

Лабораторная работа 1 - задание

Требуемые условия завершения

Задание:

1. Создать кроссплатформенную статическую библиотеку, содержащую класс по варианту. Класс должен иметь:

1. 1. Приватные поля (обычно 2-4 поля)
- 1.2. Конструкторы (по умолчанию и с параметрами)
- 1.3. get и set методы
- 1.4. 5-6 методов public
- 1.5. Перегрузку 2-3 операторов (например, ==, +, <<, >)
- 1.6. 1 статический метод (вспомогательный)

1.7. Метод для вывода состояния

* Пример класса приведен в конце файла

2. Продемонстрировать использование библиотеки в отдельном приложении.

Варианты для самостоятельного выполнения

1. Класс Time (Время)

- Поля: hours, minutes, seconds
- Методы: addSeconds(int), difference(Time), toSeconds(), isValid()
- Операторы: ==, -, <<
- Статический метод: fromString(string)

2. Класс Date (Дата)

- Поля: day, month, year
- Методы: addDays(int), dayOfWeek(), isLeapYear(), daysBetween(Date)
- Операторы: ==, <, <<
- Статический метод: isValidDate(int, int, int)

3. Класс Fraction (Дробь)

- Поля: numerator, denominator
- Методы: reduce(), toDecimal(), inverse(), power(int)
- Операторы: +, ==, <<
- Статический метод: gcd(int, int) (НОД)

4. Класс Complex (Комплексное число)

- Поля: real, imag
- Методы: modulus(), conjugate(), argument()
- Операторы: +, *, ==, <<
- Статический метод: fromPolar(double, double)

5. Класс Vector2D (2D Вектор)

- Поля: x, y
- Методы: length(), normalize(), rotate(double angle)
- Операторы: +, * (скалярное произведение), <<
- Статический метод: distance(Vector2D, Vector2D)

6. Класс Point2D (2D Точка)

- Поля: x, y
- Методы: distanceTo(Point2D), move(double, double)
- Операторы: ==, - (возвращает вектор), <<
- Статический метод: midpoint(Point2D, Point2D)

7. Класс Line (Прямая)

- Поля: A, B, C (коэффициенты Ax + By + C = 0)
- Методы: containsPoint(Point2D), isParallelTo(Line), yFromX(double)
- Операторы: == (совпадение), <<
- Статический метод: fromPoints(Point2D, Point2D)

8. Класс Circle (Окружность)

- Поля: center (Point2D), radius

- Методы: area(), circumference(), containsPoint(Point2D)
- Операторы: ==, <<
- Статический метод: fromThreePoints(Point2D, Point2D, Point2D)

9. Класс Rectangle (Прямоугольник)

- Поля: topLeft (Point2D), width, height
- Методы: area(), perimeter(), containsPoint(Point2D)
- Операторы: ==, <<
- Статический метод: intersection(Rectangle, Rectangle) (возвращает прямоугольник/флаг)

10. Класс Triangle (Треугольник)

- Поля: point1, point2, point3 (Point2D)
- Методы: area(), perimeter(), type()
(равносторонний/равнобедренный/разносторонний)
- Операторы: ==, <<
- Статический метод: isValid(Point2D, Point2D, Point2D)

11. Класс RGBColor (Цвет)

- Поля: red, green, blue (0-255)
- Методы: toHex(), invert(), grayscale()
- Операторы: ==, + (смешивание), <<
- Статический метод: fromHex(string)

12. Класс Book (Книга)

- Поля: ISBN, title, author, year
- Методы: isClassic() (старше 50 лет), shortInfo()
- Операторы: ==, < (по году), <<
- Статический метод: validateISBN(string)

13. Класс Student (Студент)

- Поля: id, name, averageGrade

- Методы: isExcellent() (средний балл ≥ 4.5), upgradeGrade(double)

- Операторы: ==, > (по среднему баллу), <<

- Статический метод: generateId()

14. Класс BankAccount (Банковский счет)

- Поля: number, owner, balance

- Методы: deposit(double), withdraw(double) (с проверкой), applyInterest(double percent)

- Операторы: ==, > (по балансу), <<

- Статический метод: generateAccountNumber()

15. Класс Currency (Деньги)

- Поля: amount, code ("USD", "EUR" и т.д.)

