

Национальный исследовательский университет
«Высшая Школа Экономики»
Факультет компьютерных наук

C++ как первый язык программирования

Лекция №2

Лекторы: Кирилл Лунёв, Роман Халкечев

Цели лекции и семинара

- Познакомиться со встроенными типами данных, выражениями и операторами C++
- Изучить основные инструкции C++ (if, while, do, for, switch, goto, ...)



Поехали!

План лекции

- Про типизации
- Типы данных и переменные
- Ключевое слово const
- Про lvalue и rvalue
- Операторы
- Инструкции (if, while, do, for, switch, goto, ...)
- Ключевые слова break и continue

Про типизации

Существует два вида типизаций:

- **Статическая** - тип переменной фиксируется в момент объявления и не может быть в последствии изменён
- **Динамическая** - тип переменной определяется в момент присваивания ей значения и может быть неоднократно изменён

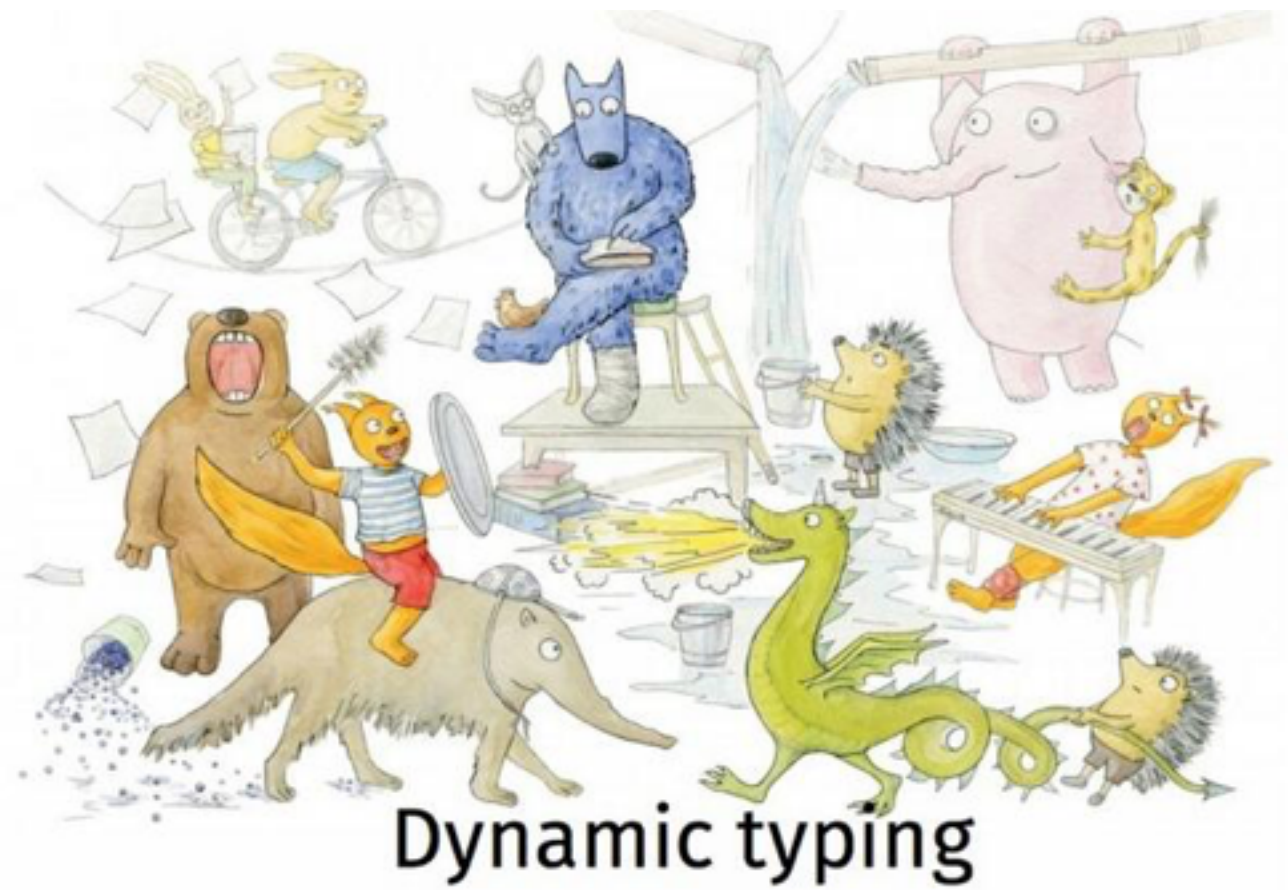
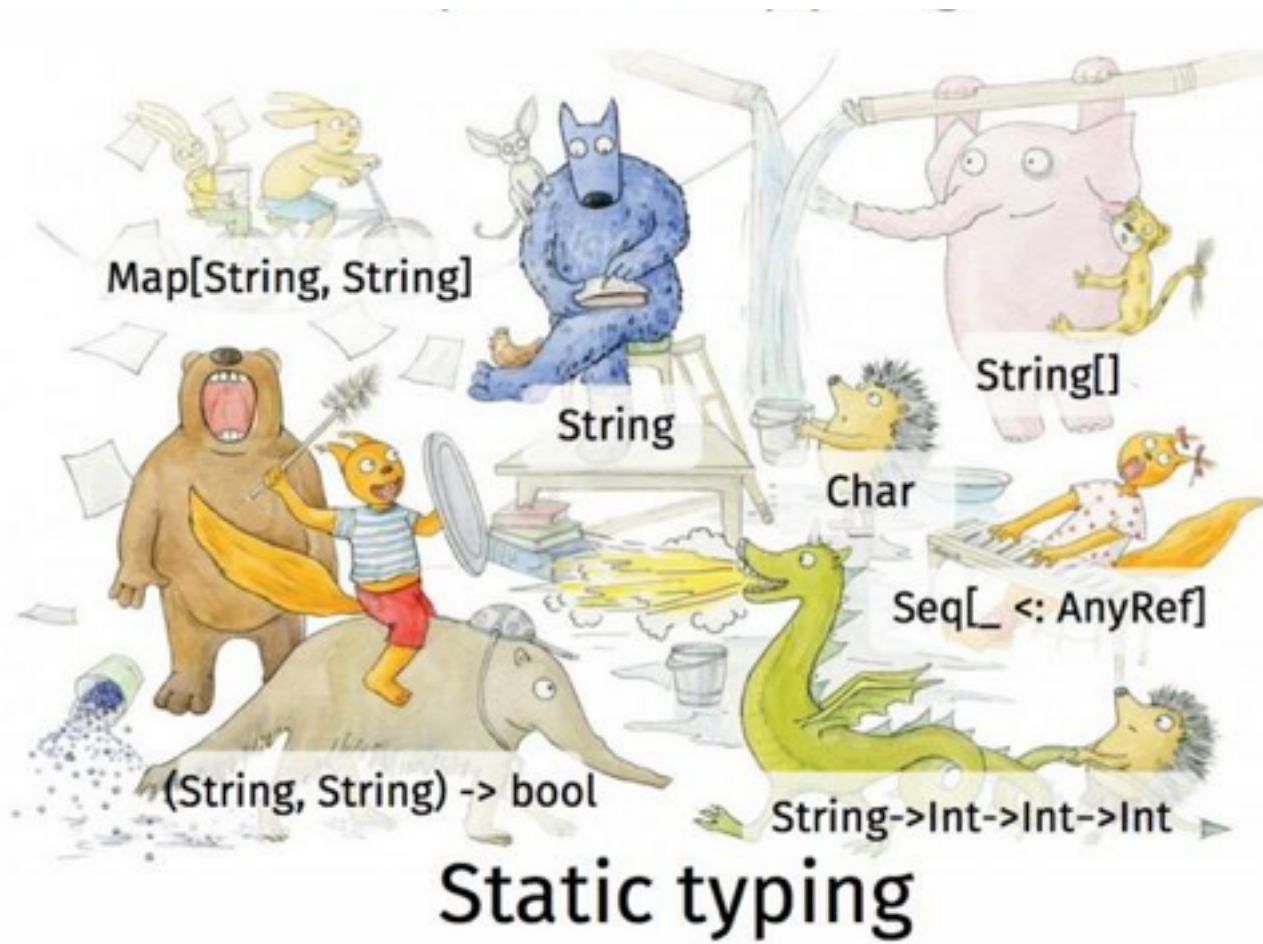
Про типизации

```
int variable;
```

```
variable = 33;
```

```
variable = "Mr. Smith";
```

Про типизации



Статическая типизация

Преимущества:

- Многие ошибки исключаются на стадии компиляции
- Даёт самый простой машинный код
- Скорость работы увеличивается за счёт знаний о типах

Недостатки:

