

В результате выполнения этого кода перед вызовом метода `System.gc()` без ссылки останется только один объект.

объект будет удален, id=1

Если закомментировать строку 1 и снять комментарий со строки 2, то перед выполнением `gc()` ссылки потеряют уже два объекта.

объект будет удален, id=1

объект будет удален, id=2

Если не вызвать явно метод `finalize()` суперкласса, то он не будет вызван автоматически. Еще одна опасность заключается в том, что если при выполнении данного метода возникнет исключительная ситуация, то она будет проигнорирована и приложение будет продолжать выполняться, что также представляет опасность для его корректной работы.

Задания к главе 4

Вариант А

Создать приложение, удовлетворяющее требованиям, приведенным в задании. Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы `equals()`, `hashCode()`, `toString()`.

1. Создать объект класса **Текст**, используя класс **Абзац**. Методы: дополнить текст, вывести на консоль текст, заголовок текста.
2. Создать объект класса **Автомобиль**, используя класс **Колесо**. Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.
3. Создать объект класса **Самолет**, используя класс **Крыло**. Методы: летать, задавать маршрут, вывести на консоль маршрут.
4. Создать объект класса **Беларусь**, используя класс **Область**. Методы: вывести на консоль столицу, количество областей, площадь, областные центры.
5. Создать объект класса **Планета**, используя класс **Материк**. Методы: вывести на консоль название материка, планеты, количество материков.
6. Создать объект класса **Звездная система**, используя классы **Планета**, **Звезда**, **Луна**. Методы: вывести на консоль количество планет в звездной системе, название звезды, добавление планеты в систему.
7. Создать объект класса **Компьютер**, используя классы **Винчестер**, **Дисковод**, **ОЗУ**. Методы: включить, выключить, проверить на вирусы, вывести на консоль размер винчестера.
8. Создать объект класса **Квадрат**, используя классы **Точка**, **Отрезок**. Методы: задание размеров, растяжение, сжатие, поворот, изменение цвета.
9. Создать объект класса **Круг**, используя классы **Точка**, **Окружность**. Методы: задание размеров, изменение радиуса, определение принадлежности точки данному кругу.
10. Создать объект класса **Котёнок**, используя классы **Животное**, **Кошка**. Методы: вывести на консоль имя, подать голос, рожать потомство (создавать себе подобных).

11. Создать объект класса **Наседка**, используя классы **Птица**, **Кукушка**. Методы: летать, петь, нести яйца, высиживать птенцов.
12. Создать объект класса **Текстовый файл**, используя класс **Файл**. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.
13. Создать объект класса **Одномерный массив**, используя класс **Массив**. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить).
14. Создать объект класса **Простая дробь**, используя класс **Число**. Методы: вывод на экран, сложение, вычитание, умножение, деление.
15. Создать объект класса **Дом**, используя классы **Окно**, **Дверь**. Методы: закрыть на ключ, вывести на консоль количество окон, дверей.
16. Создать объект класса **Роза**, используя классы **Лепесток**, **Бутон**. Методы: расцвести, завянуть, вывести на консоль цвет бутона.
17. Создать объект класса **Дерево**, используя классы **Лист**. Методы: зацвести, опадать листьям, покрыться инеем, пожелтеть листьям.
18. Создать объект класса **Пианино**, используя класс **Клавиша**. Методы: настроить, играть на пианино, нажимать клавишу.
19. Создать объект класса **Фотоальбом**, используя класс **Фотография**. Методы: задать название фотографии, дополнить фотоальбом фотографией, вывести на консоль количество фотографий.
20. Создать объект класса **Год**, используя классы **Месяц**, **День**. Методы: задать дату, вывести на консоль день недели по заданной дате, рассчитать количество дней, месяцев в заданном временном промежутке.
21. Создать объект класса **Сутки**, используя классы **Час**, **Минута**. Методы: вывести на консоль текущее время, рассчитать время суток (утро, день, вечер, ночь).
22. Создать объект класса **Птица**, используя класс **Крылья**. Методы: летать, питаться.
23. Создать объект класса **Тигр**, используя класс **Когти**. Методы: рычать, бежать, добывать пищу.
24. Создать объект класса **Гитара**, используя класс **Струна**. Методы: играть, натягивать струну.

Вариант В

Построить модель программной системы.

1. Система **Факультатив**. **Преподаватель** объявляет запись на **Курс**. **Студент** записывается на **Курс**, обучается и по окончании **Преподаватель** выставляет **Оценку**, которая сохраняется в **Архиве Студентов, Преподавателей и Курсов** при обучении может быть несколько.
2. Система **Платежи**. **Клиент** имеет **Счет** в банке и **Кредитную Карту (КК)**. **Клиент** может оплатить **Заказ**, сделать платеж на другой **Счет**, заблокировать **КК** и аннулировать **Счет**. **Администратор** может заблокировать **КК** за превышение кредита.
3. Система **Больница**. **Пациенту** назначается лечащий **Врач**. **Врач** может сделать назначение **Пациенту** (процедуры, лекарства, операции). **Медсестра** или другой **Врач** выполняют назначение. **Пациент** может быть выписан из **Больницы** по окончании лечения, при нарушении режима или при иных обстоятельствах.

