

Программа экзамена

- 1. Основы объектно-ориентированного программирования:** абстракция, инкапсуляция, полиморфизм, наследование; классы, структуры и объекты; поля и методы (константные, статические); модификаторы доступа; ключевое слово *mutable*<sup>1</sup>. ед способ инициализации константных полей
- 2. Конструкторы и деструкторы:** конструктор по умолчанию; конструктор с параметрами; и полей-ссылок преобразование; конструктор копирования; ключевое слово *explicit*; список инициализации; делегирующие конструкторы; деструктор; конструкции *= default* и *= delete*; правило трех, uniform initialization (list initialization)<sup>2</sup> A a {}, A a {2.0}, return {p, q}
- 3. Перегрузка операторов:** оператор присваивания<sup>1</sup>; правила перегрузки операторов; операторы как члены класса; операторы как внешние функции; приведения типов в стиле C++ (*static\_cast*, *dynamic\_cast*, *const\_cast*, *reinterpret\_cast*); overload A(1)...3, over A(0.0...1) L... перегрузка операторов приведения типа; дружественные функции и классы.
- 4. Шаблоны:** статический полиморфизм; шаблоны функций; вывод типов шаблонных аргументов; шаблоны классов; инстанцирование шаблонов; специализация и частичная специализация шаблонов; параметры шаблона, не являющиеся типами; ключевое слово *auto* (C++11); variadic templates (шаблоны с переменным числом аргументов)<sup>2</sup>.
- 5. Наследование I:** protect режимы наследования; срезка; работа с производными классами через указатель или ссылку на базовый; порядок вызова конструкторов и деструкторов при наследовании; замещение методов базового класса.
- 6. Наследование II:** динамический полиморфизм; виртуальные методы; ключевые слова *override* и *final*; чисто виртуальные методы; абстрактные классы; таблицы виртуальных функций.
- 7.1. Наследование III:** множественное наследование; проблема ромбовидного наследования; виртуальное наследование; порядок вызова конструкторов и деструкторов при виртуальном наследовании.
- 7.2. Исключения I:** обработка ошибок, проблемы; ключевое слово *throw*; сворачивание стека; *try-catch* блок; ловля исключений по ссылке и по значению; исключения в конструкторах и деструкторах.
- 8. Исключения II:** преобразования типов при выборе блока *catch*; библиотека исключений C++ (*std::exception*, *std::runtime\_error*, ...); *std::bad\_alloc*, не бросающий *new*; спецификация исключений (*noexcept*); гарантии безопасности исключений.
- 9. Геометрия на плоскости:** точка, вектор, прямая, луч, отрезок, многоугольник; взаимные расположения и расстояния; выпуклые многоугольники, выпуклые оболочки; алгоритм Джарвиса (gift wrapping); алгоритм Грэхема.
- 10. Семантика перемещения:** lvalue и rvalue ссылки; перегрузка функций по типу ссылки; перемещающий конструктор и оператор присваивания; правило пяти; *std::move*; copy/move-and-swap идиома; copy elision, RVO, NRVO.
- 11. Умные указатели:** RAII; *std::auto\_ptr*, *std::unique\_ptr*, *std::shared\_ptr*, *std::weak\_ptr*; *std::make\_unique*, *std::make\_shared*.
- 12. Последовательные контейнеры:** *std::array*, *std::vector*, *std::deque*, *std::list*, *std::forward\_list*
- 13. Итераторы:** концепция итераторов; категории итераторов; *std::iterator\_traits*; range-based for; *std::initializer\_list*
- 14.1. Адаптеры:** контейнерные адаптеры; *std::stack*, *std::queue*, *std::priority\_queue*; итераторные адаптеры; *std::reverse\_iterator*, *std::insert\_iterator*, *std::back\_insert\_iterator*, *std::front\_insert\_iterator*
- 14.2. Ассоциативные контейнеры:** *std::map*, *std::set*, *std::multimap*, *std::multiset*, unordered- версии; structured bindings
- 15. Функторы:** функции, указатели на функции, перегрузка функционального вызова (круглые скобки); фунctors в стандартной библиотеке C++; лямбда выражения; *std::function*
- 16. Алгоритмы стандартной библиотеки<sup>1</sup>:** модифицирующие алгоритмы, немодифицирующие алгоритмы, сортировки, алгоритмы упорядоченных последовательностей, алгоритмы для работы с пирамидой; erase-remove идиома.

<sup>1</sup> Обсуждалось на семинарах

<sup>2</sup> Обсуждалось на лекции 8