- Программа становится более многословной*

Динамическая типизация

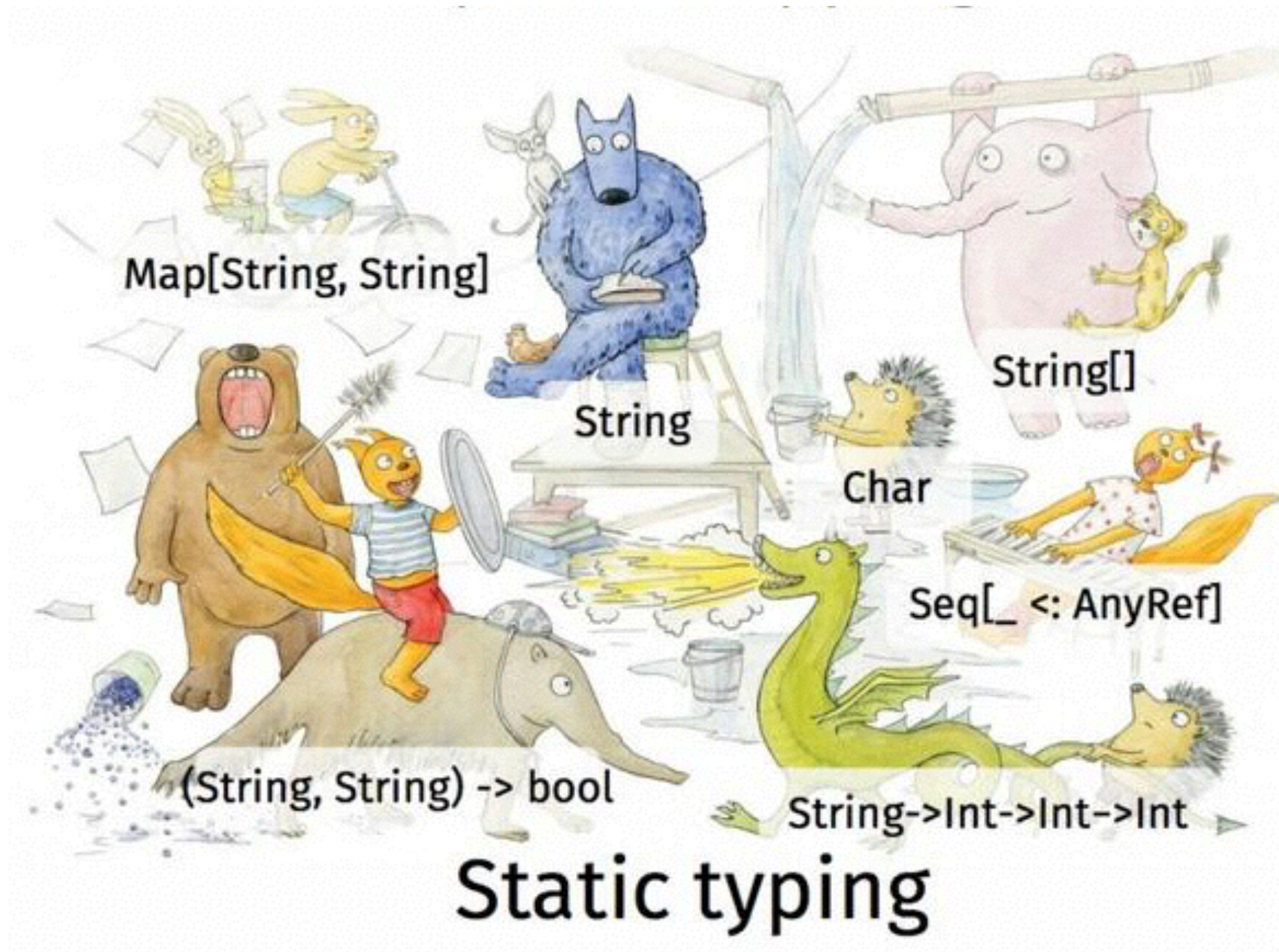
Преимущества:

- Упрощается написание несложных программ
- Код программ становится лаконичней

Недостатки:

- Невозможность отловить многие ошибки на этапе компиляции
- Низкая скорость, связанная с динамической проверкой типа
- Невозможна перегрузка функций по типу данных

