Объектно-ориентированное программирование

Лекция 13

Основные термины

- *Предметная область* – это набор понятий, представляющих важные аспекты решаемой задачи
- *Объект* – это конструкция программы, использующая термины реального мира и абстракции (включает состояние и поведение)
- *Класс* объединяет объекты с общими свойствами и поведением

Основные термины

- *Признаки* – это видимые извне свойства класса
- *Поведение* – это действия, выполняемые объектом в ответ на сообщение или изменение состояния, (то, что объект делает)
- *Конструкторы* – это методы, используемые для инициализации объектов в процессе их конкретизации

Три базовых понятия

- *Инкапсуляция*
- *Наследование*
- *Полиморфизм*

Инкапсуляция – это объектно-ориентированная характеристика модульности (защищенная от внешних воздействий часть программы).

- *Интерфейс* представляет собой список служб, указывающий какие запросы внешние объекты могут посылать данному объекту, скрывая его конкретную реализацию
- *Уровни доступа*:
  - Общедоступный
  - Защищенный
  - Частный

- *Абстрактный тип данных* – это набор данных и операций над ними
- *Типы* определяют виды переменных, которые можно использовать в программе, определяя область допустимых значений, которые может принимать переменная

Три базовых понятия

- *Инкапсуляция*
- *Наследование*
- *Полиморфизм*

Наследование– это механизм, который дает возможность создавать новый класс на основе определения уже существующего класса. Все методы и свойства интерфейса ранее существовавшего класса автоматически появляются в интерфейсе наследника.

- Передача (делегирование) функций – это способ обработки, с помощью которого один объект передает сообщение другому с целью выполнения некоторого запроса
- Наследственная иерархия – это древовидное отображение отношений, которые устанавливаются между классами в результате наследования

- *Потомственный класс* (потомок) – это класс, созданный в результате наследования (подкласс, производный класс, субкласс)
- *Родительский класс* – это класс, непосредственным наследником которого является потомственный класс (суперкласс, надкласс)

Три базовых понятия

- *Инкапсуляция*
- *Наследование*
- *Полиморфизм*

Полиморфизм – это такое состояние, когда одно имя может обозначать различное поведение, то есть представлять различный код, выбранный автоматическим механизмом.

- *Полиморфная переменная* – это переменная, которая может хранить данные различных типов

СРЕДСТВА ЯЗЫКА С++

Состав языка

Алфавит языка C++

- Множество символов языка C включает:
  - прописные и строчные буквы латинского алфавита и знак подчеркивания
  - арабские цифры от 0 до 9, шестнадцатиричные цифры от A до F
  - специальные знаки: { } , . ; : ? ! ' " | / \ ~ _ ^ ( ) [ ] < > # - = + * % &
  - символы пробела, табуляции, перевода строки
- Остальные символы могут быть использованы только в символьных строках, символьных константах и комментариях.
  - Язык C++ различает большие и маленькие буквы, таким образом, name и Name – ***разные идентификаторы***

Список ключевых слов C++

alignas (*начиная с C++11*) alignof (*начиная с C++11*) and and_eq asm auto (1) bitand bitor bool break case catch char char16_t (*начиная с C++11*) char32_t (*начиная с C++11*) class compl const constexpr (*начиная с C++11*) const_cast continue decltype (*начиная с C++11*) default delete do double dynamic_cast else	enum explicit export extern false float for friend goto if inline int long mutable namespace new noexcept (*начиная с C++11*) not not_eq nullptr (*начиная с C++11*) operator or or_eq private protected public register reinterpret_cast	return short signed sizeof static static_assert (*начиная с C++11*) static_cast struct switch template this thread_local (*начиная с C++11*) throw true try typedef typeid typename union unsigned using virtual void volatile wchar_t while xor xor_eq

Типы данных языка C++

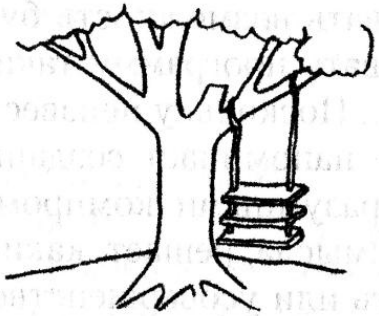
Имя	Размер	Представляемые значения	Диапазон
bool	1 байт	логические	false, true
(signed) char	1 байт	символы , целые числа	от –128 до 127
wchar_t	2 байта	символы Unicode	от 0 до 65535
(signed) short int	2 байта	целые числа	от -32768 до 32767
(signed) int	зависит от реализации (в последних компиляторах обычно 4 байта)	целые числа	
(signed) long int	4 байта	целые числа	от -2147483648 до 2147483647
(signed) long long int (signed) __int64 (MS)	8 байт	целые числа	от –9,223,372,036, 854,775,808 до 9,223,372,036, 854,775,807
unsigned char	1 байт	символы , целые числа	от 0 до 255
unsigned short int	2 байта	целые числа	0 до 65535
unsigned int	зависит от реализации (в последних компиляторах обычно 4 байта)	целые числа	
unsigned long int	4 байта	целые числа	от 0 до 4294967295
(unsigned) long long int (unsigned) __int64 (MS)	8 байт	целые числа	от 0 до 18,446,744, 073,709,551,615
float	4 байта	вещественные числа	от 1.175494351e–38 до 3.402823466e+38
double	8 байт	вещественные числа	от 2.2250738585072014e–308 до 1.7976931348623158e+308
long double	зависит от реализации	вещественные числа	

