실습 056 this 포인터를 이용한 Builder 클래스 정의

제출 마감일: 2023-04-09 23:59
업로드 가능한 파일 수: 2
제출 방식: 개인

목적

이 실습은 class 내부에서 this 키워드를 사용하는 연습을 합니다.

설명

C++에서 일반적으로 객체는 클래스를 이용해서 생성합니다.

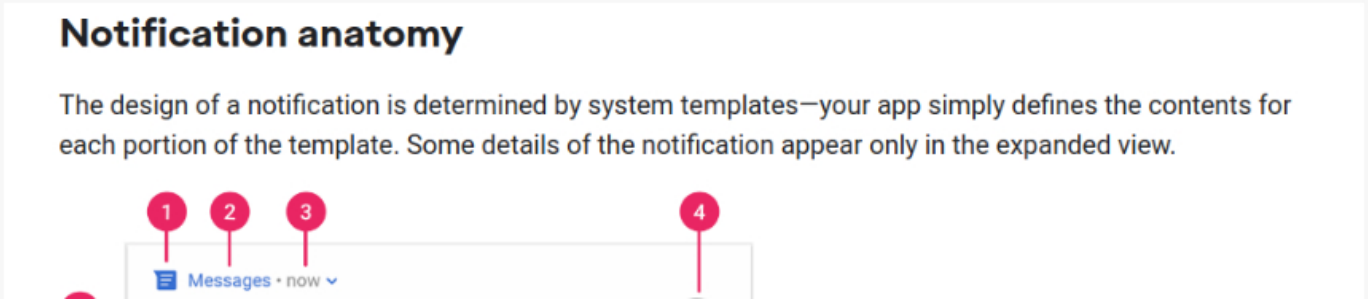
우리가 생성하는 객체가 좀 복잡한 객체일 수 있습니다.

클래스로 한번에 생성하기에는 복잡합니다.

MyObject myObject {3, 4, "me", &std::cout, 500, 12, 550, [] (auto a, auto b) {return a+b;} };

이럴 때, 객체를 생성하는 방법으로 많이 사용하는 패턴이 빌더 (Builder) 패턴입니다.

휴대폰에서 알림을 보내는 객체를 생성해 봅시다.



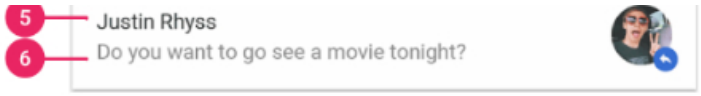


Figure 7. A notification with basic details

The most common parts of a notification are indicated in figure 7 as follows:

1. Small icon: This is required and set with `setSmallIcon()`.
2. App name: This is provided by the system.
3. Time stamp: This is provided by the system but you can override with `setWhen()` or hide it with `setShowWhen(false)`.
4. Large icon: This is optional (usually used only for contact photos; do not use it for your app icon) and set with `setLargeIcon()`.
5. Title: This is optional and set with `setContentTitle()`.
6. Text: This is optional and set with `setContentText()`.

Notification 클래스를 정의하여 알림 객체를 생성합니다.

알림 객체 생성을 도와 주는 NotificationBuilder 클래스 역시 정의합니다.

```
class Notification {

public:
   static NotificationBuilder create();

private:
   Notification() = default;
   int icon;
   std::string appName = "PNU";
   std::time_t timestamp = std::time(nullptr);
   std::string title;
   std::string text;
   int priority;
};
```

생성자를 private 으로 선언했으므로 Notification 객체를 직접 생성할 수 없습니다.

알림 객체를 생성할 빌더를 정의해 보겠습니다.

```
class NotificationBuilder {
public:
   //user-defined conversion function
   operator Notification() const { return std::move(notification); }
   NotificationBuilder& setSmallIcon(int icon);
   NotificationBuilder& setContentTitle(std::string title);
   NotificationBuilder& setContentText(std::string text);
   NotificationBuilder& setPriority(int priority);
private:
```

```
    Notification notification;
};
```

특이한 점은 빌더의 멤버 함수가 자신의 참조를 반환하고 있는 점입니다.

빌더를 이용해 알림 객체를 생성해 보겠습니다.

```
Notification notification = Notification::create()
        .setSmallIcon(1)
        .setContentTitle("Justin Rhyss")
        .setContentText("Do you want to go see a movie tonight?")
        .setPriority(5);
```

문제

알림 객체를 생성하는데 필요한 Notification 과 NotificationBuilder 클래스를 정의하시오.

<참고>

//builderTest.cpp

```
int main(){
    Notification notification = Notification::create()
            .setSmallIcon(1)
            .setContentTitle("Justin Rhyss")
            .setContentText("Do you want to go see a movie tonight?")
            .setPriority(5);
    std::cout << notification << std::endl;
}
```

//Notification.h

```
class NotificationBuilder;
class Notification {
public:
    static NotificationBuilder create();
    friend class NotificationBuilder;
    friend std::ostream& operator << (std::ostream &os, const Notification& n);

private:
    Notification() = default;
    int icon;
    std::string appName = "PNU";
    std::time_t timestamp = std::time(nullptr);
    std::string title;
    std::string text;
    int priority;
};
```

```
class NotificationBuilder {
public:
    //user-defined conversion function
    operator Notification() const { return std::move(notification); }
    NotificationBuilder& setSmallIcon(int icon);
    NotificationBuilder& setContentTitle(std::string title);
    NotificationBuilder& setContentText(std::string text);
    NotificationBuilder& setPriority(int priority);
private:
    Notification notification;
};
```

//Notification.cpp

```
std::ostream& operator << (std::ostream& os, const Notification& n){
    os << n.icon << " " << n.appName << " " ;
    int diff = (int)std::difftime(std::time(nullptr), n.timestamp) / (60 * 60 * 24);
    os <<  ((diff==0) ? "now" : std::to_string(diff) );
    os << ((diff==0) ? "" : " days" ) << std::endl;
    os << n.title << std::endl;
    os << n.text << std::endl;
    return os;
}

NotificationBuilder Notification::create(){
    // your code here
}

NotificationBuilder& NotificationBuilder::setSmallIcon(int icon) {
    // your code here
}

NotificationBuilder& NotificationBuilder::setContentTitle(std::string title) {
    // your code here
}

NotificationBuilder& NotificationBuilder::setContentText(std::string text) {
    // your code here
}

NotificationBuilder& NotificationBuilder::setPriority(int priority) {
    // your code here
}
```

입력

없음

출력

1 PNU now

Justin Rhyss

Do you want to go see a movie tonight?

제출파일

Notification.cpp

56.csv

입출력

입력	출력
	1 PNU now Justin Rhyss Do you want to go see a movie tonight?

VPL