|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
| Test końcowy | | | *PS-06-01-Z1/03*  Ważne od: 2021.07.02 | |
| Wzorce projektowe w inżynierii oprogramowania z zastosowaniem języków C++/Java ***SK-PYTHON-DEEPIMPL***  ***Data: 09.06.2025 – 11.06.2025***  ***Wykładowca: Marcin Albiniak***  ***TES\_240917\_2551500INN\_140408AK\_*** | | |  | |
| ***Dane uczestnika testu*** | ***Wyniki – punktacja*** | |
| ***Imię:*** proszę uzupełnij | ***Maksymalna: 12 pkt.*** | |
| ***Nazwisko:*** proszę uzupełnij | ***Uzyskana: …*** | |
| ***Data wypełnienia testu: 2025-06-11*** |  | |
| ***Miejscowość: Internet*** |  | |

**1. Który z poniższych wzorców należy do kategorii kreacyjnych?**  
 Adapter  
 Observer  
Singleton  
 Proxy

**2. Kto jest autorem tzw. „Gang of Four”, który opisał klasyczne wzorce projektowe?**  
 Donald Knuth  
 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides  
 Bjarne Stroustrup  
 James Gosling

**3. Czym różni się wzorzec Abstract Factory od Factory Method?**  
 Abstract Factory tworzy pojedyncze obiekty  
 Factory Method wymaga wielu klas  
Abstract Factory zwraca zestawy powiązanych obiektów, Factory Method tylko jeden obiekt  
 Factory Method jest zawsze singletonem

**4. Który z poniższych wzorców najlepiej zastosować do dynamicznej wymiany algorytmów podczas działania programu?**  
 Observer  
 Command  
Strategy  
 Prototype

**5. Wzorzec Composite jest używany, gdy:**  
 Chcemy mieć tylko jeden egzemplarz klasy  
 Chcemy traktować obiekty indywidualne i grupy obiektów jednolicie  
 Potrzebujemy ukryć szczegóły implementacji  
 Chcemy dynamicznie wymieniać zachowanie

**6. Które z poniższych stwierdzeń jest zgodne z zasadą Open/Closed z SOLID?**  
 Klasa powinna być zawsze zamknięta na zmiany  
 Klasa powinna być otwarta na rozszerzenia, ale zamknięta na modyfikacje  
 Każda klasa może być modyfikowana w dowolnym momencie  
 Klasa powinna być zamknięta na dziedziczenie

**7. Który wzorzec umożliwia powiadamianie wielu obiektów o zmianie stanu?**  
 State  
 Visitor  
 Observer  
 Decorator

**8. Który wzorzec strukturalny pozwala na „opakowywanie” obiektów nową funkcjonalnością w czasie działania programu?**  
 Facade  
Decorator  
 Proxy  
 Adapter

**9. Co charakteryzuje wzorzec Proxy?**  
 Tworzy kopie obiektów  
 Kontroluje dostęp do innego obiektu  
 Umożliwia iterowanie po kolekcji  
v Realizuje powtarzalne polecenia

**10. Który wzorzec architektoniczny polega na rozdzieleniu interfejsu użytkownika od logiki biznesowej i modelu danych?**  
 Composite  
 Strategy  
 MVC  
 Observer

**11. Kiedy wzorce projektowe mogą przeszkadzać?**  
 Gdy są poprawnie zaimplementowane  
 Gdy nie stosujemy zasad SOLID  
 Gdy stosujemy je nadmiernie lub bez potrzeby, w prostych problemach  
 Gdy mamy wiele projektantów

**12. Wzorzec Chain of Responsibility stosujemy wtedy, gdy:**  
 Potrzebujemy wielu kopii obiektu  
 Chcemy ukryć szczegóły implementacji  
 Chcemy przekazywać żądanie przez łańcuch obiektów aż któreś je obsłuży  
 Chcemy rozdzielić interfejs od logiki biznesowej