

Главная

Статьи

О проекте

Профиль

## Топ контрибаторов

За последние сутки

Нет данных

Получение Q

## Знаете ли Вы, что

После прохождения теста можно комментировать вопросы теста, а Ваши комментарии увидят модераторы теста и пользователи, которым когда-либо эти вопросы попадались.

## Опросы

Какой у Вас реальный опыт программирования?

☐ меньше 1 года☐ 1 - 2 года☐ более 3 лет

Ответить

## Лента обновлений

Jul 23 01:57

ссылка

Комментарий от GeorgeMayoh:

Вообще я слышал такую версию: Если класс просто соде...

Jul 22 21:30

ссылка

Комментарий от leonx7: Красиво подловил)

Jul 22 21:09

ссылка

Комментарий от leonx7: Вопрос хороший, но точно не для теста, где время ограниче...

Jul 21 20:08

ссылка

Добавлен вопрос в тест ASP.NET MVC

Jul 20 20:59

ссылка

Комментарий от brave\_t: Опечатка в отличиЕ от

## Статистика

Тестов: 153, вопросов: 8596.

## Тест "ООП - Основы" - не пройден пройти еще раз

Правильных ответов: 10 / 20 (50 %) требуется: 15

Дата прохождения теста: 23.07.2020 22:02

✗ Суть отличия классов от других абстрактных типов данных состоит в следующем (выбрать верные утверждения):

<input type="checkbox"/> ✓	При задании типа данных класс определяет интерфейс	3380 / 7264
<input type="checkbox"/>	При задании типа данных класс не определяет интерфейс	1387 / 7264
<input type="checkbox"/> ✓	При задании типа данных класс определяет реализацию	5243 / 7264
<input type="checkbox"/>	При задании типа данных класс не определяет реализацию	826 / 7264
<input checked="" type="checkbox"/>	Отличия нет	648 / 7264

Комментировать вопрос (всего: 6)

Автор: [tehnik](#)

✓ Термин "наследование" обозначает, что...

<input type="radio"/>	В производных классах присутствует часть состояния родительского класса.	1816 / 17723
<input checked="" type="radio"/>	Производные классы содержат поля и методы родительского.	13277 / 17723
<input type="radio"/>	Производные классы содержат методы родительского класса.	1085 / 17723
<input type="radio"/>	Производные классы наследуют поля родительского класса.	993 / 17723
<input type="radio"/>	Производные классы наследуют модификаторы доступа членов родительского класса.	11723

Пояснение: Производные классы содержат поля и методы родительского.

При этом из экземпляров производного класса нельзя получать доступ к полям и методам родительского класса, которые являются закрытыми (если такая поддержка есть в ЯП).

Комментировать вопрос (всего: 5)

Автор: [kukulya](#)

✗ Выберите наиболее точное определение абстракции:

<input checked="" type="radio"/>	это принцип ООП согласно которому каждый объект может использоваться более чем в одной программе.	1053 / 17479
<input type="radio"/>	это механизм, который объединяет данные и методы, манипулирующие этими данными, и защищает и то и другое от внешнего вмешательства или неправильного использования.	1480 / 17479
<input type="radio"/>	это механизм, позволяющий создавать классы объектов на основе других классов, расширяя и частично изменяя их функциональность и набор атрибутов.	2597 / 17479
<input checked="" type="radio"/>	это процесс создания классов более высокого уровня, от которых можно создать более специфические сущности.	9523 / 17479
<input type="radio"/>	это процесс сокрытия компонентов данных и кода, реализующего функциональность, за некоторым интерфейсом	2796 / 17479

Комментировать вопрос (всего: 12)

Автор: [VS\\_Revan](#)

✓ Виртуальные методы...  
Необходимо продолжить утверждение (укажите все подходящие варианты).

