических операций.

Пример кода:

```
#include <iostream>

using namespace std;

int main()
{
	int a = 11;
	int b = 5;
	cout << a + b << endl;
	cout << a - b << endl;
	cout << a * b << endl;
	cout << a / b << endl;
	cout << a % b << endl;
}
```

Также важными механизмами взаимодействия с целыми числами являются инкрементирование (++) и декрементирование (--). По своей сути это операции прибавления и вычитания единицы.

Если мы ставим ++ перед именем переменной, то сначала меняется значение переменной, а потом используется само значение. Если после имени переменной, то сначала используется значение из переменной, а потом оно меняется. С -- ситуация аналогичная.

Пример кода:

```
#include <iostream>

using namespace std;

int main()
{
	int a = 0;
	int b = 0;
	cout << a++ << endl;
	cout << ++b << endl;
}
```

С целочисленными типами разобрались, теперь поговорим про дробные числа.

Дробные числа представлены типами float и double.

float принимает значения от - 3.4*10^38 до 3.4*10^38 (очень большие числа)

double принимает значения ещё больше: - 1,7*10^308 до 1,7*10^308

long double принимает ещё большие значения, но размер зависит от операционной системы.

Главный недостаток типов с плавающей запятой – неточность. Если нам требуются сверхсложные вычисления с огромной точностью – такие типы нам не подойдут, но для повседневных задач самое то.

Все математические операции работают также, даже деление работает как в привычной математике. Остаток от деления найти уже нельзя с помощью одного оператора.

Проверим кодом:

```
#include <iostream>

using namespace std;

int main()
{
	float a = 11;
	float b = 5;
	cout << a + b << endl;
	cout << a - b << endl;
	cout << a * b << endl;
	cout << a / b << endl;
}
```

Самое главное правило – на ноль делить нельзя. В случае с целыми числами мы можем получить ошибку, а с дробными получим значение inf, -inf или undefined.

Последнее про что мы поговорим – символьные типы.

Тип char хранит один символ – цифры, буквы, различные знаки препинания.

string хранит много символов внутри, т.н. символьную строку.

Пример кода:

```
#include <iostream>

using namespace std;

int main()
{
	string input;
	cin >> input;
	cout << "Your word: " << input;
}
```

Ввод и вывод данных

С помощью cin мы можем вводить не только текстовые данные, но и численные значения, с которыми потом можно работать.

Пример кода:

```
#include <iostream>

using namespace std;

int main()
{
	int a;
	int b;
	cin >> a >> b;
	cout << a + b;
}
```

Дробные числа вводим через ., т.е. 12.11, 17.33322 и т.д.

Далее можно попрактиковаться с вводом выводом, на усмотрение преподавателя.

Домашнее задание

Написать две программы:

1. Программа, которая складывает два числа (ввод a,b: вывод a+b)
2. Написать приложение «Диалог», творческое задание
   Пример решения:

```
#include <iostream>

using namespace std;

int main()
{
	string input;
	int years_old;
	cout << "Hello! What's your name?" << endl;
	cin >> input;
	cout << "Hi, " << input << "! ";
	cout << "How old are you?" << endl;
	cin >> years_old;
	cout << "It's cool! I am " << years_old + 5 << " years
old.";
}
```