C++ - язык со статической типизацией

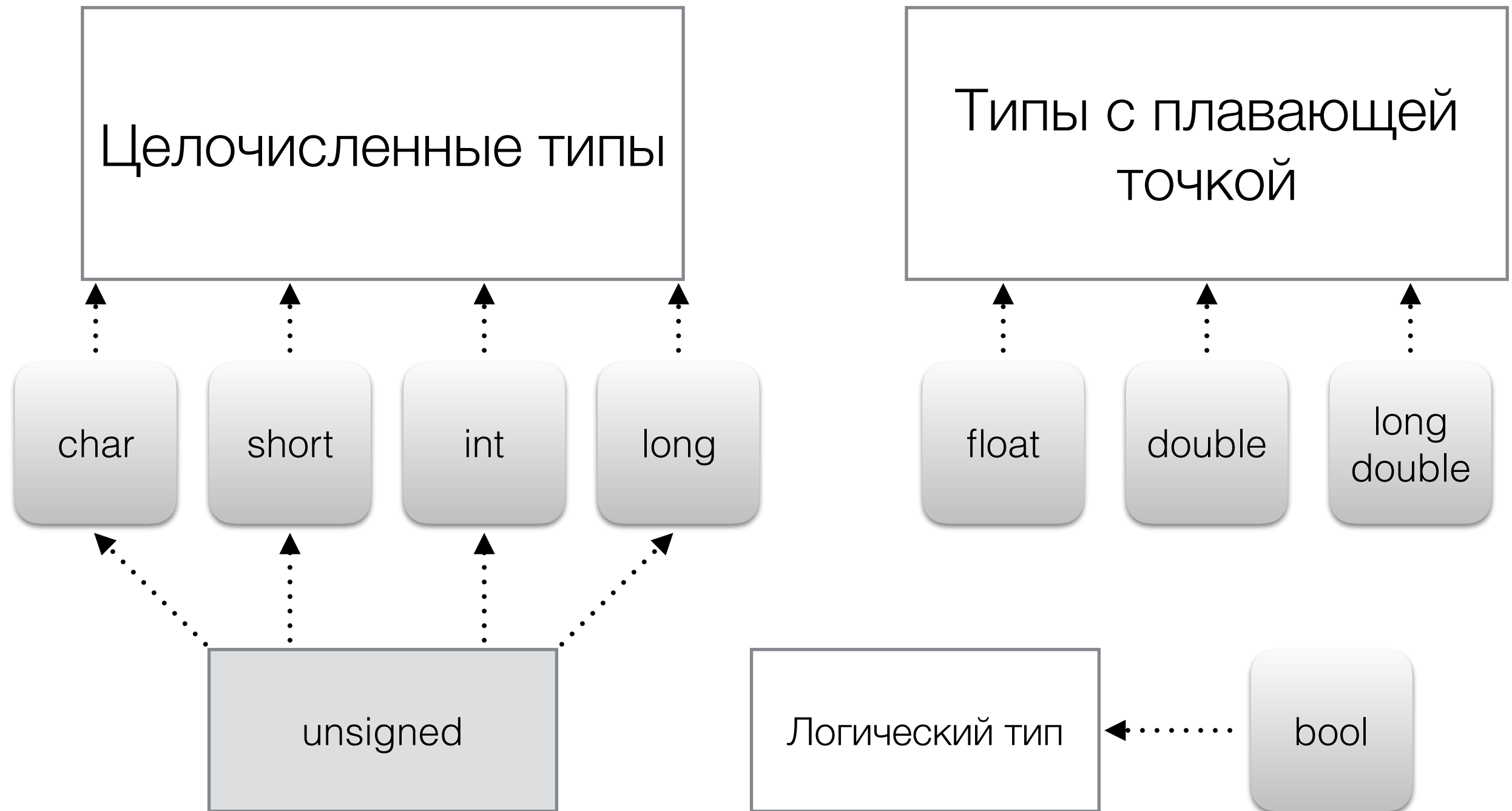


Типы данных и переменные

Типы данных в C++:

