

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Пермский национальный исследовательский политехнический университет
Электротехнический факультет
Кафедра информационных технологий и автоматизированных систем

ОТЧЕТ
о работе по информатике
Семестр: 2

На тему: «Простое наследование. Принцип подстановки.»

Выполнил студент ИВТ-22-26:
Дияров Тимур Артурович

Проверил доцент кафедры ИТАС:
Полякова Ольга Андреевна

Пермь 2023

Постановка задачи

Определить методы проверки на равенство и операцию перемножения полей. Реализовать операцию вычитания пар по формуле $(a,b)-(c,d)=(a-b,c-d)$. Создать производный класс ПРОСТАЯ_ДРОБЬ(RATIONAL), с полями Числитель и Знаменатель. Переопределить операцию вычитания и определить операции сложения и умножения простых дробей.

Описание классов

Класс "PAIR" с двумя полями "first" (типа int) и "second" (типа int). В структуре имеются шесть методов:

- Get_first(); Get_second(); Set_first(); Set_second(): инициализация переменных
- Compare(int C, int P): сравнение переменных
- Multiply(int C, int P): умножение переменных

Класс "RATIONAL" с двумя полями "numerator" (типа int) и "divider" (типа int) и который наследуется от класса "PAIR".

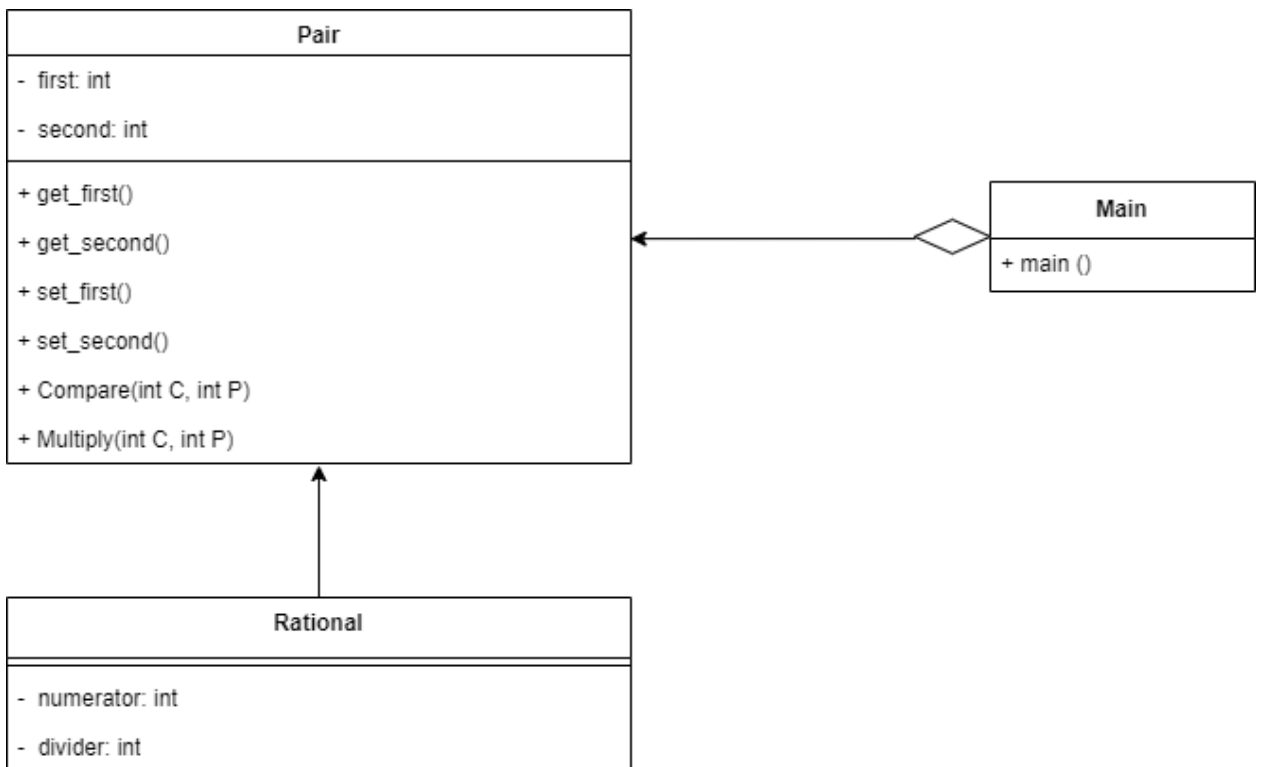
Скриншот работы программы

```
first:10
second:2
10 /= 2
10 * 2 = 20
first:5
second:5
5 == 5
5 * 5 = 25
pair a - pair b = 5,-3
numerator:15
divider:3

numerator:20
divider:4

fraction c + fraction d = 10
fraction c * fraction d = 25
```

UML-диаграмма



Контрольные вопросы

1. Механизм наследования используется для создания новых классов, используя существующие классы как основу.
2. Компоненты этого класса будут публично доступны во всей программе.
3. Компоненты этого класса будут доступны только внутри него.
4. Компоненты этого класса будут доступны только внутри него и в классах, которые наследуются от того же класса, что и рассматриваемый класс.
5. Производный класс описывается как все другие классы, за исключением указания класса-основы:

```
class Derived1 : public Base {
    /* ... */
};
```

6. Конструкторы наследуются.
7. Деструкторы не наследуются.
8. Конструкторы производных классов вызываются в порядке их определения в классе.
9. Деструкторы производных классов вызываются в обратном порядке.

10. Виртуальные функции позволяют классам переопределять методы, что позволяет переопределять поведение производного класса без необходимости повторять код. Спецификатор `virtual` допускает динамическую привязку. Это означает, что при создании объекта он является ссылкой на базовый класс, а когда в базовом классе выполняются вызовы функций, вызов функции разрешается во время выполнения в зависимости от типа объекта.