Дополнительные возможности классов

№ урока: 5 **Курс:** C++ Essential

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать и применять на практике inline-функции и методы; константные методы;
статические поля и методы; абстрактные классы и чисто виртуальные методы; дружественные классы, методы, функции; перегрузку операторов.

Изучив материал данного занятия, учащийся сможет

- Понимать, как работают inline-функции, определять, нужно ли использовать это ключевое слово в конкретной ситуации.
- Применять константные методы при защите данных класса, когда в методе не нужно их изменять.
- Уметь различать принцип работы статических и нестатических членов класса, определять на практике, что относится к свойствам всего класса, а не объекта в частности.
- Различать интерфейс от абстрактного класса, знать синтаксис чисто виртуальной функции, понимать, как они помогают выстраивать иерархию классов и управлять полиморфизмом.
- Понимать принцип работы дружественных классов и функций, видеть плохие стороны данной возможности и стараться применять только по мере надобности.
- Определять перегрузку базовых операторов (самых востребованных для перегрузки), отличать статический полиморфизм от динамического.

Содержание урока

1. Inline-функции и методы
2. Константные методы
3. Статические поля и методы
4. Абстрактные классы и чисто виртуальные методы
5. Дружественные классы, методы, функции
6. Перегрузка операторов

Резюме

- inline int sum(int a, int b){ return a+b; } – inline-функция (встраиваемая функция), подставляет свое тело функции во все места, где происходит вызов данной функции для повышения производительности. Является рекомендацией, компилятор может проигнорировать Ваше желание сделать функцию inline, если не посчитает это нужным.
- void foo() const; - объявление константного метода (только внутри класса можно так писать), запрещает модифицировать данные-члены класса внутри тела данного метода.
- static void set_vibr(uint vibr){ vibr = vibr; } … static uint vibr; - статический метод set_vibr() и статическое поле vibr. К ним можно обращаться через сам класс, а также через объект класса. Принадлежат всему классу, а не объекту. Статический метод может работать только со статическими полями. Нестатический метод может работать со статическим полем. Время жизни статических членов класса – время жизни всей программы в целом.

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- Абстрактный класс – класс, который содержит хотя бы одну чисто виртуальную функцию. Интерфейс – класс, который содержит только чисто виртуальные функции.
- virtual void foo() = 0; - синтаксис чисто виртуальной функции.
- Дружественные классы, методы, функции – им предоставляется доступ ко всем закрытым данным класса, который у себя объявил какую-то сущность как friend.
- class A{ int a; friend B; }; -классу B теперь доступно поле a класса A, которое является private полем.
- Перегрузка операторов – пример статического полиморфизма, позволяет переопределить поведение стандартных операторов (+,-,*,=,(),+=, <<, >>, [], &, и т.д.). Есть операторы, которые нельзя перегружать.
- Number& operator=(const Number& other); - перегрузка оператора присвоения
- Number operator+(const Number& other) const; - перегрузка оператора сложения
- friend ostream& operator<<(ostream &s, const Number& number); - перегрузка оператора вывода

Закрепление материала

- Что такое inline- функция? В каких случаях она может быть полезна? Какой недостаток у нее?
- Для чего нужны константные методы? Можно ли обойтись без них?
- Опишите принцип работы со статическими членами класса.
- В чем разница между абстрактным классом и интерфейсом? Каков синтаксис чисто виртуальной функции? Можно ли описать тело в чисто виртуальной функции? Целевое назначение чисто виртуальной функции?
- Опишите принцип работы дружественных классов, функций и методов. Является ли их использование хорошей практикой? Почему?
- Для чего нужна перегрузка операторов? Какие операторы можно перегружать?

Дополнительное задание

Задание
Разобрать реализацию остальных перегрузок операторов, которые не были рассмотрены на уроке.
Изучить «Правило трех». Подумайте, может ли виртуальная функция быть встроенной (inline)?

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Вновь пересмотрите медицинскую карточку, которую вы рассматривали на прошлом уроке. Добавьте поле PatientID, подумайте, как можно сделать так, чтобы для каждого пациента идентификатор был уникален и его нельзя было сломать извне. Какие интерфейсы можно выделить или добавить для Вашей системы? Перегрузите операторы выводов для классов, которые требуют форматированного вывода информации о себе. Добавьте const в методы, где не подразумевается изменение данных.

Задание 3
Зайдите на сайт MSDN.

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы» описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

https://en.cppreference.com/w/cpp/language/operators
https://msdn.microsoft.com/ru-ru/library/5tk49fh2.aspx

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com