- Основные типы
- Производные типы
- Пользовательские

Основные типы данных



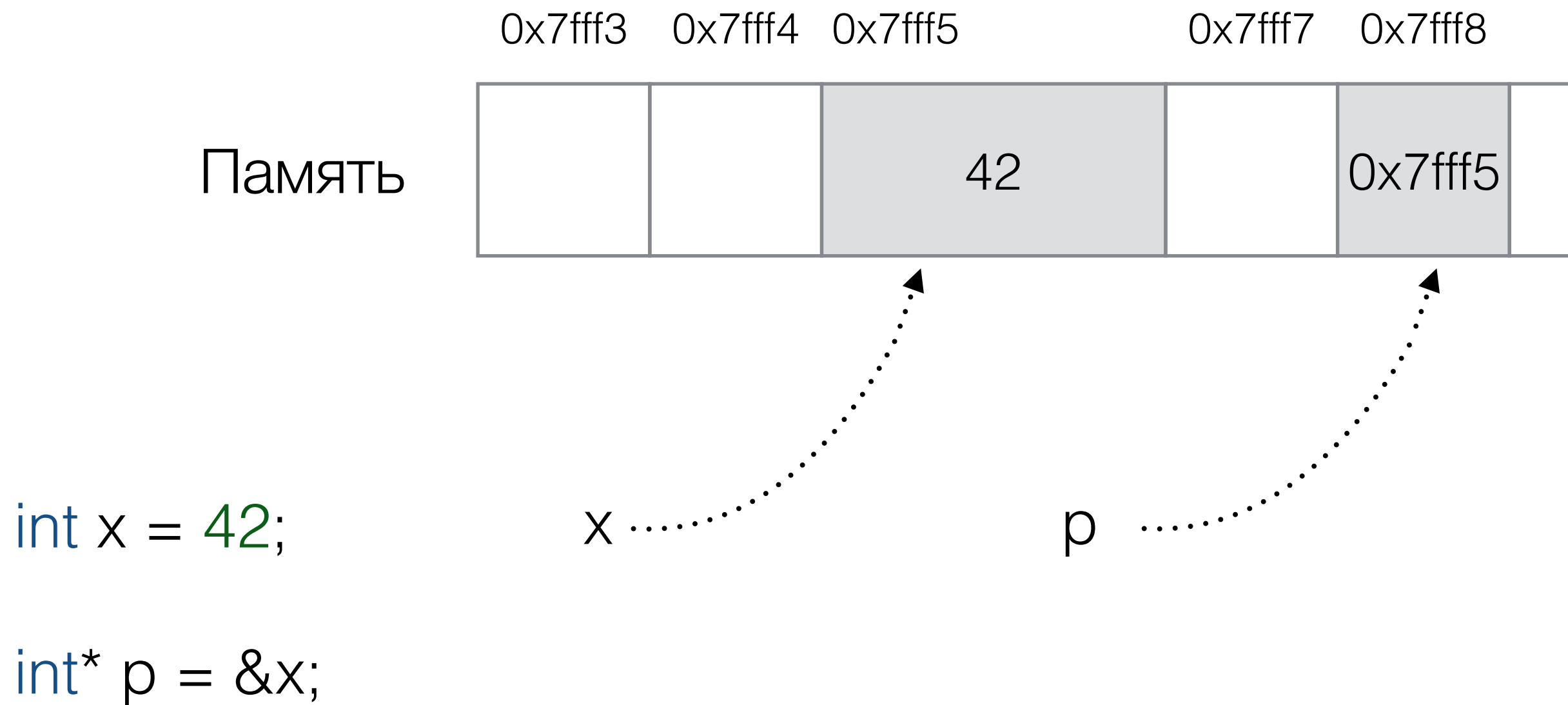
ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