<input type="checkbox"/>	служат исключительно для реализации полиморфизма	3013 / 17420
<input checked="" type="checkbox"/> ✓	позволяют наследникам переопределять методы родителя	13215 / 17420
<input type="checkbox"/>	не реализуются в базовом классе	4295 / 17420
<input checked="" type="checkbox"/> ✓	используются при реализации полиморфизма	13507 / 17420

✓ Для того, чтобы изменить значение переменной

```
Integer i = new Integer(9)
```

нужно выполнить след. код:

<input type="radio"/> i = 9;	3829 / 10376
<input type="radio"/> i.value = 9;	1580 / 10376
<input type="radio"/> i.value(9);	804 / 10376
<input checked="" type="radio"/> i = new Integer(9);	4122 / 10376

Пояснение: т.к. Integer не является примитивным типом в языке java, а так же в языке джава нету перегрузки операторов(как в языке C плюс плюс) то, правильный ответ, нужно выделить объекту новый участок памяти.

[Комментировать вопрос](#) (всего: 59)

Автор: [eddihoc](#)

✓ Драконы умеют летать (как, например, птицы) и ползать (как, например, ящерицы). С точки зрения ООП, примером чего является данная ситуация (выберите наиболее точный вариант)?

<input type="radio"/> Полиморфизм	3370 / 25959
<input type="radio"/> Композиция	1882 / 25959
<input checked="" type="radio"/> Множественное наследование	18060 / 25959
<input type="radio"/> Инкапсуляция	304 / 25959
<input type="radio"/> Наследование	2294 / 25959

[Комментировать вопрос](#) (всего: 29)

Автор: [admin](#)

✗ Какими бывают отношения между классами (укажите все подходящие варианты)?

<input checked="" type="checkbox"/> Агрегация	12967 / 17569
<input type="checkbox"/> Специализация	3029 / 17569
<input checked="" type="checkbox"/> Абстракция	5163 / 17569
<input checked="" type="checkbox"/> Ассоциация	9423 / 17569
<input checked="" type="checkbox"/> Наследование	16434 / 17569

Пояснение: Еще выделяют отношения: композиция и класс-метакласс.

[Комментировать вопрос](#) (всего: 7)

Автор: [VS\\_Revan](#)

✓ Вызов статического метода, это:

<input checked="" type="radio"/> Раннее связывание.	9438 / 10805
<input type="radio"/> Позднее связывание.	1330 / 10805

Пояснение: Раннее связывание это когда метод который будет известен во время компиляции. Позднее связывание это связывание проводимое во время выполнения программы, в зависимости от типа объекта. Позднее связывание также называют динамическим (dynamic) или связыванием на стадии выполнения (runtime binding).

[Комментировать вопрос](#) (всего: 3)

Автор: [berezoviy\\_vasil](#)




✗ Время жизни агрегируемого объекта меньше либо равно времени жизни агрегата.

<input type="radio"/> Да	9517 / 17461
<input checked="" type="radio"/> Нет	7822 / 17461

[Комментировать вопрос](#) (всего: 8)

Автор: [Evguenitch](#)

✔ Система обеспечивает принцип наследования в том случае, если:






 в любом месте, где допустимо использование объекта, принадлежащего классу-наследнику, точно также допустимо использование класса-предка.	6421 / 24435
 в системе имеется хотя бы один "наследник".	6019 / 24435
 ✔ в любом месте, где допустимо использование объекта, принадлежащего классу-предку, точно также допустимо использование класса-наследника.	11878 / 24435

*Пояснение: Принцип Барбары Лисков (принцип подстановочности). По этому принципу можно определить, обеспечивает ли система наследование.*

[Комментировать вопрос](#) (всего: 5) Автор: [JuliaUr](#)






✘ Какой принцип ООП необходимо использовать, чтобы заменить конструкции if-then-else в данном фрагменте кода:

```
if (animal.IsCat()) { /* код */ }
else if (animal.IsDog()) { /* код */ }
else if (animal.IsKoala()) { /* код */ }
. . .
else if (animal.isMouse()) { /* код */ }
```

 Композиция	2609 / 26128
 ✔ Полиморфизм	19191 / 26128
 Персистентность	1014 / 26128
 Инкапсуляция	1263 / 26128
 Агрегация	1911 / 26128





[Комментировать вопрос](#) (всего: 12) Автор: [admin](#)

✔ Метод, который определяет состояние объекта, не изменяя его.

