

Главная

Объявления

Статьи

▼ Файлы

▶ 1_семестр_АЛГ

▶ 2_семестр_АЛГ(Delphi)


▶ 2_семестр_ООП_Java

▶ 3_семестр_TA

Книги

It is not so very important for a person to learn facts. For that he does not really need a college. He can learn them from books. The value of an education in a liberal arts college is not the learning of many facts, but the training of the mind to think something that cannot be learned from textbooks.

Albert Einstein



(немного правды жизни)

Д.з.№6 Наследование

Вы - разработчик информационной системы для американской военной базы. Как и всякая военная база, ваша база находится в пятиметровым забором. У каждой дыры в заборе установлен автоматизированный КПП. Вся информация о перемещении лично средств и материальных ценностей через эти КПП поступает в информационную систему базы. Ваша задача на сегодня - разработать перемещений через эти КПП - из внешнего мира на территорию базы и обратно.

1. Для этого вы разработали класс Base, а в нем - четыре переменные:

- int people_on_base; - количество людей на территории базы;
- int vehicles_on_base; - количество транспортных средств на территории базы;
- double petrol_on_base; - количество бензина на местной "заправке" (в литрах);
- double goods_on_base; - количество "груза" - различных материальных ценностей (в тоннах).

Так как база у нас одна, и несколько штук мы их создавать не собираемся - то все поля будут **статическими**. Начальные значения вышеперечисленных переменных можно задать произвольно по своему усмотрению.

2. Далее, Вам необходимо разработать иерархию классов транспортных средств:

- Базовый класс **Vehicle** ("транспортное средство"). Управляется одним водителем, ни груза, ни пассажиров не перевозит, не расходует топлива.
- Наследник класса Vehicle - класс **Bus** ("машина для перевозки личного состава" - автобус). Отличается от Vehicle тем, что имеет места для пассажиров.
- Наследник класса Vehicle - класс **Truck** ("грузовик"). Отличается от Vehicle тем, что может перевозить груз ограниченной массы.

3. В каждом классе имеются методы **arrive()** (приехать на базу) и **leave()** (покинуть территорию базы), которые соответствующим образом изменяют переменные базы (прибавляют или убавляют счетчики людей, машин, груза, бензина). При этом каждая машина должна заправить полный бак бензина, каждый автобус *старается* забрать с собой как можно больше людей, каждый грузовик - база потихоньку перемещается на новое место дислокации в соседней стране). При приезде на территорию базы нового транспорта (и водитель) высаживаются (добавляются "на баланс" базы), груз выгружается (но бензин не сливается).

4. **Интерфейс класса Vehicle** должен состоять из следующих методов:

- Vehicle(double petrol_amount, double tank_volume);
- double getTankVolume(); - узнать объем бензобака в литрах;
- double getPetrolAmount(); - узнать текущее количество топлива в баке;
- void arrive(); - приехать на базу;
- boolean leave(); - залить полный бак и покинуть базу; в случае невозможности уехать возвращает false;

5. **Интерфейс класса Bus** расширяет интерфейс класса Vehicle следующими операциями:

- Bus(int people, int max_people, double petrol, double max_petrol);
- int getPeopleCount(); - получить текущее количество пассажиров (водитель в это число не входит);
- int getMaxPeople(); - получить количество пассажирских мест;

- операции arrive() и leave(), естественно, **переопределяются** и работают по-другому - выгружают/увозят, помимо водителя и бензина.

6. **Интерфейс класса Truck** расширяет интерфейс класса Vehicle следующими операциями:

- Truck(double load, double max_load, double petrol, double max_petrol);
- double getCurrentLoad(); - получить текущую загрузенность машины (масса груза в тоннах);
- double getMaxLoad(); - узнать максимальную вместительность (грузоподъемность) машины;

- операции arrive() и leave(), естественно, переопределяются и работают по-другому - выгружают/увозят, помимо водителя и бензина.

Технические подробности

- Отдельные классы удобно помещать в отдельные файлы.
- Если у вас несколько классов в одном файле - они должны быть расположены "рядом". Помещать одни классы внутри других - только один из них может быть "public".
- "Балансовые" статические переменные XXXX_on_base, естественно, не могут быть меньше нуля, и ваши классы (вернее, методы) должны следить за этим.
- По приезде на базу машина добавляет к "балансу" базы себя (vehicles_on_base), своего водителя (people_on_base), всех с собой.
- Если при попытке уехать (leave()) машина не может залить полный бак или на базе не осталось ни одного человека (нет водителей) - возвращает false. Замечание: будем считать, что машине не нужен *конкретно ее водитель* - управлять ей может любой человек.
- Автобус и грузовик при попытке уехать ведут себя точно так же, как Vehicle, но дополнительно - пытаются увезти полный бак бензина соответственно. Если на базе осталось людей или груза меньше, чем вмещает данная машина - она забирает столько, сколько возможно.

Примеры

Вход	Выход
<pre>Vehicle v = new Vehicle(19.5, 60);</pre>	<pre>19.5 60.0</pre>

// get System.out.println(v.getPetrolAmount()); System.out.println(v.getTankVolume()); System.out.println();	Initially: 49 99 1000.0 2000.0
// init Base.vehicles_on_base = 49; Base.people_on_base = 99; Base.petrol_on_base = 1000.0; Base.goods_on_base = 2000.0;	Arrived: 50 100 1000.0 2000.0
System.out.println("Initially:"); System.out.println(Base.vehicles_on_base); System.out.println(Base.people_on_base); System.out.println(Base.petrol_on_base); System.out.println(Base.goods_on_base); System.out.println();	Left: 49 99 959.5 2000.0
// arrive v.arrive(); System.out.println("Arrived:");	
System.out.println(Base.vehicles_on_base); // arrived System.out.println(Base.people_on_base);	
System.out.println(Base.petrol_on_base); // same System.out.println(Base.goods_on_base); System.out.println();	
// leave v.leave(); System.out.println("Left:");	
System.out.println(vehicles_on_base); // left System.out.println(people_on_base);	
// should be 1000-(60-19.5) System.out.println(petrol_on_base); // changed!	
System.out.println(goods_on_base); // same System.out.println();	

Контрольные вопросы

- Какое ключевое слово языка Java позволяет указать, что определяемый класс - наследник какого-то другого класса?
- Как соотносятся между собой размеры экземпляров базового и производного классов? Почему именно так?
- Перечислите все поля, которые хранит в себе объект класса Bus.
- Создается объект производного класса. В этом классе есть два конструктора - собственный и унаследованный от класса-оба - то в каком порядке?) будет вызван? Почему именно так? (проиллюстрировать примером кода)
- Что означает ключевое слово **super**? Приведите типичный пример его использования.
- Что такое переопределение методов (в производном классе)? Продемонстрировать его полезность на примере кода.
- Из каких частей программы видны, а из каких - не видны поля и методы класса с модификатором доступа private? protected?
- Как из метода производного класса вызвать метод базового класса с тем же именем? (привести пример кода)

Скачать [Base_Tests.java](#)

Комментарии

У вас нет прав для добавления комментариев.