

C++ course

name: Victoria

e-mail: kewtree1408@gmail.com

C++ course

source: <https://bitbucket.org/Kewtree/coursecpp/src>

login: coursecpp

password: 12345

Plan 8

- Struct (ex: employee.cpp)
- OOP: 3 principles
- OOP. Encapsulation
- First class. Definitions
- Private/Public
- Constructor/Destructor

Plan 8

- Types of Constructor
- Method-overload
- Method-definition without class
- Class in Memory
- Static data
- Method-constant

Struct

- Example: `employee.cpp`

ООР

- Encapsulation (Инкапсуляция)
- Inheritance (Наследование)
- Polimorph (Полиморфизм)

Encapsulation

- Данные скрыты от внешнего воздействия, что защищает их от случайного изменения. Поэтому говорят, что методы и данные инкапсулированы (скрыты).

Definitions

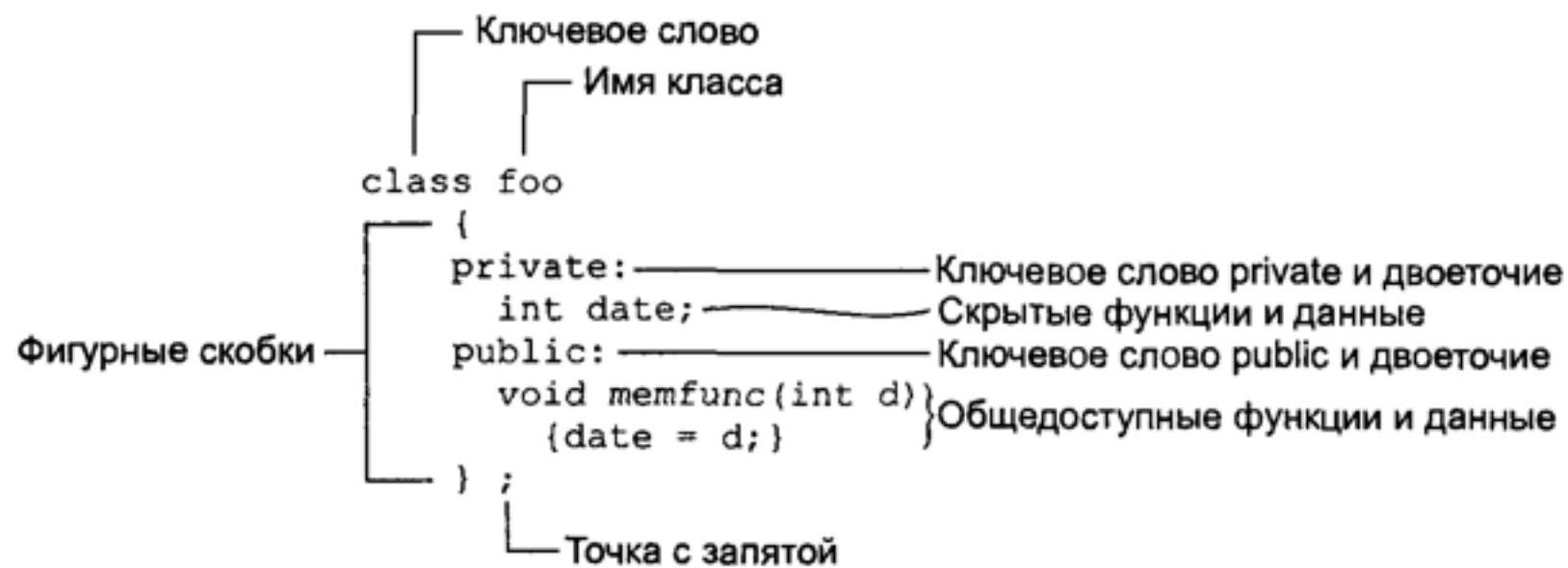
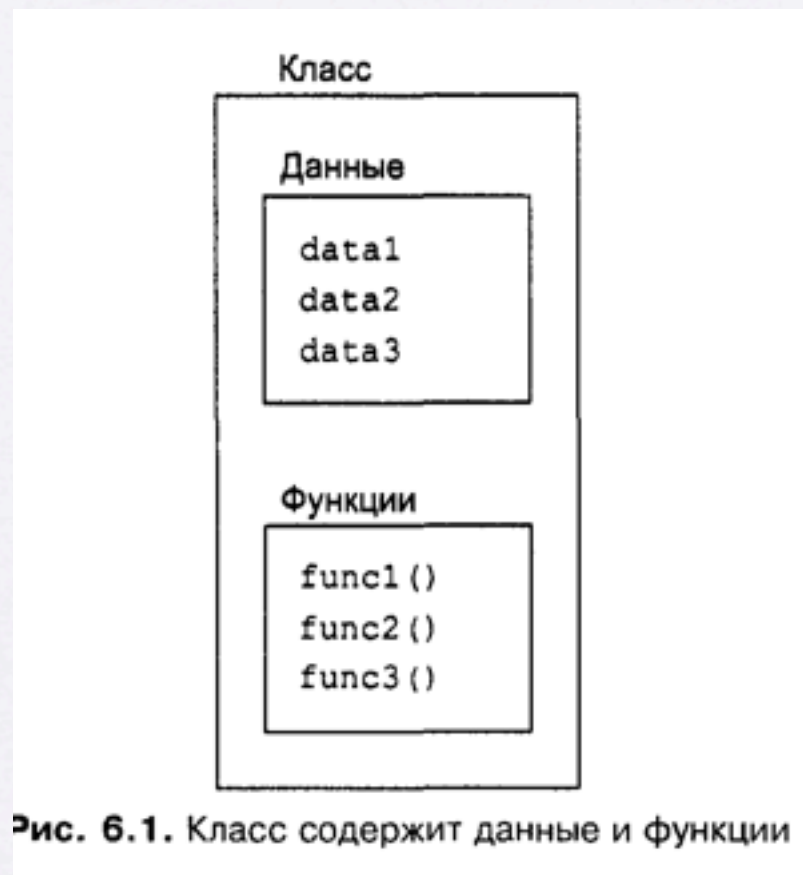


