Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

**РЕФЕРАТ**

По предмету “Основы алгоритмизации и программирования”

На тему: “Отличие перегрузки функции от переопределения функции”

Выполнил:

Студент 1 курса 9 группы

Павлович Ян Андреевич

Преподаватель: Белодед Н.И.

2023, Минск

Содержание

1. Введение
2. Основная часть
   1. Перезагрузка функции
      1. Определение понятия перегрузки функции.
      2. Примеры перегрузки функций на языке C++ и особенности использования
   2. Переопределение функции
      1. Определение понятия переопределения функции.
      2. Примеры переопределения функций на языке C++.
   3. Сравнение и отличия перегрузки и переопределения функций
3. Заключение
4. Источники

**Введение**

Функции являются одним из основных строительных блоков в программировании. Они представляют собой блоки кода, которые выполняют специфическую задачу, и могут использоваться многократно в программе. Функции обеспечивают четкую структуру кода, улучшают его читаемость и облегчают отладку. Все языки программирования предоставляют механизмы для определения функций, но они могут различаться в зависимости от конкретного языка.

Перегрузка методов и переопределение методов являются важными концепциями в объектно-ориентированном программировании, позволяющими разработчикам гибко использовать и изменять функциональность своих классов. Однако, хотя эти два понятия имеют общую задачу — изменение поведения метода, между ними есть существенные различия.

**Перегрузка функции**

Перегрузка функции - это концепция, которую можно легко обнаружить в языках программирования, таких как C++. Это возможность создать несколько функций с одинаковым именем, но с различными параметрами или типами аргументов. Это дает возможность использовать одно и то же имя функции для выполнения различных задач, учитывая типы и количество аргументов. Это значительно улучшает читаемость кода, делая его более организованным, и снижает вероятность ошибок, поскольку уменьшает количество уникальных имен функций, которые нужно запомнить и управлять. Перегрузка функций улучшает читаемость кода, упрощает его поддержку и делает его более гибким для изменений.

Особенностью перегрузки методов является использование разных сигнатур — наборов параметров, которые позволяют различать методы. Сигнатура метода это сочетание количества, типов и порядка параметров.

Зачем нужна перегрузка методов? Она позволяет создавать более удобные и гибкие интерфейсы для работы с классами. Например, если у нас есть класс, выполняющий математические операции, то перегрузка методов позволяет вызвать одну и ту же функциональность с разными параметрами. Например, метод можно вызвать как с двумя числами, так и с тремя.

Для определения перегруженных методов необходимо использовать такие же имена, но разные параметры. Компилятор самостоятельно будет выбирать нужный метод при вызове, основываясь на переданных аргументах и их типах.

Перегрузка методов — это возможность определить несколько методов с одним и тем же именем в одном классе, но с различным числом параметров или разными типами параметров. Таким образом, при вызове метода с определенными аргументами, компилятор будет выбирать метод с наиболее подходящими параметрами.

В языках программирования, поддерживающих перегрузку методов, различные параметры, которые могут быть перегружены, включают в себя:

* 1. Число параметров: Методы могут быть перегружены по количеству параметров. Например, класс может иметь несколько методов с одним и тем же именем, но с разным числом параметров. Таким образом, можно создать методы для разной логики выполнения в зависимости от передаваемых параметров.
  2. Типы параметров: Методы могут быть перегружены по типу параметров. Например, класс может иметь несколько методов с одним и тем же именем, но с разными типами параметров. Таким образом, можно создать методы, работающие с разными типами данных.
  3. Порядок параметров: Методы могут быть перегружены по порядку параметров. Например, класс может иметь несколько методов с одним и тем же именем, но с разным порядком параметров. Таким образом, можно создать методы, для которых порядок передачи аргументов имеет значение.

При перегрузке методов важно учесть, что тип возвращаемого значения не может использоваться для разрешения перегрузки методов. Компилятор выбирает метод на основе типа аргументов, а не типа возвращаемого значения.

**// Перегруженная функция add, принимающая два целочисленных аргумента**

**int add(int a, int b){**

**// Возвращает сумму двух целых чисел**

**return a + b;**

**}**

**// Перегруженная функция add, принимающая два аргумента типа double**

**double add(double a, double b){**

**// Возвращает сумму двух чисел типа double**

**return a + b;**

**}**

Этот пример демонстрирует, как можно использовать одно и то же имя функции для работы с различными типами данных. Это делает код более чистым и понятным, облегчая его чтение и понимание.

Особенности использования перегрузки функций для различных типов данных включают в себя возможность использовать одно и то же имя функции для работы с разнообразием типов данных. Это улучшает читаемость кода, упрощает его структуру и делает более удобным для модификации. Так, например, функция для сложения может быть перегружена для работы как с целыми числами, так и с вещественными числами, что обеспечивает большую гибкость в использовании этой функции. Тем не менее, важно помнить, что перегруженные функции должны отличаться по типу и/или количеству параметров, чтобы компилятор мог различать их.