Знак операции	Наименование	Ассоциативность
::	Разрешение области видимости	*Слева направо*
( ) [ ] . -> *++ --* *static_cast* *dynamic_cast* *reinterpret_cast* *const_cast*	Первичные Постфиксный инкремент и декремент Преобразование с проверкой во время компиляции Преобразование с проверкой во время выполнения Преобразование без проверки Константное преобразование	*Слева направо*
*- ~ ! * &* *++ --* *sizeof* *(<тип>)<выражение>* *new* *delete*	Унарные Префиксный инкремент и декремент Вычисление размера Приведение типа Выделение памяти Освобождение памяти	*Справа налево*
. ->**	Выбор члена класса	*Слева направо*
** / %*	Мультипликативные	*Слева направо*
+ -	Аддитивные	*Слева направо*
<< >>	Сдвиг	*Слева направо*
< > <= >=	Отношение	*Слева направо*
== !=	Отношение	*Слева направо*
&	Поразрядное И	*Слева направо*
^	Поразрядное исключающее ИЛИ	*Слева направо*
|	Поразрядное ИЛИ	*Слева направо*
&&	Логическое И	*Слева направо*
||	Логическое ИЛИ	*Слева направо*
? :	Условная операция	*Справа налево*
*= *= /= %= += -=* *<<= >>= &= ^= |=*	Простое и составное присваивания	*Справа налево*
throw	Генерация исключения	*Слева направо*
,	Операция последовательного вычисления	*Слева направо*

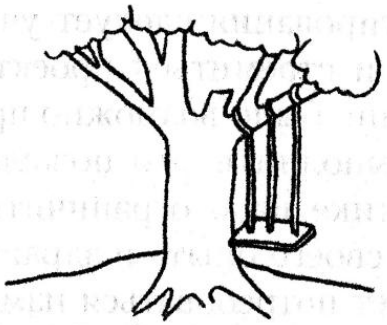
Области действия имен

- Блок
- Файл
- Функция
- Прототип функции
- Класс
- Поименованная область

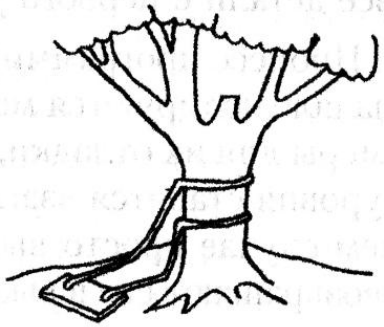
Область видимости совпадает с областью действия за исключением ситуации, когда во вложенном блоке описана переменная с таким же именем. В этом случае внешняя переменная во вложенном блоке невидима, хотя он и входит в ее область действия.



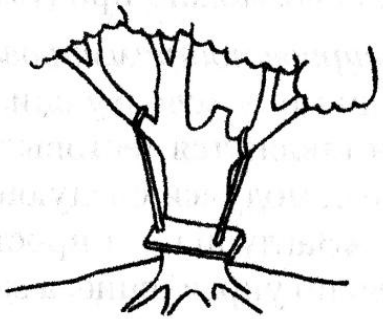
Как было предложено
организатором разработки



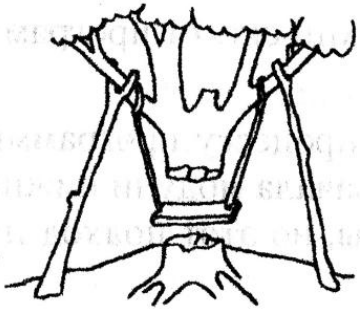
Как было описано
в техническом задании



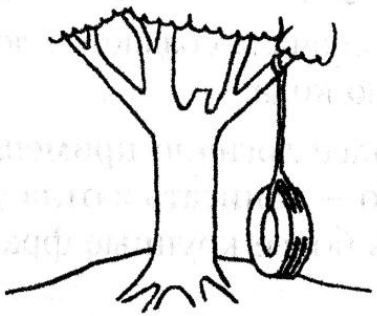
Как было спроектировано
ведущим системным специалистом



Как было реализовано
программистами



Как было внедрено



Что хотел пользователь

Цикл лекций подготовлен в 2014/2015 уч. году Кузнецовым Игорем Ростиславовичем, доцентом кафедры радиоэлектронных средств Санкт-Петербургского Государственного электротехнического университета «ЛЭТИ»

Прочитан в дисциплине «Информатика»

© Кузнецов И.Р.