**Урок 1. Понятие класса и объекта**

|  |
| --- |
|  |

|  |
| --- |
| Все задания про кошек связаны между собой. Выполните их в проекте Cat, в директории 03\_ObjectsAndClasses вашего репозитория Git. |

**Цель задания**

Научиться создавать объекты класса и вызывать методы объекта.

**Что нужно сделать**

Создайте 5–7 кошек, вызовите у них различные методы:

1. Выведите в консоль вес созданных кошек.
2. Покормите двух кошек, после этого распечатайте их вес (убедитесь, что вес изменился).
3. Перекормите кошку, чтобы ее статус стал Exploded.
4. «Замяукайте» кошку до статуса Dead.
5. Отправьте в репозиторий Git коммит с выполненным заданием.

**Рекомендации**

Чтобы перекормить или «замяукать» кошку, используйте цикл while с условием проверки статуса кошки.

**Критерии оценки**

«Зачёт» — выполнены условия задания и отправлена ссылка на коммит в форме ответа.  
«Незачёт» — задание не выполнено.

**Урок 2. Метод, параметры, return**

**Цель задания**

Научиться писать методы класса с возвратом значений и без возврата.

**Что нужно сделать**

1. Создайте в классе Cat метод, который возвращает сумму съеденной еды текущей кошки.
2. Создайте в классе Cat метод «Сходить в туалет» pee(), который уменьшает вес кошки и что-нибудь печатает.
3. Отправьте в репозиторий Git коммит с выполненным заданием.

**Рекомендации**

Протестируйте верную работу метода возврата съеденной еды:

* Создайте кошку.
* Покормите кошку кормом весом 150.00 граммов.
* Вызовите метод pee() несколько раз.
* Распечатайте значение съеденного корма, в консоль должно быть выведено значение: 150.00 граммов.
* Отправьте в репозиторий Git коммит с выполненным заданием.

**Критерии оценки**

«Зачёт» — выполнены условия задания и отправлена ссылка на коммит в форме ответа.  
«Незачёт» — задание не выполнено.

**Урок 3. Статические методы и переменные**

**Цель задания**

Научиться создавать статическую переменную и управлять ей, используя условный оператор if.

**Что нужно сделать**

1. Создайте у класса Cat статическую переменную count, которая увеличивается на 1, если кошку создали, и убывает на 1, если кошка взорвалась или умерла.
2. Создайте статический метод getCount(), который возвращает количество кошек.
3. Отправьте в репозиторий Git коммит с выполненным заданием.

|  |
| --- |
|  |

|  |
| --- |
| **Дополнительное задание\***  Придумайте способ, который поможет проверить, жива ли кошка, на основе значения её веса. Используя это, запретите неживой кошке есть, пить, ходить в туалет, выполняя проверку веса внутри методов, которые изменяют вес кошки. |

**Критерии оценки**

«Зачёт» — создана переменная и количество кошек считается верно.  
«Незачёт» — задание не выполнено.

**Урок 4. Константы и Enum**

**Цель задания**

Научиться создавать константы в классе и Enum.

**Что нужно сделать**

1. Создайте в классе Cat константы: «количество глаз», «минимальный вес», «максимальный вес».
2. Создайте enum Color с окрасами кошек.
3. Отправьте в репозиторий Git коммит с выполненным заданием.

**Критерии оценки**

«Зачёт» — созданы в классе Cat константы и Enum с окрасом в отдельном файле.  
«Незачёт» — задание не выполнено.

**Урок 5. Создание объектов и конструктор**

**Цель задания**

Научиться создавать объекты, используя перегруженные конструкторы.

**Что нужно сделать**

1. Сделайте в классе Cat ещё один конструктор, чтобы можно было создать кошку с весом, переданным в конструктор.
2. Создайте в главном классе Loader метод генерации котёнка private static Cat getKitten(). Он должен возвращать кошку с весом 1100.00 грамм с помощью конструктора, в который передаём вес кошки. Используйте описание из видеоурока.
3. Создайте три объекта класса Cat в методе main(), используя метод getKitten().
4. Отправьте в репозиторий Git коммит с выполненным заданием.

**Критерии оценки**

«Зачёт» — создан перегруженный конструктор и написан метод создания кошки в Loader.  
«Незачёт» — задание не выполнено.

**Урок 6. Инкапсуляция, геттеры и сеттеры**

**Цель задания**

Научиться создавать методы возврата (геттеры) и методы установки значения (сеттеры).

**Что нужно сделать**

1. В классе Car проекта RoadController создайте геттеры и сеттеры для всех переменных класса.
2. Создайте в классе Cat переменную, в которой хранится окрас. Напишите геттер и сеттер для окраса.
3. Отправьте в репозиторий Git коммит с выполненным заданием.

**Рекомендации**

К именам методов, возвращающих значения полей типа boolean, добавляют префиксы is/has. Например:

* Метод, возвращающий результат проверки состояния, например «Жива ли кошка?», можно назвать isAlive(). Если кошка жива, ожидаем возврат значения true.
* Метод, возвращающий результат проверки наличия у объекта свойства, например «Есть ли у кошки хвост?», можно назвать hasTail(). Если кошка с хвостом, ожидаем возврат значения true.

**Критерии оценки**

«Зачёт» — созданы геттеры и сеттеры в классе Car, а также геттер и сеттер для окраса в классе Cat в соответствии с рекомендациями.  
«Незачёт» — задание не выполнено.

**Урок 7. Копирование объектов**

**Цель задания**

Научиться копировать параметры объекта с помощью геттеров и сеттеров.

**Что нужно сделать**

1. Сделайте метод создания «глубокой» копии кошки. Это означает, что все параметры одной кошки равны параметрам другой. Используйте для этого метод копирования или конструктор копирования.
2. Отправьте в репозиторий Git коммит с выполненным заданием.

**Рекомендации**

Читайте параметры первого объекта геттерами и переносите их в другой с помощью сеттеров.

**Критерии оценки**

«Зачёт» — написан метод копирования или конструктор копирования.  
«Незачёт» — задание не выполнено.

**Урок 8. Внешние библиотеки классов**

**Цель задания**

Научиться подключать внешние библиотеки в проект с помощью среды разработки IDEA и  использовать классы библиотеки.

**Что нужно сделать**

Повторите действия из видео:

* создайте новый проект,
* подключите к нему библиотеку airport.jar,
* получите список самолётов,
* распечатайте количество (число) самолётов в аэропорту,
* отправьте в репозиторий Git коммит с выполненным заданием.

**Критерии оценки**

«Зачёт» — при запуске проекта в консоль выводится число самолётов в аэропорту.  
«Незачёт» — задание не выполнено.

**Урок 9. Создание JAR-файлов**

**Цель задания**

Научиться создавать исполняемый файл jar из своего проекта.

**Что нужно сделать**

1. Упакуйте приложение с экспериментами над кошками в JAR-файл. Убедитесь, что он запускается.
2. Приложите jar-файл к вашему ответу. Отправлять jar-файл в репозиторий не требуется.

**Критерии оценки**

«Зачёт» — прислан файл jar.  
«Незачёт» — задание не выполнено.