4. Система **Вступительные экзамены**. **Абитуриент** регистрируется на **Факультет**, сдает **Экзамены**. **Преподаватель** выставляет **Оценку**. Система подсчитывает средний балл и определяет **Абитуриентов**, зачисленных в учебное заведение.
5. Система **Библиотека**. **Читатель** оформляет **Заказ** на **Книгу**. Система осуществляет поиск в **Каталоге**. **Библиотекарь** выдает **Читателю Книгу** на абонемент или в читальный зал. При невозвращении **Книги Читателем** он может быть занесен **Администратором** в «черный список».
6. Система **Конструкторское бюро**. **Заказчик** представляет **Техническое Задание (ТЗ)** на проектирование многоэтажного **Дома**. **Конструктор** регистрирует **ТЗ**, определяет стоимость проектирования и строительства, выставляет **Заказчику Счет** за проектирование и создает **Бригаду Конструкторов** для выполнения **Проекта**.
7. Система **Телефонная станция**. **Абонент** оплачивает **Счет** за разговоры и **Услуги**, может попросить **Администратора** сменить номер и отказаться от услуг. **Администратор** изменяет номер, **Услуги** и временно отключает **Абонента** за неуплату.
8. Система **Автобаза**. **Диспетчер** распределяет заявки на **Рейсы** между **Водителями** и назначает для этого **Автомобиль**. **Водитель** может сделать заявку на ремонт. **Диспетчер** может отстранить **Водителя** от работы. **Водитель** делает отметку о выполнении **Рейса** и состоянии **Автомобиля**.
9. Система **Интернет-магазин**. **Администратор** добавляет информацию о **Товаре**. **Клиент** делает и оплачивает **Заказ** на **Товары**. **Администратор** регистрирует **Продажу** и может занести неплательщиков в «черный список».
10. Система **Железнодорожная касса**. **Пассажир** делает **Заявку** на станцию назначения, время и дату поездки. Система регистрирует **Заявку** и осуществляет поиск подходящего **Поезда**. **Пассажир** делает выбор **Поезда** и получает **Счет** на оплату. **Администратор** вводит номера **Поездов**, промежуточные и конечные станции, цены.
11. Система **Городской транспорт**. На **Маршрут** назначаются **Автобус**, **Троллейбус** или **Трамвай**. Транспортные средства должны двигаться с определенным для каждого **Маршрута** интервалом. При поломке на **Маршрут** должен выходить резервный транспорт или увеличиваться интервал движения.
12. Система **Аэрофлот**. **Администратор** формирует летную **Бригаду** (пилоты, штурман, радист, стюардессы) на **Рейс**. Каждый **Рейс** выполняется **Самолетом** с определенной вместимостью и дальностью полета. **Рейс** может быть отменен из-за погодных условий в **Аэропорту** отлета или назначения. **Аэропорт** назначения может быть изменен в полете из-за технических неисправностей, о которых сообщил командир.
13. Система **Периодические издания**. **Читатель** может сделать **Заявку**, предварительно выбрав периодические **Издания** из списка. Система подсчитывает сумму для оплаты. **Читатель** оплачивает **заявку**. **Администратор** добавляет **Заявку** в «черный список», если **Клиент** не оплачивает её в определённый срок.

14. Система **Заказ гостиницы**. Клиент оставляет **Заявку** на **Номер**, указав количество мест в номере, класс апартаментов и время пребывания. **Администратор** рассматривает **Заявку**, подтверждает или отклоняет её. Результат просматривает **Клиент**. В случае подтверждения **Заявки** **Клиент** оплачивает услуги.
15. Система **Жилищно-коммунальные услуги**. **Квартиросъемщик** отправляет **Заявку**, в которой указывает род работ, масштаб и желаемое время выполнения. **Диспетчер** формирует соответствующую **Бригаду** и регистрирует её в **Плане работ**. **Диспетчер** может отклонить **Заявку** в случае занятости всех **Бригад**.
16. Система **Прокат автомобилей**. Клиент выбирает **Автомобиль** из списка доступных, заполняет форму **Заказа**, указывая паспортные данные, срок аренды. **Администратор** может отклонить **Заявку**, указав причины отказа. При подтверждении **Заявки** **Клиент** оплачивает **Заказ**. Система выписывает сумму. В случае повреждения **Автомобиля** **Клиентом** **Администратор** вносит соответствующие пометки.

Тестовые задания к главе 4

Вопрос 4.1.

Дан код:

```
class Base {}
class A extends Base {}
public class Quest{
    public static void main(String[] args){
        Base b = new Base();
        A ob = (A) b;
    } }
```

Результатом компиляции и запуска будет:

- 1) компиляция и выполнение без ошибок;
- 2) ошибка во время компиляции;
- 3) ошибка во время выполнения.

Вопрос 4.2.

Классы **A** и **Quest2** находятся в одном файле. Что необходимо изменить в объявлении класса **Quest2**, чтобы оно было корректным?

```
public class A{}
class Quest2 extends A, Object {}
```

- 1) необходимо убрать спецификатор **public** перед **A**;
- 2) необходимо добавить спецификатор **public** к **Quest2**;
- 3) убрать после **extends** один из классов;
- 4) класс **Object** нельзя указывать явно.

Вопрос 4.3.

Дан код:

```
class A {A(int i) {}} // 1
class B extends A {} // 2
```