
Przyjąć, że udostępniona jest przestrzeń nazw std i są dodane niezbędne nagłówki z biblioteki standardowej

Zadanie 1 (2pkt)

Obiekty typu `DigitalMedia` pozwalają odtwarzać utwory o danej nazwie. Zmień strukturę, aby unikać powielania kodu. Przykład użycia:

```
void playMedia(const DigitalMedia& dm) {
    dm.play("Pan_Tadeusz");
}
```

```
static const int USER_ID = 1;
VideoDigitalMedia player1(USER_ID);
AudioDigitalMedia player2(USER_ID);
TextDigitalMedia player3(USER_ID);
playMedia(player1);
playMedia(player2);
playMedia(player3);
```

```
class DigitalMedia {
public:
    DigitalMedia(int userId) : userId_(userId) {}
    virtual void play(const string& name) const = 0;

    bool checkCredentials() const; //sprawdza, czy uzytkownik moze odtwarzac
private:
    int userId_;
};

class VideoDigitalMedia : public DigitalMedia {
public:
    VideoDigitalMedia(int userId) : DigitalMedia(userId) {}

    void play(const string& name) const {
        if( checkCredentials() ) {
            //kod, ktory odtwarza wideo
        }
    }
};

class AudioDigitalMedia : public DigitalMedia {
public:
    AudioDigitalMedia(int userId) : DigitalMedia(userId) {}

    void play(const string& name) const {
        if( checkCredentials() ) {
            //kod, ktory odtwarza audio
        }
    }
};

class TextDigitalMedia : public DigitalMedia {
public:
    TextDigitalMedia(int userId) : DigitalMedia(userId) {}

    void play(const string& name) const {
        if( checkCredentials() ) {
            //kod, ktory odtwarza tekst
        }
    }
};
```

Uwagi do prowadzącego (R.Nowak):

Zadanie 2 (2pkt)

Opisz fragment aplikacji tworzonej obiektowo, w którym warto stosować jednocześnie C++ i Python. Fragment powinien zawierać minimum 3 klasy. Opisz odpowiedzialności klas i uzasadnij dlaczego uważasz, że powinna być dostarczona w C++ albo w Python. Skrócony opis odpowiedzialności podsystemu: 1-2 zdania:

Nazwa klasy, odpowiedzialność, język programowania, uzasadnienie wyboru:

Zadanie 3 (2pkt)

Daj możliwość kopiowania obiektów typu Foo. Kopia składowej może to być kopia płytka.

```
class Goo {  
};  
  
//chcemy dac mozliwosc kopiowania, kopia jest akopi lapytk lskadowej g_  
class Foo {  
public:  
    Foo() : g_(new Goo) {}  
    Foo(const Foo& f) {}  
    ~Foo() {}  
private:  
    unique_ptr<Goo> g_;  
};
```

Imię i nazwisko:

25L-1 60 min

Zadanie 4 (2pkt)

Następujący kod skutkuje pętlą nieskończoną. Wyłumacz dlaczego tak się dzieje. Napraw kod tak, aby pętla wykonała się 10 razy.

```
#include <cstdlib>
#include <print>

int main() {
    int16_t i;
    unsigned int n = 9;
    for (i = 0; n - i >= 0; ++i) {
        std::println("Hi_{}!", i);
    }
    return 0;
}
```

Zadanie 5 (2pkt)

Czym jest monomorfizacja w kontekście programowania generycznego/szablonów?

Uwagi do prowadzącego (Ł. Neumanna):