Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«**Информационные технологии и автоматизированные системы**»**

направление подготовки: 09.03.04 – **«**Программная инженерия»

**Лабораторная работа №4.**

**“Простое наследование. Принцип подстановки”  
Вариант 10**

Выполнил студент гр. РИС-24-2б

Бартов Игорь Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

1. **Постановка задачи:**

1.Определить пользовательский класс.2.Определить в классеследующие конструкторы: без параметров, с параметрами, копирования. 3.Определить в класседеструктор.

4.Определитьв классекомпоненты-функции для просмотра и установки полей данных(селекторы и модификаторы).5.Перегрузить операцию присваивания.6.Перегрузить операции ввода и вывода объектов с помощью потоков.7.Определить производный класс.8.Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.9.Реализовать функции, получающиеи возвращающиеобъект базового класса. Продемонстрировать принцип подстановки.

Задача:

Базовый класс:

ТРОЙКА\_ЧИСЕЛ (TRIAD)

Первое\_число (first) -int

Второе\_число (second) –int

Третье\_число (third) -int

Определить методы изменения полей и увеличения полей на 1. Создать производный класс DATE с полями год, месяц и число.Переопределить методы увеличения полей на 1 и определить метод увеличения даты на n дней

1. **Блок-схема:**
2. **Контрольные вопросы:**

 **Для чего используется наследование?**  
Для повторного использования кода и создания иерархии.

 **Как наследуются public компоненты?**  
Как public.

 **Как наследуются private компоненты?**  
Не наследуются.

 **Как наследуются protected компоненты?**  
Как protected.

 **Как описывается производный класс?**  
class Derived : public Base {}.

 **Наследуются ли конструкторы?**  
Нет, но можно вызывать их в списке инициализации.

 **Наследуются ли деструкторы?**  
Нет, но базовый вызывается автоматически.

 **Порядок конструирования производных классов?**  
Сначала базовый, затем производный.

 **Порядок уничтожения производных классов?**  
Сначала производный, затем базовый.

 **Что такое виртуальные функции и позднее связывание?**  
Виртуальные функции позволяют полиморфизм, позднее связывание — выбор метода во время выполнения.

 **Могут ли быть виртуальными конструкторы? Деструкторы?**  
Конструкторы — нет, деструкторы — да.

 **Наследуется ли virtual?**  
Да.

 **Какое отношение устанавливает открытое наследование?**  
"Является" (is-a).

 **Какое отношение устанавливает закрытое наследование?**  
"Реализован в терминах" (has-a).

 **Принцип подстановки?**  
Производный класс может заменять базовый.

 **Какие данные у объекта Teacher x?**  
age (private), name (public), post (protected), stage (protected).

 Конструкторы и оператор присваивания:

// Student

Student::Student() : age(0), name("") {}

Student::Student(string n, int a) : name(n), age(a) {}

Student::Student(const Student& s) : name(s.name), age(s.age) {}

Student& Student::operator=(const Student& s) { name = s.name; age = s.age; return \*this; }

// Employee

Employee::Employee() : Student(), post("") {}

Employee::Employee(string n, int a, string p) : Student(n, a), post(p) {}

Employee::Employee(const Employee& e) : Student(e), post(e.post) {}

Employee& Employee::operator=(const Employee& e) { Student::operator=(e); post = e.post; return \*this; }

// Teacher

Teacher::Teacher() : Employee(), stage(0) {}

Teacher::Teacher(string n, int a, string p, int s) : Employee(n, a, p), stage(s) {}

Teacher::Teacher(const Teacher& t) : Employee(t), stage(t.stage) {}

Teacher& Teacher::operator=(const Teacher& t) { Employee::operator=(t); stage = t.stage; return \*this; }