Федеральное государственное автономное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Отчёт по лабораторной работе № 5

Выполнил:

Студент группы РИС-23-1б

Гордеев Василий Андреевич

Проверила:

Доцент кафедры ИТАС

О. А. Полякова

2024

1. Постановка задачи
   1. Определить абстрактный класс.
   2. Определить иерархию классов, в основе которой будет находиться абстрактный класс (см. лабораторную работу №4).
   3. Определить класс Вектор, элементами которого будут указатели на объекты иерархии классов.
   4. Перегрузить для класса Вектор операцию вывода объектов с помощью потоков.
   5. В основной функции продемонстрировать перегруженные операции и полиморфизм Вектора.

1. Анализ задачи
   1. Базовый класс:
   2. ПАРА\_ЧИСЕЛ (PAIR)
   3. Первое\_число (first) - int
   4. Второе\_число (second) – int
   5. Определить методы проверки на равенство и операцию перемножения полей.
   6. Реализовать операцию вычитания пар по формуле (a,b)-(c,d)=(a-b,c-d)
   7. Создать производный класс ПРОСТАЯ\_ДРОБЬ(RATIONAL), с полями Числитель и Знаменатель. Переопределить операцию вычитания и определить операции сложения и умножения простых дробей.

3. Код

|  |
| --- |
| #pragma once  #include <vector>  #include <iostream>  #include <stdexcept>  #include <utility>  template<typename T>  class Vector {  public:  // Конструктор по умолчанию  Vector() : \_size(0), \_capacity(0), \_data(nullptr) {}  // Деструктор  ~Vector() {  delete[] \_data;  }  // Добавление элемента в конец вектора  void push\_back(const T& value) {  if (\_size >= \_capacity) {  if (\_capacity == 0) \_capacity = 1;  else \_capacity \*= 2;  T\* newData = new T[\_capacity];  for (size\_t i = 0; i < \_size; ++i) {  newData[i] = \_data[i];  }  delete[] \_data;  \_data = newData;  }  \_data[\_size++] = value;  }  // Получение элемента по индексу  T& operator[](size\_t index) {  if (index >= \_size) {  throw std::out\_of\_range("Index out of range");  }  return \_data[index];  }  // Получение размера вектора  size\_t size() const {  return \_size;  }  private:  size\_t \_size; // Текущий размер вектора  size\_t \_capacity; // Емкость вектора (количество выделенной памяти)  T\* \_data; // Указатель на массив элементов  }; |

1. Реализация UML

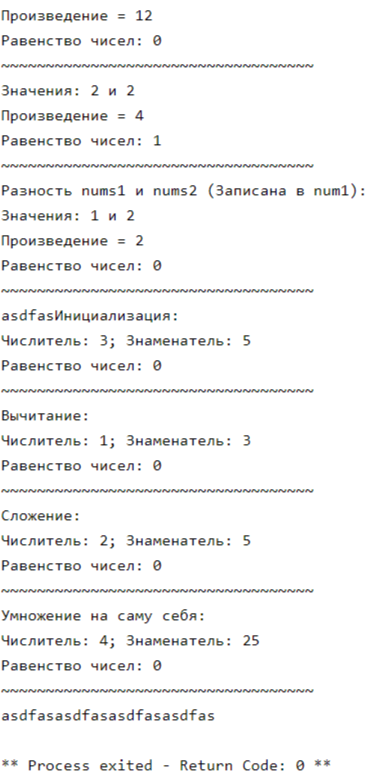
|  |
| --- |
| class Numbers |
|  |
| numbers();  ~numbers();  virtual void print() = 0; |

|  |
| --- |
| class Pair : Numbers |
| int first;  int second; |
| Pair();  Pair(int, int);  Pair(Pair&);  int getFirst();  int getSecond();  bool ecualityCheck();  int multiplication();  void print();  Pair& subtract(Pair&); |

|  |
| --- |
| class Rational : public Pair |
|  |
| Rational();  Rational(int, int);  Rational(Pair&);  Rational& subtract(Pair&);  Rational& addition(Pair&);  Rational& multiplication(Pair&);  void print(string); |

|  |
| --- |
| class Vector |
| size\_t \_size; // Текущий размер вектора  size\_t \_capacity; // Емкость вектора (количество выделенной памяти)  T\* \_data; |
| Vector()  ~Vector()  void push\_back(const T& value)  T& operator[](size\_t index)  size\_t size() const |

1. Консоль:



Вывод: программа работает корректно и выдаёт требуемый результат.