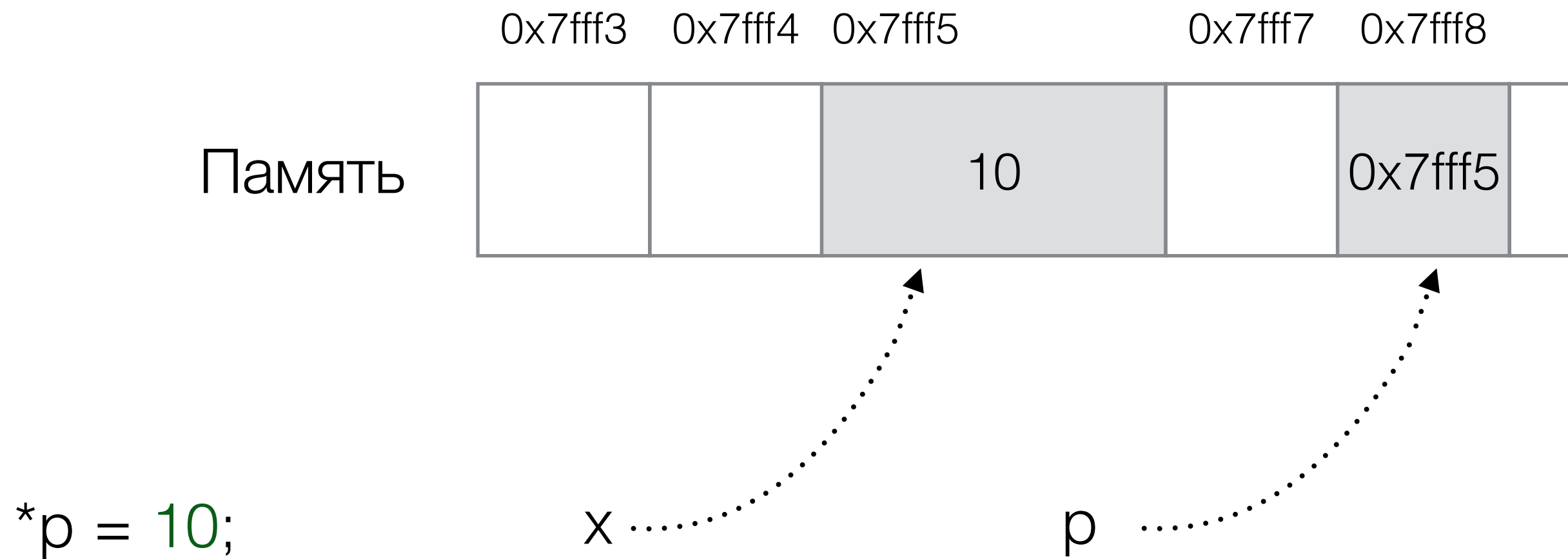
ОСНОВНЫЕ ТИПЫ ДАННЫХ

- Размер основных встроенных типов зависит от архитектуры
- `sizeof()`
- `#include <climits>` и `#include <float>`
- Тип литералов по умолчанию (`5`, `5L`, `5UL`, `5.`, `05`, `0x5`)
- Сравнение чисел с плавающей точкой

Указатели



Указатели



Ссылки

```
int x = 42;
```

```
int& r = x;
```

```
r = 10;
```

```
cout << x << endl;
```

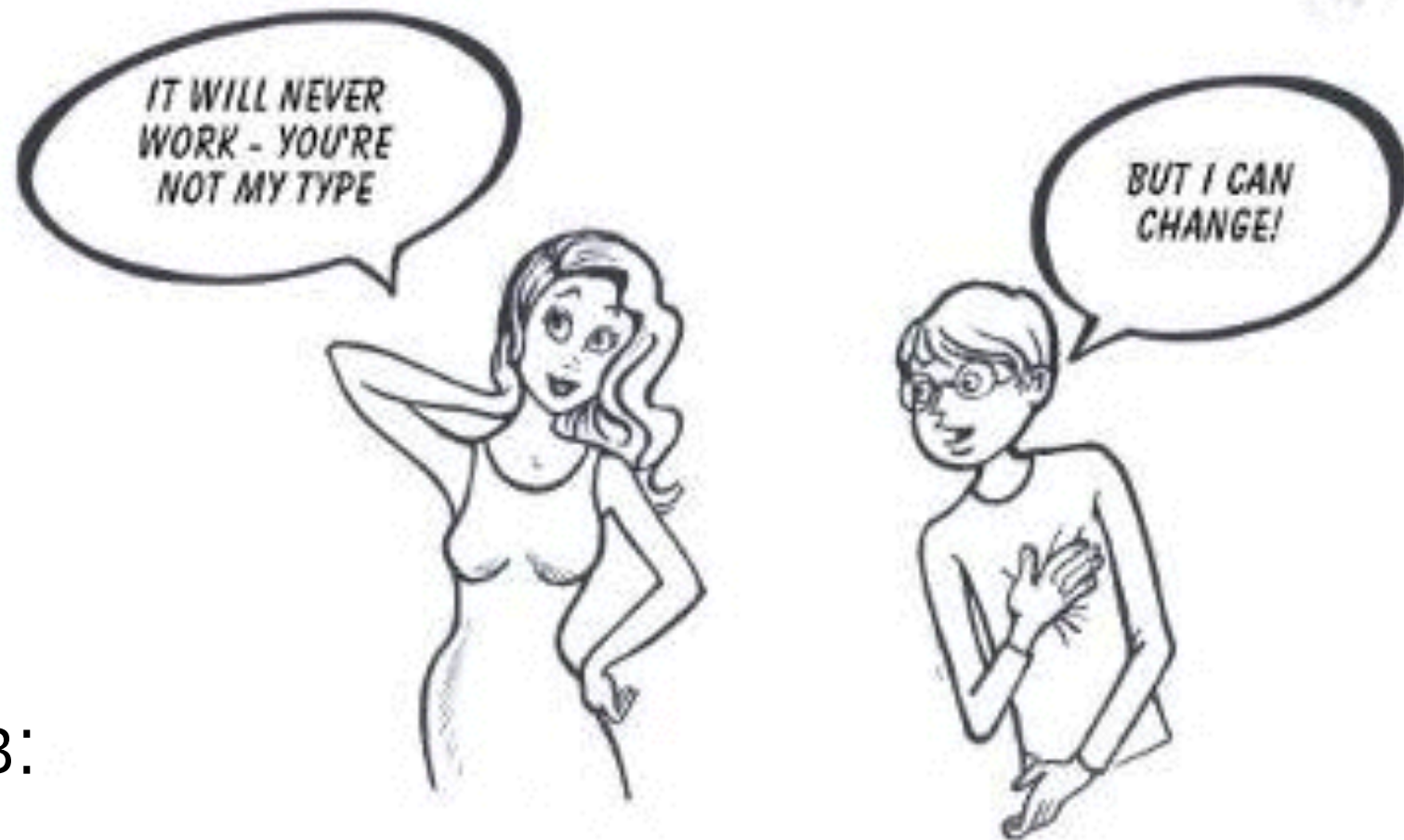

Преобразования типов данных

- Неявное в выражениях
- Неявное при передаче аргументов
- Явное приведение типов:

в стиле C: `new_type()`

в стиле C++: `static_cast<new_type>()`, `dynamic_cast<new_type>()`,

`reinterpret_cast<new_type>()`, `const_cast<new_type>()`



Вопрос

Какого типа будет x?

```
int y = 10;
```

```
int x;
```

```
x = static_cast<double>(y);
```

Производные типы данных

Массивы

Массив - тип данных, содержащий несколько значений одного типа:

```
double numbers[3] = {1, 2.1, 2.7};
```

```
numbers[0] = 13.4;
```

```
TypeName arrayName[arraySize];
```

Строки в стиле C

Строка в стиле C - массив символов, заканчивающийся символом `'\0'`:

```
char dog[5] = {'m', 'o', 'g', 'g', '\0'}; // OK
```

```
char cat[5] = {'b', 'e', 't', 't', 'y'}; // NOT OK
```

```
char bat[5] = "mogg"; // OK
```

```
char* frog = "wogg"; // OK
```

Производные типы данных

Особенности:

- Размер должен быть известен на этапе компиляции
- Строка в стиле C должна завершаться терминирующим нулём — `'\0'` (`strlen()`, `std::cout <<`)
- Выход за пределы массива — неопределенное поведение (`undefined behaviour`)

Ключевое слово `const`

Ключевое слово `const` — одно из самых многозначных в C++

Значение 1: константные данные

```
const double PI = 3.14;
```

```
const double* pPI = &PI;
```

```
int const MONTHS_COUNT = 12;
```

```
const char NAME[11] = "Mr. Meyers";
```

Создают переменную, значение которой изменить нельзя (есть правда - `const_cast<new_type>()`).

Про lvalue и rvalue

Каждое выражение C++ является значением **lvalue** либо значением **rvalue**.

Значение **lvalue** — объект, существующий за пределами одного выражения (объект с именем).

Значение **rvalue** — это временное значение, которое не сохраняется за пределами выражения, в котором оно используется.

Вопрос

Какие из этих выражений корректны?

```
int x = 3 + 4;
```

```
cout << x << endl;
```

```
7 = x;
```

```
x * 7 = 49;
```

Унарные операторы



+



-

Вопрос

Какие из этих выражений корректны?

`int x;`

`-x = 49;`

`+x = 49;`

Бинарные операторы

+

-

*

/

%

Операторы с присваиванием

=

+=

-=

*=

/=

%=

Вопрос

Что выведется на экран?

```
int x = 10;
```

```
int y;
```

```
int z;
```

```
z = y = x;
```

```
z += y += x;
```

```
cout << x << " " << y << " " << z << endl;
```

Операторы сравнения

`==`

`!=`

`<`

`>`

`<=`

`>=`

Логические операторы

A light gray rounded square button with a subtle gradient and a thin shadow, containing the text '&&' in black.

&&

A light gray rounded square button with a subtle gradient and a thin shadow, containing the text '||' in black.

||

Ленивые

A light gray rounded square button with a subtle gradient and a thin shadow, containing the text '!' in black.

!

Вопрос

Что выведется на экран?

```
bool f() {  
    cout << "f()" << endl;  
    return true;  
}  
  
...  
  
int x = 10;  
  
if (x > 5 || f()) {  
    cout << "Yeah" << endl;  
}
```

Побитовые операторы

$\&$

$|$

\ll

\gg

\sim

\wedge

Побитовые операторы с присваиванием

$\&=$

$|=$

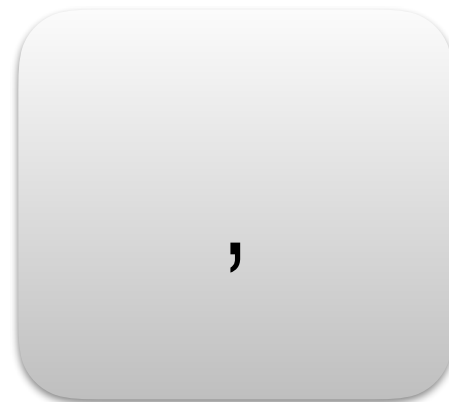
$<<=$

$>>=$

$\sim=$

$\wedge=$

Оператор запятая



Вопрос

Что выведется на экран?

...

```
double a;
```

```
a = 1,2;
```

```
cout << a << endl;
```

...

Вопрос

Что выведется на экран?

...

```
double a;
```

```
a = (1, 2);
```

```
cout << a << endl;
```

...