- Методы: convertTo(string code, double rate), format() ("\$100.00")

- Операторы: +, ==, <<

- Статический метод: getSymbol(string code)

16. Класс Phone (Телефонный номер)

- Поля: countryCode, number

- Методы: isValid(), format() ("+7 (999) 123-45-67")

- Операторы: ==, <<

- Статический метод: parse(string)

17. Класс Email (Адрес электронной почты)

- Поля: localPart, domain

- Методы: isValid(), getService() (gmail.com → "Google")

- Операторы: ==, <<

- Статический метод: generate(string name, string domain)

18. Класс GeoLocation (Географические координаты)

- Поля: latitude, longitude

- Методы: distanceTo(GeoLocation) (формула Хаверсина), hemisphere()
- Операторы: ==, <<
- Статический метод: fromDMS(double, double, double, char) (из градусов/минут/секунд)

19. Класс Polynomial (Многочлен 2-й степени)

- Поля: a, b, c (коэффициенты $ax^2 + bx + c$)
- Методы: valueAt(double x), findRoots() (возвращает структуру с корнями), derivative()
- Операторы: +, <<
- Статический метод: fromRoots(double, double)

20. Класс Matrix2x2 (Матрица 2x2)

- Поля: m11, m12, m21, m22
- Методы: determinant(), inverse(), transpose()
- Операторы: +, * (на число), <<
- Статический метод: identity()

21. Класс Counter (Счетчик)

- Поля: value, min, max
- Методы: increment(), decrement(), reset(), isAtMin(), isAtMax()
- Операторы: ++ (префиксный), ==, <<
- Статический метод: createPercentageCounter()

22. Класс Dice (Игральная кость)

- Поля: sides, lastRoll
- Методы: roll(), isLucky() (выпало 6 для 6-гранной), rollMultiple(int)
- Операторы: ==, > (сравнение по последнему броску), <<
- Статический метод: rollTwoDice(Dice, Dice)

23. Класс Card (Игральная карта)

- Поля: rank ("Ace", "2"...), suit ("Hearts", "Spades")

- Методы: isRed(), value() (для игры в 21), fullName() ("Ace of Spades")
- Операторы: ==, <<
- Статический метод: createDeck() (возвращает вектор из 52 карт)

24. Класс Temperature (Температура)

- Поля: value, scale ('C', 'F', 'K')
- Методы: convertTo(char scale), isBoiling() (для воды), isFreezing()
- Операторы: ==, <, <<
- Статический метод: celsiusToFahrenheit(double)

25. Класс Password (Пароль)

- Поля: hash (строка), length
- Методы: strength() (weak/medium/strong), matches(string input)
- Операторы: ==, <<
- Статический метод: generate(int length, bool useSymbols)

Требования к main.cpp:

1. Создать 2-3 объекта разными конструкторами
2. Продемонстрировать работу всех методов
3. Показать перегруженные операторы
4. Вызвать статический метод
5. Вывести информацию об объектах

Пример класса для

```
// include/mylib/Time.h

#pragma once

#include <iostream>

#include <string>

namespace mylib {

    class Time {

```

```
private:
```

```
    int hours;
```

```
    int minutes;
```

```
    int seconds;
```

```
public:
```

```
    // Конструкторы
```

```
    Time();
```

```
    Time(int h, int m, int s);
```

```
    // get и set методы
```

```
    int getHours() const;
```

```
    void setHours(int h);
```

```
    // ... для minutes и seconds
```

```
    // Публичные методы (5-6)
```

```
    void addSeconds(int sec);
```

```
    Time difference(const Time& other) const;
```

```
    int toSeconds() const;
```

```
    bool isValid() const;
```

```
    std::string toString() const;
```

```
    // Перегрузка операторов (2-3)
```

```
    bool operator==(const Time& other) const;
```

```
    Time operator-(const Time& other) const;
```

```
    friend std::ostream& operator<<(std::ostream& os, const Time& t);
```

```
// Статический метод  
static Time fromString(const std::string& str);  
  
// Метод вывода состояния  
void printState() const;  
  
};  
}
```