Рис. 6.3. Синтаксис определения класса

Simple examples

- first.cpp
- monitor.cpp

Private/public



Constructor

- Выполняется автоматически в момент создания объекта
- Имя конструктора совпадает с именем класса
- Конструктор не имеет возвращающего значения
- У типа нет возвращаемого типа

Destructor

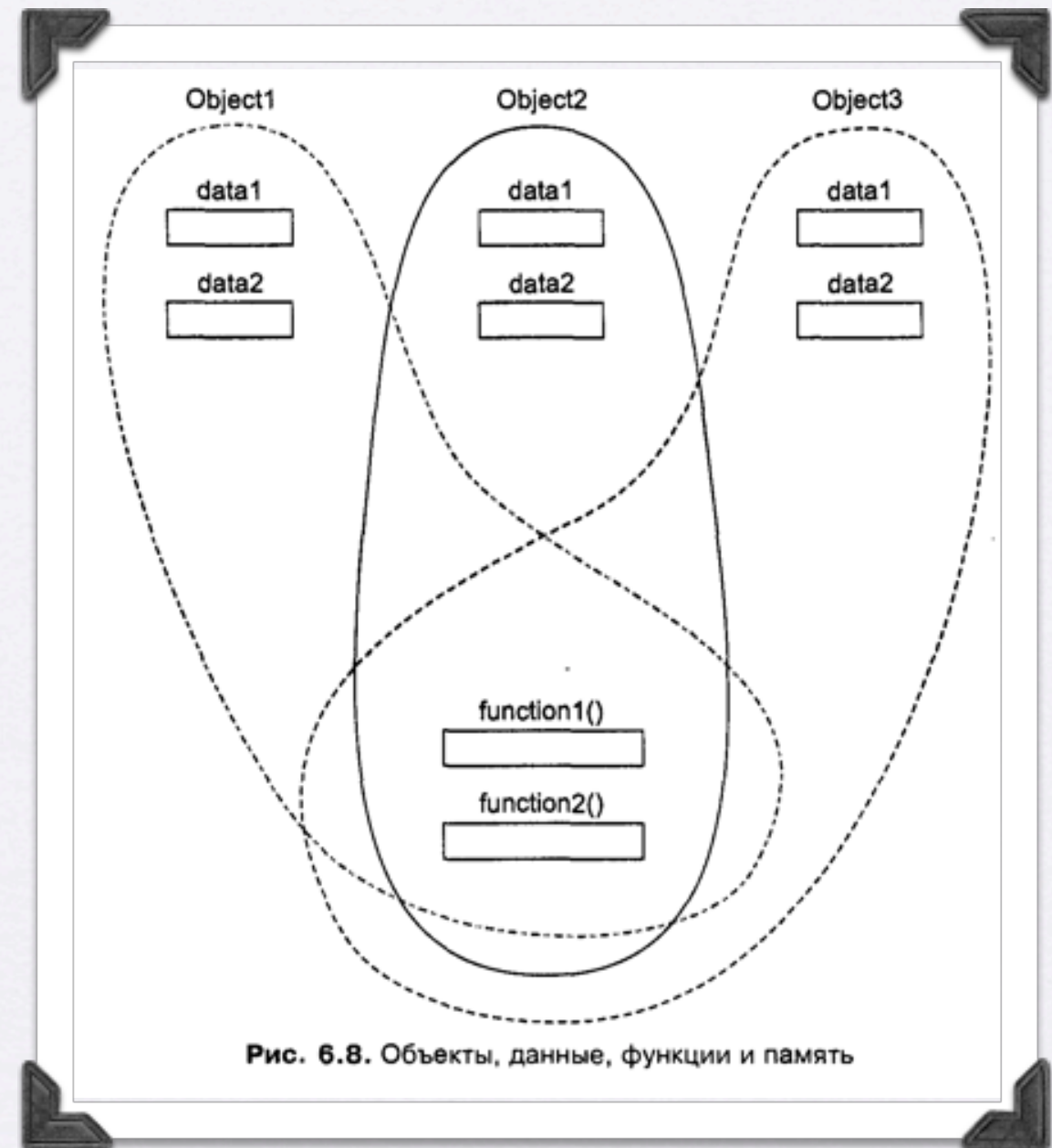
- Вызывается автоматически при удалении объекта
- Совпадает с именем класса, но предваряется ~
- Не имеет возвращаемого значения и возвращаемого типа