Тернарный оператор

? :

<условие> ? оператор1 : оператор2;

```
int max = (a > b) ? a : b;
```

Операторы инкремента

Постфиксный ++

Постфиксный --

Префиксный ++

Префиксный --

Вопрос

Что выведется на экран?

...

```
int x = 10;
```

```
cout << x++ << endl;
```

```
cout << ++x << endl;
```

...

Вопрос

Что из этого корректно?

```
int i = 10;
```

```
i++ = 50;
```

```
++i = 50;
```


Вопрос

Что из этого корректно?

```
int i = 10;
```

```
++++i;
```

```
i++++;
```

```
++i++;
```

Приоритеты операторов

1	(), постфиксные ++, --
2	!, ~, унарные +, -, префиксные +, --
3	*, /, %
4	+, -
5	<<, >>
6	<, >, <=, >=
7	!=, ==

8	&
9	^
10	
11	&&
12	
13	?:, =, +=, -=, *=, /=, %=
14	,

Приоритеты операторов

Некоторые комментарии:

- Арифметические бинарные операторы — приоритеты как в математике
- Постфиксные операторы имеют больший приоритет, чем префиксные
- Если есть сомнения — лучше ставить скобки

Инструкции (statements)

- if
- for
- while
- do while
- switch
- goto



Инструкция if

```
if (<условие>) {
```

```
    оператор;
```

```
    оператор;
```

```
    ...
```

```
} else {
```

```
    ...
```

```
}
```

Инструкция if

```
int applesCount = 5;  
  
if (applesCount > EnoughCountForPie()) {  
    std::cout << "We have enough apples\n";  
  
} else {  
    std::cout << "We need more apples\n";  
  
}
```

Инструкция while

```
while (<условие>) {  
    оператор;  
    оператор;  
    ...  
}
```

Инструкция while

```
std::vector<int> numbers = {1, -1, 23, -42, 42}
```

```
int i = 0;
```

```
while (i < numbers.size() && numbers[i] != 23) {
```

```
    ++i;
```

```
}
```


Инструкция do while

```
do {
```

```
    оператор;
```

```
    оператор;
```

```
    ...
```

```
} while (<условие>);
```

Инструкция do while

```
std::vector<int> numbers = {1, -1, 23, -42, 42}
```

```
int i = 0;
```

```
do {
```

```
    ++i;
```

```
} while (i < numbers.size() && numbers[i] != 23)
```

Инструкция for

```
for (<до начала>; <условие>; <после каждой итерации>) {  
    оператор;  
  
    оператор;  
  
    ...  
}
```

Инструкция for

```
std::vector<int> a = {1, -1, 23, -42, 42};
```

```
std::vector<int> b = {0, 10, 2};
```

```
for (int i = 0, j = b.size(); i < a.size() && j > -1; ++i, --j) {
```

```
    ...
```

```
}
```

Ключевые слова `break` и `continue`

- `break` — выход из цикла
- `continue` — переход на следующую итерацию цикла

Инструкция for

```
std::vector<int> a = {1, -1, 23, -42, 42};
```

```
std::vector<int> b = {0, 10, 2};
```

```
for (int i = 0, j = b.size(); i < a.size() && j > -1; ++i, --j) {
```

```
    ...
```

```
    if (a[i] == b[j]) {
```

```
        break;
```

```
    }
```

```
    ...
```

```
}
```

Инструкция goto

L:

goto L;

Инструкция switch

```
switch (<переменная простого типа>) {
```

```
    case value1:
```

```
        оператор;
```

```
        оператор;
```

```
    case value2:
```

```
        оператор;
```

```
        ...
```

```
    ...
```

```
    default:
```

```
        оператор;
```

```
        ...
```

```
}
```


Инструкция switch

```
switch (x) {  
    case 0:  
        x += 10;  
        break;  
    case 1:  
        x += 20;  
        break;  
    default:  
        x = -1;  
        std::cout << "Strange value\n";  
}
```

Пустая инструкция

;

Вопрос

Что выведется на экран?

```
int a = 0;
```

```
while (a < 10);
```

```
    std::cout << ++a << std::endl;
```

Вопросы?

Домашнее задание

- Прочитать главы 3-6 в книге Стивена Прата «Язык программирования C++»
- Дорешать задачи с семинара

Домашнее задание

- Завести репозиторий на github.com с именем `cpp_hse_spring_2016`
- Решения задач выкладывать в репозиторий в папки `seminars/seminar_01/`, `seminars/seminar_02/`, ...
- Прислать на почту курса ссылку на репозиторий

Материалы в помощь

- <https://progit.org> - онлайн книга про git
- <http://git-scm.com/book/ru/v2> - русский перевод