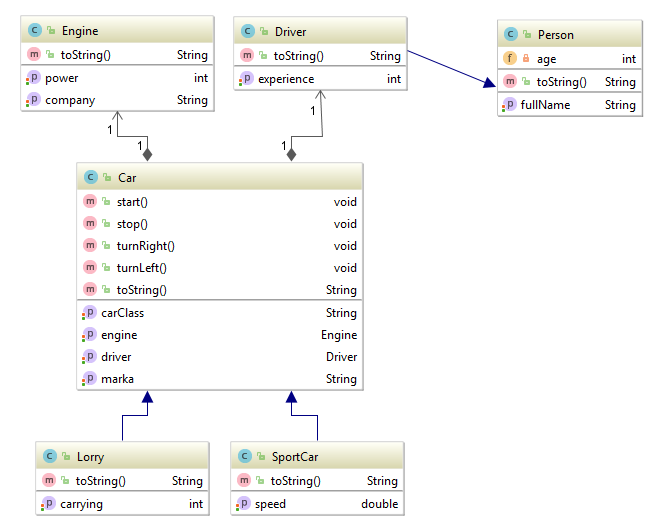
# Задания

1. Класс Phone.  
   a) Создайте класс Phone, который содержит переменные number, model и weight.  
   б) Создайте три экземпляра этого класса.   
   в) Выведите на консоль значения их переменных.   
   г) Добавить в класс Phone методы: receiveCall, имеет один параметр – имя звонящего. Выводит на консоль сообщение “Звонит {name}”. getNumber – возвращает номер телефона. Вызвать эти методы для каждого из объектов.  
   д) Добавить конструктор в класс Phone, который принимает на вход три параметра для инициализации переменных класса - number, model и weight.   
   е) Добавить конструктор, который принимает на вход два параметра для инициализации переменных класса - number, model.   
   ж) Добавить конструктор без параметров.  
   з) Вызвать из конструктора с тремя параметрами конструктор с двумя.   
   и) Добавьте перегруженный метод receiveCall, который принимает два параметра - имя звонящего и номер телефона звонящего. Вызвать этот метод.  
   к) Создать метод sendMessageс аргументами переменной длины. Данный метод принимает на вход номера телефонов, которым будет отправлено сообщение. Метод выводит на консоль номера этих телефонов.  
   л) Изменить класс Phone в соответствии с концепцией JavaBean.
2. Создать класс Person, который содержит:   
   a) поля fullName, age.   
   б) методы move() и talk(), в которых просто вывести на консоль сообщение -"Такой-то  Person говорит".   
   в) Добавьте два конструктора  - Person() и Person(fullName, age).  
   Создайте два объекта этого класса. Один объект инициализируется конструктором Person(), другой - Person(fullName, age).
3. Создать класс "Матрица". Класс должен иметь следующие поля:  
   1) двумерный массив вещественных чисел;  
   2) количество строк и столбцов в матрице.  
     