 ✔ Селектор	7909 / 12211
 Модификатор	1144 / 12211
 Сеттер	1227 / 12211
 Конструктор	1730 / 12211
 Деструктор	149 / 12211



[Комментировать вопрос](#) (всего: 10) Автор: [MrQwert](#)

✘ Что из перечисленных относится к объектно-ориентированной абстракции?

 ✔ Абстракция данных.	7547 / 10883
 ✔ Абстракция функции.	6610 / 10883
 ✔ Абстракция структуры.	6185 / 10883
 Абстракция имени.	906 / 10883


[Комментировать вопрос](#) (всего: 4) Автор: [berezoviy\\_vasil](#)

✔ Наследование и композиция взаимоисключающие понятия. То есть при создании иерархии объектов используется либо наследование, либо композиция.

 Да	6989 / 24929
 ✔ Нет	17849 / 24929

[Комментировать вопрос](#) (всего: 12) Автор: [admin](#)

✘ Отношение композиции (укажите все подходящие варианты)

 ✔ является отношением has-a	8328 / 17130
---	--------------

<input type="checkbox"/>	синоним агрегации	3096 / 17130
<input type="checkbox"/>	является отношением is-a	2825 / 17130
<input checked="" type="checkbox"/>	частный случай агрегации	7774 / 17130
<input checked="" type="checkbox"/>	указывает на физическое включение	6390 / 17130
<input checked="" type="checkbox"/>	является отношением part-of	8048 / 17130
<input type="checkbox"/>	указывает на логическое включение	4484 / 17130

[Комментировать вопрос](#) (всего: 13) Автор: [Evguenitch](#)

✖ Выберите наиболее точное определение полиморфизму:

- ☐ это механизм, позволяющий создавать классы объектов на основе других классов, расширяя и частично изменяя их функциональность и набор атрибутов. 2825 / 17407
- ☐ это механизм, который объединяет данные и методы, манипулирующие этими данными, и защищает и то и другое от внешнего вмешательства или неправильного использования. 585 / 17407
- ☒ это принцип, согласно которому объекты, имеющие одинаковый интерфейс, могут вести себя по-разному 12803 / 17407
- ☐ это процесс сокрытия компонентов данных и кода, реализующего функциональность, за интерфейсом, не позволяющим пользователю исказить данные. 450 / 17407
- ☐ это принцип ООП, согласно которому каждый объект может использоваться более чем в одной программе. 729 / 17407

[Комментировать вопрос](#) (всего: 6)

Автор: [VS\\_Revan](#)

✖ Какие утверждения из перечисленных верны:

- ☐ Отношение "has-a" всегда реализуется с помощью наследования 6070 / 24278
- ☐ Отношение "has-a" требует как минимум две различные сущности 9298 / 24278
- ☐ Отношение "has-a" всегда реализуется с помощью полиморфизма 2994 / 24278
- ☒ Отношение "has-a" всегда подразумевает сильную связь между объектами 8682 / 24278
- ☐ Ничего из перечисленного 7358 / 24278

*Пояснение: 2 сущности не обязательны (пример - деревья с одним типом узлов).  
Отношение "has-a" может приводить к сильной связи между объектами, но это далеко не всегда так.*

[Комментировать вопрос](#) (всего: 14)

Автор: [admin](#)

✔ Как называется объект, задача которого состоит в реакции на возникновение определенного события?