Constructors

- Конструкторы по умолчанию
- Перегруженные конструкторы
- Конструктор копирования
- Конструктор копирования по умолчанию
`X(X& x);`

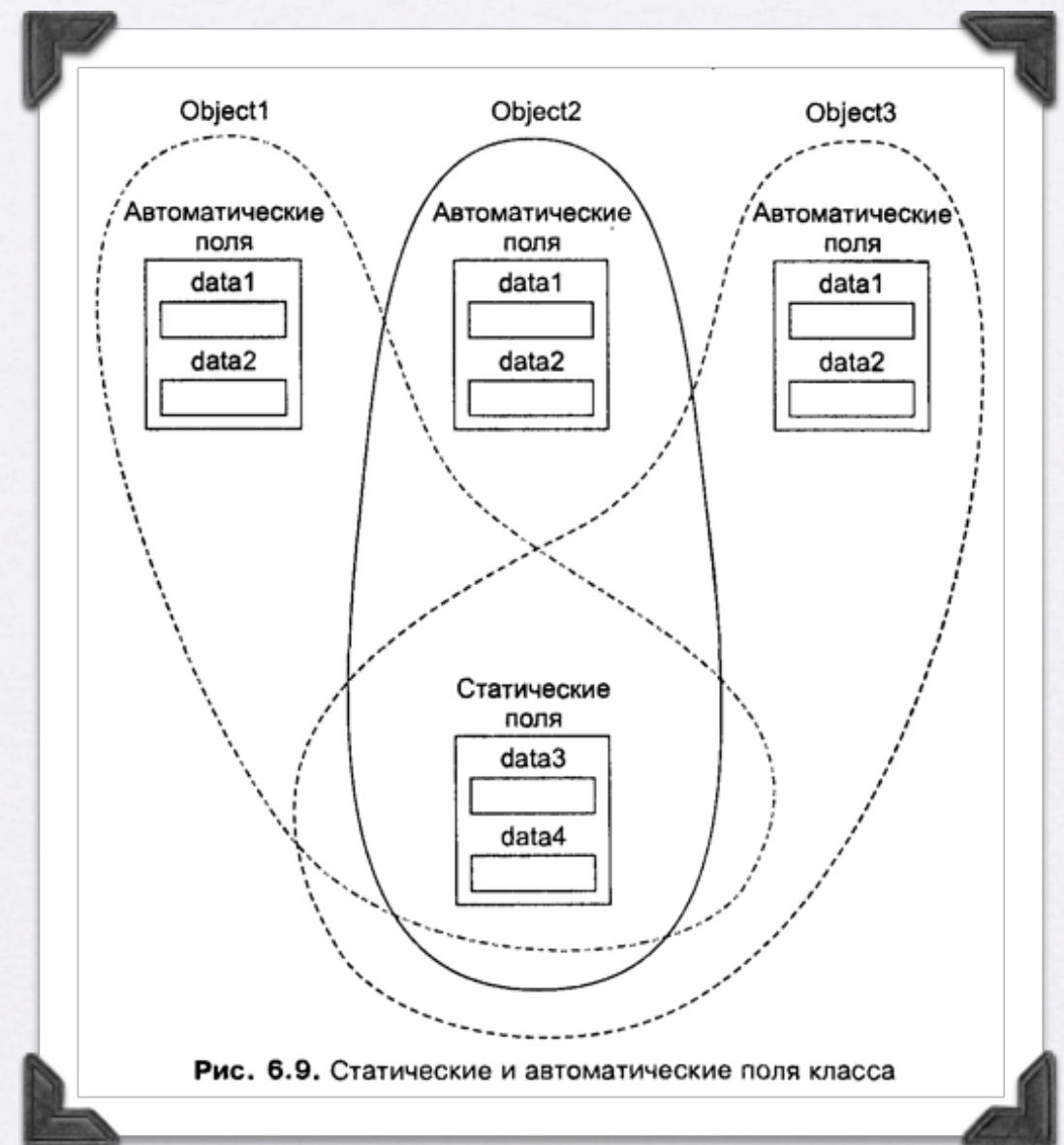
Class in Memory

- Объекты разные - данные разные
- Функции одни и те же



Static data

- Статические поля относятся ко всему классу, а не к отдельным объектам



Method-constant

- Example: distance.cpp
- Зачем?
 - Предотвращает неправильное использование метода
 - Указывает читателю на то, что нет необходимости менять значение полей объекта

Test

- Complex class
- Data: re, im
- Methods: show, add, get, sub
- Создать 3 объекта с обычными конструкторами и копирующим конструктором. Протестировать все написанные выше методы.

Home-task

- Создать класс `Int`, имитирующий стандартный тип `int`. Единственное поле класса должно иметь тип `int`. Создать методы, которые будут устанавливать значение поля равным 0, инициализировать его целым значением, выводить значение поля на экран и складывать два значения типа `Int`. Написать программу, в которой будут созданы три объекта класса `Int`, два из которых будут инициализированы. Сложить два инициализированных объекта, присвоить результат третьему, а затем отобразить результат на стандартном выводе.
- Создать класс, одно из полей которого хранит “порядковый номер” объекта, то есть для первого созданного объекта значение этого поля равно 1, для второго созданного объекта значение равно 2 и т.