   Класс должен иметь следующие методы:  
   1) сложение с другой матрицей;  
   2) умножение на число;  
   3) вывод на печать;   
   4) умножение матриц - по желанию.
4. Определить класс Reader, хранящий такую информацию о пользователе библиотеки: ФИО, номер читательского билета, факультет, дата рождения, телефон. Методы takeBook(), returnBook().  
   Разработать программу, в которой создается массив объектов данного класса. Перегрузить методы takeBook(), returnBook():  
   - takeBook, который будет принимать количество взятых книг. Выводит на консоль сообщение "Петров В. В. взял 3 книги".  
   - takeBook, который будет принимать переменное количество названий книг. Выводит на консоль сообщение "Петров В. В. взял книги: Приключения, Словарь, Энциклопедия".  
   - takeBook, который будет принимать переменное количество объектов класса Book (создать новый класс, содержащий имя и автора книги). Выводит на консоль сообщение "Петров В. В. взял книги: Приключения, Словарь, Энциклопедия".  
    Аналогичным образом перегрузить метод returnBook(). Выводит на консоль сообщение "Петров В. В. вернул книги: Приключения, Словарь, Энциклопедия". Или "Петров В. В. вернул 3 книги".
5. Даны два целых числа A и В. Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае. Использовать рекурсию. (Не обязательно)
6. Создайте пример наследования, реализуйте класс Student и класс Aspirant, аспирант отличается от студента наличием некой научной работы.  
   а) Класс Student содержит переменные: String firstName, lastName, group. А также double averageMark, содержащую среднюю оценку.  
   б) Создать переменную типа Student, которая ссылается на объект типа Aspirant.  
   в) Создать метод getScholarship() для класса Student, который возвращает сумму стипендии. Если средняя оценка студента равна 5, то сумма 100 грн, иначе 80. Переопределить этот метод в классе Aspirant.  Если средняя оценка аспиранта равна 5, то сумма 200 грн, иначе 180.  
   г) Создать массив типа Student, содержащий объекты класса Student и Aspirant. Вызвать метод getScholarship() для каждого элемента массива.
7. Создать класс Car в пакете by.academy.classwork.vehicles, Engine в пакете by.academy.classwork.details и Driver в пакете by.academy.classwork.professions. Класс Driver содержит поля - ФИО, стаж вождения.  
   Класс Engine содержит поля - мощность, производитель.  
   Класс Car содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа Driver, мотор типа Engine. Методы start(), stop(), turnRight(), turnLeft(), которые выводят на печать: "Поехали", "Останавливаемся", "Поворот направо" или "Поворот налево". А также метод printInfo(), который выводит полную информацию об автомобиле, ее водителе и моторе.   
   Создать производный от Car класс  - Lorry (грузовик), характеризуемый также грузоподъемностью кузова. Создать производный от Car класс - SportCar, характеризуемый также предельной скоростью. Пусть класс Driver расширяет класс Person.
8. Создать класс Animal и расширяющие его классы Dog, Cat, Horse. Класс Animal содержит переменные food, location и методы makeNoise, eat, sleep. Метод makeNoise, например, может выводить на консоль "Такое-то животное спит". Dog, Cat, Horse переопределяют методы makeNoise, eat. Добавьте переменные в классы Dog, Cat, Horse, характеризующие только этих животных. Создайте класс Ветеринар, в котором определите метод void treatAnimal(Animal animal). Пусть этот метод распечатывает food и location пришедшего на прием животного. В методе main создайте массив типа Animal, в который запишите животных всех имеющихся у вас типов. В цикле отправляйте их на прием к ветеринару.
9. Заполнить табличку:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | private | default | protected | public |
| 1. | Один и тот же класс |  |  |  |  |
| 2. | Подкласс, находящийся в том же пакете, что и суперкласс |  |  |  |  |
| 3. | Класс из того же самого пакета, не являющийся подклассом. |  |  |  |  |
| 4. | Подкласс, находящийся в другом пакете. |  |  |  |  |
| 5. | Класс из другого пакета, не являющийся подклассом данного. |  |  |  |  |

1. Создать классы для заполнения предыдущей таблицы. (Не обязательно)
2. Создайте суперкласс Shape и его наследники Circle, Rectangle. Класс Shape содержит абстрактный метод draw() и переменную хранящую цвет. Классы Circle, Rectangle содержат координаты точек. Создать массив содержащий эти фигуры. В цикле нарисовать их (вызвать метод draw). Добавить метод equals() для классов Shape, Circle, Rectangle.
3. а) Создать класс Товар, имеющий переменные имя, цена, рейтинг.   
   б) Создать класс Категория, имеющий переменные имя и массив товаров. Создать несколько объектов класса Категория.   
   в) Создать класс Basket, содержащий массив купленных товаров.   
   г) Создать класс User, содержащий логин, пароль и объект класса Basket. Создать объект класса User. (Интернет магазин)