- ☐ Repeater 327 / 19912
- ☐ Callback 3651 / 19912
- ☒ Observer 8852 / 19912
- ☐ Publisher 526 / 19912
- ☐ Trigger 6491 / 19912

[Комментировать вопрос](#) (всего: 18)

Автор: [admin](#)

✖ Выберите правильные утверждения по отношению к ad hoc полиморфизму:

- ☒ Поддерживается во многих языках посредством перегрузки функций и методов 5418 / 10494
- ☒ Позволяет определять функцию или тип данных обобщённо, так что значения обрабатываются идентично вне зависимости от их типа 3213 / 10494
- ☒ Позволяет давать одинаковые имена программным сущностям с различным поведением 5026 / 10494
- ☐ Доступен в нескольких объектно-ориентированных языках, где он часто идет под названием "джереник" или "шаблоны" 2850 / 10494
- ☐ Не поддерживается в языках Java и C++ 1499 / 10494

[Комментировать вопрос](#) (всего: 6)

Автор: [nvgup](#)

## ✓ Почему в некоторых языках программирования отказываются от поддержки множественного наследования (имеется в виду наследование реализации)?

- 👤 ✓ Из-за неоднозначности выбора поведения, в случае если суперклассы некоторого класса содержат методы с одинаковыми сигнатурами 17244 / 24527
- 👤 Множественное наследование практически никогда не используется, в отличие от обычного наследования от одного класса 1001 / 24527
- 👤 Поддержка множественного наследования ведет к большим потерям производительности, так как для каждого класса необходимо держать сильно-ветвящуюся иерархию его предков 49301 / 24527
- 👤 Множественное наследование невозможно реализовать с помощью таблицы виртуальных функций, поэтому требуются другие намного более сложные алгоритмы 1209 / 24527

[Комментировать вопрос](#) (всего: 15)

Автор: [admin](#)


## [Предложить свой вопрос](#)

[Share](#)[Tweet](#)

**Важно:** Ниже обсуждается только тест в целом: покрытие тематик, типы вопросов, пути улучшения теста и так далее.

Замечания, касающиеся отдельных вопросов теста, просьба оставлять в соответствующих топиках, которые доступны на странице результатов прохождения теста возле каждого вопроса.

Комментариев: 114 [↑ обновить](#)

 **Zyx** 05.07.2020 | 16:09:40 #

Фигня а не тест

[ответить](#)

 **esessfdser** 22.01.2020 | 16:55:38 #

на quizful есть вообще нормальные тесты? складывается впечатление, что это помойка какая-то, с кучей ошибочных вопросов, авторы которых ничего не понимают сами, но зачем-то придумывают вопросы

[ответить](#)

 **BUFFY** 12.02.2019 | 14:09:50 #

бесполезный тест, и уж точно не уровня джуниор. Я с твердым миддл уровнем половину тупо не помню, ибо на практике не нужно.

[ответить](#)

 **lazloSoot** 19.10.2018 | 19:02:53 #

худшего теста я еще не видел....

[ответить](#)

 **romanelli** 14.08.2018 | 21:07:48 #


куча неоднозначных и странных вопросов. в помойку

[ответить](#)

 **mamedova1999** 10.05.2018 | 23:28:56 #

Внимательность прочтения еще проверяют

[ответить](#)

 **mamedova1999** 10.05.2018 | 23:28:42 #

Немного впечатление, что тут опора не только на ООП

[ответить](#)

 **Entrery** 22.04.2018 | 10:13:52 #


вроде выбрал ООП в сишарпе, а тут вопросы по джаве...

[ответить](#)

 **inginiir** 26.05.2019 | 22:43:36 #

это не джава... какие деструкторы

[ответить](#)

 **Mishka\_Zabl** 04.04.2018 | 11:11:33 #

куча неоднозначных вопросов .Такие тесты только в школе давали нам в детстве . Бред.  
Много бесполезной терминологии .

[ответить](#)

 **artem\_pechenkin** 20.02.2018 | 05:06:07

#

Тест неадекватный

[ответить](#)

1 2 3 4 5 6 7 8 9

Добавить комментарий

Отправить