д. Для того, чтобы создать такое поле, необходимо иметь еще одно поле, в которое будет записываться количество созданных объектов класса (это означает, что последнее поле должно относиться не к отдельным объектам класса, а ко всему классу в целом. Каждый раз при описании нового объекта конструктор может получить значение этого поля и в соответствии с ним назначить объекту индивидуальный порядковый номер. В класс следует включить метод, который будет выводить на экран порядковый номер этого объекта. Создать функцию `main()`, в которой будут созданы три объекта, и каждый объект выведет на экран свой порядковый номер.
- Создать класс `Fraction`. Данные представлены двумя полями: числитель и знаменатель. Методы класса должны получать от пользователя значение числителя и знаменателя дроби в форме $3/5$ и выводить значение дроби в этом же формате. Кроме того, должен быть разработан метод, складывающий значения двух дробей. Написать функцию `main()`, которая циклически запрашивает у пользователя ввод пары дробей, затем складывает их и выводит результат на экран. После каждой такой операции программа должна спрашивать пользователя, следует ли продолжать цикл.
- Создать класс `Employee`-работники. В классе должны присутствовать поля: имя, фамилия, год рождения, зарплата. Методы: отобразить информацию о Работнике, возможность изменять поля и получать значения их полей. В главной функции создать массив Работников, заполнить их либо вручную, либо случайными значениями. После создания массива, написать функцию, которая возвращает сотрудника с максимальной зарплатой. Написать функцию, которая по входному массиву сортирует сотрудников по дате рождения. Сортировку написать свою (нельзя использовать `qsort` из namespace `std`)