Лист с заданиями к лекции №3

Теоретические задания

1. ООП в Java
2. Ограничение доступа
3. Оболочки примитивов
4. Autoboxing/Autounboxing
5. Преобразование типов
6. Класс Object
7. Абстрактные классы и интерфейсы
8. Слабая связанность
9. String

Практические задания

Требования к выполнению всех заданий:

1) Исходные .java файлы должны быть «вкомитаны» в GIT в соответствующую ветку
2) Использовать для выполнения IDE
3) Для вывода сообщения использовать - System.out.println();
4) Для генерации случайных целых чисел использовать -
   (new java.util.Random()).nextInt(МАКС_ЧИСЛО);

Задание 1 (сложность 2)

Написать программу (только одну из 4-х):

1) выводящую на экран случайно сгенерированное трёхзначное натуральное число и его наибольшую цифру
2) выводящую на экран 3 случайно сгенерированных трёхзначных числа и сумму их первых цифр
3) выводящую на экран 3 случайно сгенерированных трёхзначных числа и разницу между числом получившимся методом последовательной записи 1-го и 2-го числа и третьим числом
4) выводящую на экран случайно сгенерированное трёхзначное натуральное число и сумму его цифр

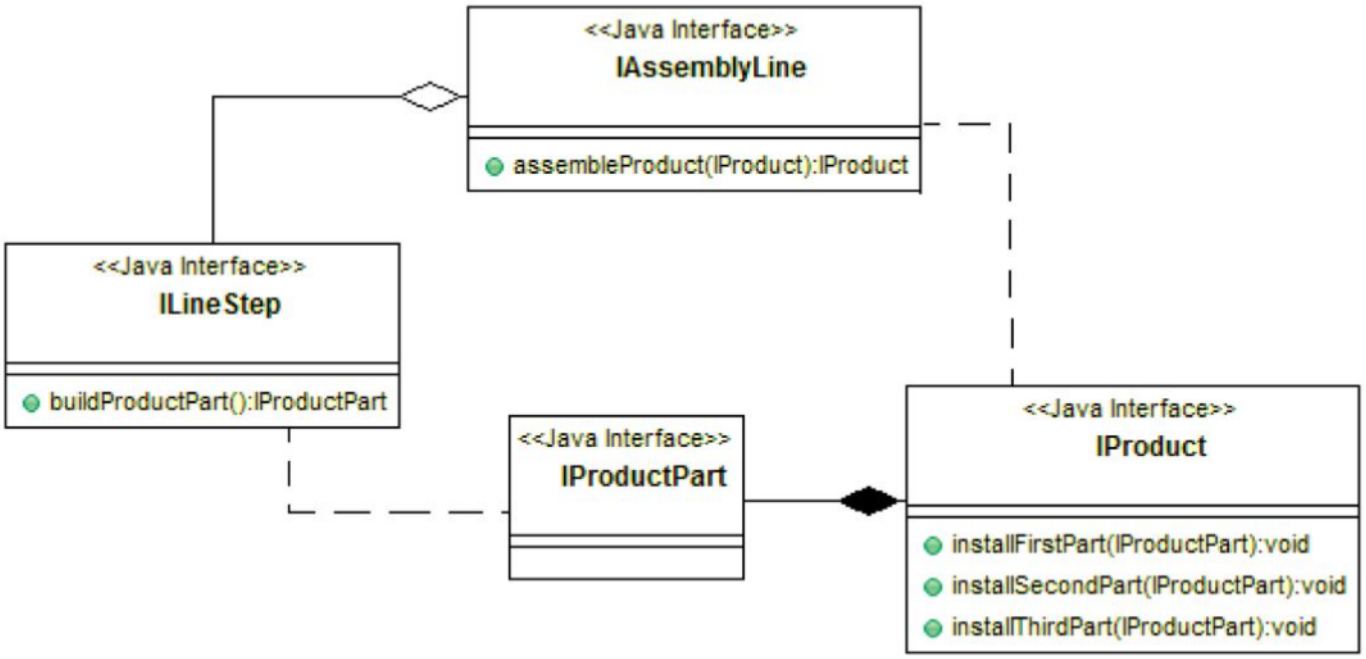
Задание 2 (сложность 3)

