Jakub Łabendowicz 17.04.2020r.

Sprawozdanie

LABORATORIUM 6. EWOLUCJA PROGRAMOWANIA OBIEKTOWEGO REALIZOWANA POPRZEZ DZIEDZICZENIE I POLIMORFIZM

# Pytania kontrolne:

## 1 . Opisz czym jest dziedziczenie.

Przekazywanie składników klasy bazowej do klasy potomnej. W klasie potomnej można dodawać nowe składniki klasy, które nie będą dostępne dla klasy bazowej.

## Odpowiedz jakie składniki się nie dziedziczą, a jakie się dziedziczą.

Klasa nie dziedziczy konstruktorów i destruktorów.

## Przedstaw jak sterować widocznością składników przy dziedziczeniu.

Poprzez modyfikatory dostępu (public, private, protected) oraz sposób dziedziczenia. Dziedziczenie „protected” powoduje, że odziedziczone składniki public i protected stają się protected. Dziedziczenie „private” powoduje, że odziedziczone składniki public i protected stają się private. Natomiast składniki, które są private w klasie bazowej są dziedziczone, ale nie ma do nich bezpośredniego dostępu z klasy pochodnej.

## Wyjaśnij jak utworzyć metodę wirtualną.

Poprzez słowo „virtual”.

## Opisz na czym polega polimorfizm dynamiczny.

Jeżeli klasa bazowa i pochodna mają zdefiniowaną identyczną metodę wirtualną, można zadeklarować wskaźnik do klasy bazowej, wskaźnikiem tym pokazać na klasę pochodną i wywołać tę metodę wirtualną. Wywołana zostanie metoda klasy pochodnej mimo tego, że wskaźnik jest do klasy bazowej.

# Zadania 6.1.-6.3.

## Listing 6.2, wiersz 4 → Skąd się wzięła metoda doTekstu w klasie Kwadrat?

Klasa Kwadrat dziedziczy ją od klasy prostokąt.

## Listing 6.2, wiersz 6 → Dlaczego to nie zadziała?

Ponieważ klasa kwadrat przyjmuje dwa argumenty

## Listing 6.3, wiersze 5 6 → Jak działa konstruktor klasy Kwadrat?

Inicjalizuje argument nazwa typu string oraz bok typu double i przekazuje je do konstruktora Prostokat.

## Listing 6.6, wiersz 12 → Po co jest znak & ? I jak on działa w tym przypadku?

W celu przekazania adresu zmiennej.

## Listing 6.7, wiersz 12 → Czy można byłoby użyć tutaj pierwszej referencji?

Tak, ale wtedy uruchamiają się metody klasy bazowej, bo nie ma polimorfizmu dynamicznego.

## Listing 6.8, wiersze 5 i 6 → Co oznacza „=0” ?

Oznacza że metoda wirtualna zostaje przesłonięta w klasie pochodnej.

## Listing 6.10 → Czy zamiast referencji można użyć wskaźnika?

Tak.

## Czy można utworzyć obiekt klasy Mieszkaniec?

Tak.

## Czy można utworzyć obiekt klasy ZamiarMieszkanca?

Tak.

## Listing 6.11, wiersze 6 i 7 → Co oznaczają fragmenty „= NIC”oraz „= NIGDZIE”?

Przypisanie wartości wstępnej.

## Listing 6.12, wiersz 1 → czy fragment class Sasiedztwo można zastąpić #inclulde sasiedztwo.h „sasiedztwo.h” ”Inna nazwa”?

Nie.

## Listing 6.12, wiersz 13 → Jak nazywa się ta metoda?

Konstruktor.

## Listing 6.15, wiersz 15 → Co powoduje słowo final ?

Zapobiega możliwości przesłaniania danej metody przez klasy potomne.

## Listing 6.12 → Które metody są abstrakcyjne ?

kimJestes, wybierzAkcje, dajPotomka, przyjmijZdobycz