

Домашнее задание

по теме «Инкапсуляция. Модификаторы доступа в Java».

Формулировка задания:

Основная структура программы должна включать следующие элементы:

Car

Базовый автомобиль обладает следующими свойствами: маркой (строка), моделью (строка), годом выпуска (int), мощностью в лошадиных силах (int), ускорением (int), подвеской (int) и долговечностью (int).

Каждый отдельный тип автомобиля дополняет эти свойства. Вот типы:

1. PerformanceCar – гоночный автомобиль.

Имеет дополнения addOns (массив строк, по умолчанию – пустой)

Увеличенная мощность двигателя на 50%.

Уменьшенная подвеска на 25%.

2. ShowCar – спортивная машина. Looking cool there, bro.

Включает поле stars (int). (по умолчанию – 0), поле для оценки популярности автомобиля.

Race

Гонка имеет следующие свойства: длина (int), маршрут (строка), призовой фонд (int) и участники (коллекция автомобилей),

- CasualRace – обычная гонка.
- DragRace – гонка за самый мощный двигатель. Идеальное переключение передач — залог победы.
- DriftRace – дрифтовая гонка.

Garage

- Garage – место, где остаются все автомобили, когда они не участвуют в

гонках. Гараж также предоставляет возможность модифицировать припаркованный автомобиль. Включает `parkedCars` (массив объектов типа `Car`).

Каждый из представленных классов должен включать:

1. Конструктор пустой и с параметрами;
2. Переопределенный метод `toString()`;
3. Геттеры и сеттеры для полей. Обратите внимание, что поля требуется сделать `private`;
4. У классов переопределены методы `equals()` и `hashCode()`.

Работу с классами проверить в методе `main` класса `App`.

Программа реализуется в отдельной ветке `git homeworks/homework09`. При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл `.gitignore`.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

Дополнительно:

Следующие доработки для исходной программы сохранить в ветке `git homeworks/homework09` отдельным коммитом/коммитами:

1. Дополнить модель предметной области. Создать 2 дополнительных специальных класса для гонок:

`TimeLimitRace`

Класс имеет дополнительное поле `goldTime (int)`.

`CircuitRace`

Класс имеет дополнительное поле `laps (int)`.

Оба новых класса имеют дополнительный параметр, помимо наследуемых у класса `Race`. Параметр вводится в качестве последнего параметра при открытии одного из этих типов гонок.

2. Реализовать ввод параметров задачи из файла.

3. Реализовать вывод результата задачи в файл.

Планируемый результат:

1. Ссылка на программу в репозитории GitHub;
2. Отчёт со скринами выполнения задач - постановка задачи, код задачи и результат в консоли IntelliJ Idea.

Описания плана работы:

Выполнение задания в соответствии с формулировкой требований к задаче. Задание является первым шагом к объектному моделированию предметной области с последующей реализацией на Java.

Перечень инструментов, необходимых для реализации деятельности:

Персональный компьютер, JDK 17 (либо OpenJDK 17), IntelliJ Idea для разработки на Java, GIT, Tortoise GIT.