Принципы и парадигмы ООП

ОБЩИЙ БАЛЛ 9

1. В чем смысл парадигмы полиморфизма? 1 балл

- Она позволяет скрывать некоторые данные от пользователя системы
- Она позволяет корректно использовать методы класса-потомка вместо базового класса, не зная об этом.
- (•) Она позволяет использовать методы производного класса, который не существует на момент создания базового
- Она предлагает добавлять в класс только существенную для системы информацию о прототипируемом объекте, и не добавлять несущественную

2. В чем заключается парадигма инкапсуляции? 1 балл

- Позволяет сделать систему более гибкой
- (•) Она позволяет упаковать все данные и функции в некоторый единый компонент
- Она позволяет скрывать некоторые данные от пользователя системы
- Она позволяет реализовать класс, выполняющий одну определенную функцию

3. Зачем применяется сокрытие данных объекта? 1 балл

- [ ] Чтобы другие программисты не могли создать такой же объект
- [x] Чтобы отделить пользовательский интерфейс объекта от служебных переменных и методов
- [x] Чтобы защитить данные объекта от некорректного изменения
- [ ] Сокрытие данных не нужно

4. Зачем применяются геттеры и сеттеры? 1 балл

- Геттеры и сеттеры не нужны
- Для организации взаимодействия между объектами
- Для повышения читаемости кода
- (•) Для реализации доступа к приватным атрибутам объекта

5. У класса MyClass есть скрытая переменная __private_variable. В программе создан объект my_object класса MyClass. Как можно получить доступ к его скрытому полю из кода программы? 1 балл

- MyClass.__private_variable
- (•) my_object._MyClass__private_variable
- my_object.__private_variable
- MyClass._my_object__private_variable

6. Сколько существует принципов объектно-ориентированного программирования? 1 балл

- (•) 5
- 3
- 7
- 4

7. Как формулируется принцип подстановки Барбары Лисков? 1 балл

- (•) Функции, которые используют базовый тип должны иметь возможность использовать его подтипы не зная об этом
- Классы должны быть закрыты для изменения, но открыты для расширения
- Объекты со схожим интерфейсом должны иметь общий надтип
- Клиенты не должны зависеть от методов, которые они не используют

8. К чему может привести нарушение принципа открытости/закрытости? 1 балл

- (•) Изменится поведение частей программы, которые используют объекты измененного класса
- Потеряется возможность повторного использования кода
- Код станет менее читаемым
- Потеряется гибкость системы

9. Почему важно соблюдать принцип инверсии зависимостей? 1 балл

- [ ] Исчезнет возможность повторного использования кода
- [x] Нарушится гибкость системы
- [x] Модули верхних уровней придется адаптировать к изменениям в модулях нижнего уровня
- [ ] Модули нижних уровней придется адаптировать к изменениям в модулях верхнего уровня

Я, **Денис Дудин Артурович**, понимаю, что отправка работы, выполненной посторонним лицом, может привести к недоступности этого курса или отключению моего аккаунта Coursera.

Узнайте больше о Кодексе чести Coursera

Сохранить | Отправить