**Переопределение функции**

При разработке программного кода на языке программирования часто бывает необходимость изменять поведение уже существующих методов в классе. Для этого используется механизм переопределения методов.

Переопределение функции, с другой стороны, - это механизм в C++, который позволяет дочернему классу предоставить свою реализацию функции, уже представленной в родительском классе. Это позволяет изменить поведение класса-наследника, сохраняя при этом общую структуру и логику класса-родителя. Это ключевой элемент объектно-ориентированного программирования и основа такого мощного концепта, как полиморфизм. Переопределение функций позволяет уточнить или изменить поведение метода для подкласса.

Ниже приведен пример переопределения функции на языке C++:

**// Базовый класс**

**class Base {**

**public:**

**// Виртуальная функция show в базовом классе**

**virtual void show() {**

**// Выводим сообщение "Base class"**

**cout << "Base class";**

**}**

**};**

**// Производный класс, наследуется от базового класса**

**class Derived: public Base {**

**public:**

**// Переопределение функции show в производном классе**

**void show() {**

**// Выводим сообщение "Derived class"**

**cout << "Derived class";**

**}**

**};**

В этом примере функция show была переопределена в производном классе Derived. Это означает, что при вызове функции show на объекте класса Derived, будет вызвана версия функции show, определенная в классе Derived, а не в базовом классе Base. Это позволяет нам изменить поведение или результат функции, не изменяя ее имя.

Синтаксически, перегрузка функции происходит, когда в одном и том же области определения создаются функции с одинаковыми именами, но разными списками параметров (разное количество параметров или разные типы данных). В свою очередь, переопределение функции связано с объектно-ориентированным программированием и используется, когда производный класс предоставляет свою реализацию функции, уже определенной в базовом классе.

Семантически, перегрузка и переопределение функций служат разным целям. Перегрузка функций позволяет функции работать с разными типами данных и выполнять различные задачи, что делает код более гибким. В то время как переопределение функций позволяет нам изменить поведение метода для подкласса, не изменяя ее имя. Это ключевой элемент объектно-ориентированного программирования и основа такого мощного концепта, как полиморфизм.

**Сравнение и отличия перегрузки и переопределения функций**

Перегрузка и переопределение функций являются основными техниками в программировании, которые обеспечивают гибкость и масштабируемость кода. Они оба позволяют использовать одно и то же имя функции для выполнения различных задач.

Однако, они различаются по своей основной цели и области применения. Перегрузка функций используется для создания нескольких функций с одним и тем же именем, но с разными параметрами или типами аргументов. Это улучшает читаемость кода, делая его более организованным. С другой стороны, переопределение функций позволяет дочернему классу предоставить свою реализацию функции, уже представленной в родительском классе. Это ключевой элемент объектно-ориентированного программирования и основа такого мощного концепта, как полиморфизм.

Перегрузка функций должна использоваться, когда вы хотите использовать одно и то же имя функции для выполнения различных задач. Это особенно полезно, когда функции выполняют схожие задачи, но с разными типами данных или параметрами.

Переопределение функций, с другой стороны, полезно в объектно-ориентированном программировании, когда вы хотите изменить поведение или функциональность метода, наследуемого от родительского класса. Это позволяет дочерним классам иметь собственные уникальные реализации функций, сохраняя при этом общую структуру и логику класса-родителя.

**Заключение**

Перегрузка методов — это возможность создавать методы с одним и тем же именем, но с разными параметрами. Переопределение методов — это возможность создавать методы с тем же именем и параметрами, но с разной реализацией в классе-наследнике.

Понимание различий между перегрузкой и переопределением функций является важным для эффективной разработки программного обеспечения на языке C++. Это позволяет создавать более гибкий и масштабируемый код, учитывая определенные требования и контексты. Кроме того, это обеспечивает большую чистоту и структурированность кода, что важно для поддержания и развития программного проекта в долгосрочной перспективе. Понимание этих концепций поможет вам писать более чистый, более эффективный и более управляемый код. Несмотря на то, что перегрузка и переопределение функций могут казаться сложными концепциями, они являются важными инструментами для любого разработчика и могут значительно улучшить качество и эффективность вашего кода.

Источники

1. Чем перегрузка методов отличается от переопределения методов. URL: <https://uchet-jkh.ru/i/cem-peregruzka-metodov-otlicaetsya-ot-pereopredeleniya-metodov/>
2. Перезагрузка и переопределение функций. URL: <https://cppforeach.wordpress.com/2014/06/18/overloading_vs_overriding/>
3. Чем отличается перегрузка от переопределения. URL: <https://itsobes.ru/JavaSobes/chem-otlichaetsia-peregruzka-ot-pereopredeleniia/>