# shandalovych01 Documentation

## Мета

Навчитись створювати об'єкти. Отримати розуміння створення об'єкта на стеку а також передачу об'єкту по значенню.

# 1.ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Створити клас даних `Wheel` та класс відображення даних - `Screen`. Об'єкт відображення конструюється на стеку функції `main()` об'єктом даних, що заздалегіть створений на стеку. Передавати `Wheel` як значення. `Wheel` має всі публічні поля та методи. `Screen` лише виконує відображення даних у формі `<назва поля>=<значення>;` всі його методи та атрибути публічні.

# 2.РОЗРОБКА ПРОГРАМИ

## 2.1 Засоби ООП

В ході розробки програми були використані такі засоби ООП:

- Абстракція - кожен об'єкт описує свою особливу сутність, яка визначається його полями.
- Інкапсуляція - поля об'єктів закриті для користувача,натомість ми даємо доступ до даних за допомогою геттерів та сеттерів, так користувач має можливість отримати готові дані, а не обробляти їх, для подпльшого вікористання.

## 2.2 Ієрархія та структура класів

На рис 2.1 дивись ієрархію класів

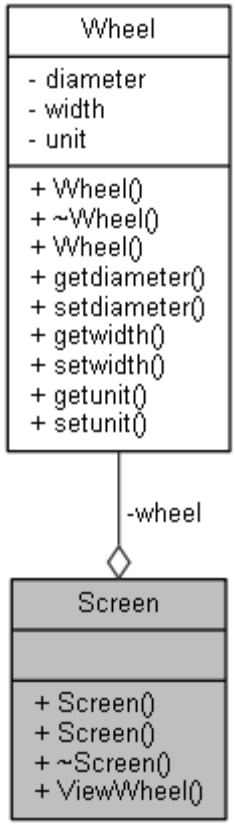


Рисунок 2.1 ієрархія класів

## 2.3 Опис програми

На рис 2.2 дивись структуру проекту.

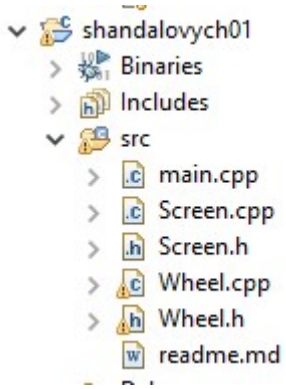


Рисунок 2.2 структура проекту

На рис 2.3 дивись призначення класів.

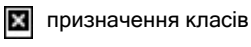


Рисунок 2.3 призначення класів

Ми маємо 2 класа: Вікно `Window` та Відображувач **`Screen`**. Вікно описує сутність програмного вікна, тобто його id та координати, а відображувач використовується для виводу даних з об'єкту вікна у консоль. У функції **`main()`** відображена робота програми.

### 2.4 Важливі фрагменти програми

У программі слід зауважити увагу на таких моментах:

**Класс `Wheel.cpp` списки ініціалізації**

```
1  Wheel::Wheel(int diameter, int width, string unit): diameter(diameter), unit
    (unit), width(width) {
2      cout << "Вызов коструктора с параметром\n";
3
4  }
```

**Класс `Screen.cpp` вивід інформації у консоль:**

```
1      cout << "diameter = " << wheel.getdiameter() << "\n";
2      cout << "unit = " << wheel.getunit() << "\n";
3      cout << "width = " << wheel.getwidth() << "\n";
4  }
```

**Класс `main.cpp` демонстрація роботи**

```
1  int main(void) {
2
3      //Константы для инициализации первого объекта.
4      const int diameter = 20;
5      const int width = 10;
6      const string unit = "см";
7
8
9      Wheel data( diameter, width, unit);
10
11     Screen screen(data);
12
13     screen.ViewWheel();
14
15     return 0;
16 }
```

# 3.РЕЗУЛЬТАТИ РОБОТИ

Результат роботи показано на рис 3.1.

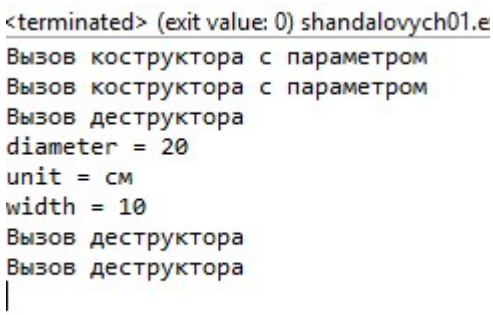


Рисунок 2.3 призначення класів

# ВИСНОВКИ

В результаті виконання лабораторної роботи мною були придбані навички програмування з використання алгоритмів STL, використання наслідування класу. Вважаю, що мета роботи була досягнута.

---