Написать программу (только одну из 3-х):

1) Содержащую иерархии цветов для цветочного магазина. Собрать букет и определить его стоимость.
2) Содержащую иерархии товаров для склада. Заполнить склад до предела и высчитать вес хранимого товара.
3) Содержащую иерархию сотрудников фирмы. Укомплектовать фирму по производству программного обеспечения и рассчитать общую месячную получку сотрудников.

Задание 3 (сложность 4)

Реализовать сборочную линию объектов, которая реализует интерфейсы, приведенные на следующей диаграмме:



Задача сборочной линии - собирать сложные продукты, состоящие из 3-х составляющих. Для установки каждой части использоваться специализированный класс (реализующий ILineStep) который поставляет необходимую часть через метод buildProductPart(). Для сборки продукта, его заготовка передается в метод assembleProduct(). Внутри метода, линия получает 3 необходимых составляющих и «устанавливает» их в соответствующие методы полученного продукта. После завершения «сборки», готовый продукт возвращается из метода assembleProduct().

Варианты сборочных линий:

1) Автомобили (кузов, шасси, двигатель)
2) Танки (корпус, двигатель, башня)
3) Ноутбуки (корпус, материнская плата, монитор)
4) Автоматические шариковые ручки (корпус, пружина, стержень)
5) Очки (корпус, линзы, дужки)

Реализовать отдельный поверочный класс с методом main() который сконструирует сборочную линию со всеми ее шагами, и протестирует ее запуском одного продукта на сборку.

Весь процесс сборки должен детально выводиться пользователю на консоль.

Требования к выполнению:

1) В конструктор сборочной линии должны передаваться 3 ссылки на необходимые сборочные шаги
2) Для разработки использовать IDE

3) Для вывода сообщений использовать - System.out.println(“ТЕКСТ_СООБЩЕНИЯ”);

Задание 4 (сложность 6)

Данное задание будет дорабатываться на протяжении всех курсов.

1. Электронный администратор гостиницы.

Программа должна предоставлять возможность:

- Поселить в номер
- Выселить из номера
- Изменить статус номера на ремонтируемый/обслуживаемый
- Изменить цену номера или услуги
- Добавить номер или услугу

2. Электронный книжный магазин.

Программа должна предоставлять возможность:

- Списать книгу со склада (перевести в статус “отсутствует”)
- Создать заказ
- Отменить заказ
- Изменить статус заказа (новый, выполнен, отменен)
- Добавить книгу на склад (закрывает все запросы книги и меняет ее статус на “в наличии”)
- Оставить запрос на книгу

Дополнение:

Все доступные книги заранее определены. Они быть “в наличии” или “отсутствует”. Под “Оставить запрос на книгу” подразумевается создание запроса на книгу, что сейчас “отсутствует”.

При создании заказа с книгой, которой нет в наличии - автоматически создается запрос на эту книгу. Заказ не может быть завершен, до тех пор, пока запрос книги не будет выполнен.

3. Электронный администратор автосервиса

Программа должна предоставлять возможность:

- Добавить/удалить мастера
- Добавить/удалить место в гараже
- Добавить/удалить/закрыть/отменить/ заказ
- Сместить время выполнения заказов (из-за задержек в выполнении текущего)

Требование к задаче:

1) К программе **НЕ** реализовывать консольный пользовательский интерфейс. Работоспособность программы проверять из тестового класса с методом main
2) К программе должна быть создана диаграмма классов
3) Программа должна соответствовать принципам ООП и паттернам «Сильное сцепление» и «Слабая связанность»
4) Для вывода результатов работы использовать System.out.println(message)
5) Исходные .java файлы должны быть «вкомитаны» в GIT в соответствующую ветку