Projektowanie i programowanie obiektowe

Interfejsy w programowaniu obiektowym

mgr inż. Krzysztof Rewak

13 listopada 2019

1 Interfejsy w programowaniu obiektowym

Należy uruchomić w dowolnym środowisku załączony program z katalogu `lab07`. Programy we wszystkich językach wykonują dokładnie to samo zadanie.

1.1 Java

W katalogu `lab07/java` znajduje się kod źródłowy aplikacji podzielonej na pakiety. Należy zbudować, skompilować i uruchomić projekt w dowolny sposób. Zalecane jest środowisko IntelliJ IDEA (dostępne dla studentów za darmo).

1.2 PHP

W katalogu `lab07/php` znajduje się kod źródłowy aplikacji wymagającej autoloadera. Należy uruchomić komendę `composer install`, aby utworzyć katalog `vendor` z wymaganymi plikami. Aplikację uruchamia się poprzez polecenie `php index.php` w głównym katalogu. Zalecane jest środowisko PHPStorm (dostępne dla studentów za darmo).

1.3 Pytania do zadania

Należy odpowiedzieć na następujące pytania dotyczące interfejsów:

- czym jest interfejs?
- jakie metody musi implementować każda klasa, której obiekt będzie podjeżdżał pod bramę parkingu?
- ile interfejsów może być zaimplementowych w jednej klasie?
- czy interfejsy są dziedziczone?
- co się stanie jeżeli pod bramę podjedzie czołg klasy `Tank`?
- czy strażnik wpuści dwa samochody o takim samym numerze rejestracyjnym? A powinien?

A także dotyczące technikaliów:

- jak działają przestrzenie nazw w PHP i pakiety w Javie?
- czy podział programu w konwencji *jedna klasa - jeden plik* jest czytelny?
- dlaczego w pliku `.gitignore` dodano foldery `.idea` oraz `vendor` dla PHP i `out` dla Javy?

2 Zadanie programistyczne

Należy rozszerzyć program tak, aby przedstawić znajomość zagadnień przedstawianych na dotychczasowych zajęciach (dziedziczenie oraz interfejsy), a także z poprzednich (pola i metody, konstruktory, enkapsulacja, hermetyzacja).

- generator kolejki powinien być konfigurowalny, tak, aby generować konkretne zestawy danych;
- generator powinien uruchomić się kilkukrotnie *w ciągu dnia*; najlepiej zrealizować to iterowaniem po pętli z godzinami wejścia;
- każde wejście powinno zostać odnotowane razem z czasem wejścia;
- parking ma określony limit miejsc parkingowych, zatem parkingowy nie powinien wpuścić więcej samochodów niż ma miejsc;
- parking powinien być płatny, analogicznie do parkingu przy kampusie; pracownicy uczelni wjeżdżają bez opłat (chciałbym!), natomiast prawie wszyscy inni płacą określoną cennikiem sumę;
- na kampus mogą wjechać bez opłat samochody dostawcze kurierów oraz pojazdy uprzywilejowane, ale nie wliczają się do liczby zajętych miejsc parkingowych;
- na kampus mogą wjechać bez opłat rowerzyści, ale jedynie w liczbie nieprzekraczającej określonej liczby stojaków na rowery;
- na kampus nie mogą wjechać czołgi, ponieważ są zbyt szerokie;
- na koniec dnia wyświetlana jest uzbierana przez parkingowego kwota;
- parkingowy posiada *czarną listę* tablic rejestracyjnych, których właścicieli nie wpuszcza na teren parkingu;

Plik z programem należy dołączyć do repozytorium Git. Zalecane jest uporządkowanie zadań w odpowiadającym im katalogach. Prezesłany program zostanie oceniony, a ocena będzie częścią składową oceny semestralnej.

2.1 Zadanie dla ambitnych

Należy zastanowić się jak sensownie zaimplementować mechanizm wypuszczania obiektów z parkingu. Idealnym wyjściem byłoby, gdyby o godzinie 23:00 kampus był pusty, a więc wypadałoby zmusić wszyskich - za wyjątkiem pracowniczych samochodów z wykupionym abonamentem na nocne parkowanie - do opuszczenia terenu.