

Частное учреждение образования
Колледж бизнеса и права

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____Е.В. Паскал
«___»_____2021

Специальность: 2-40 01 01 «Программное обеспечение информационных технологий»	Дисциплина: _____ «Основы кроссплатформенного программирования»
---	--

ЛАБОРАТОРНАЯ РАБОТА № 3
Инструкционно-технологическая карта

Тема: «Разработка классов в языке Java»

Цель: Научиться создавать классы, методы и экземпляры классов, а также обращаться к ним в языке Java

Время выполнения: 2 часа

СОДЕРЖАНИЕ РАБОТЫ

1. Контрольные вопросы.
2. Теоретические сведения для выполнения работы.
3. Порядок выполнения работы.
4. Домашнее задание.
5. Литература.

1. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Структура класса в Java.
2. Как создаются методы класса?
3. Каким образом создается экземпляр класса?
4. Синтаксис класса.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ РАБОТЫ

В этом лабораторной работе будет рассмотрено создание класса на языке java и последующее обращение к этому классу, то есть мы научимся основам работы с классами.

Создание класса в java

Создавать класс мы будем в программе Eclipse, но также можно работать в любой другой IDE или в блокноте. В качестве примера мы будем создавать класс, который описывает характеристики и поведение кота.

Создадим в нашем проекте новый класс под именем Cat. Для этого в меню Eclipse выберем File -> new -> class и, в открывшемся окне, введем имя класса – Cat. Обратите внимание, что галочка public static void main (String[] args) должна быть снята! Затем нажмем Finish

В итоге, у нас создался новый класс.

Структура класса в java

Класс в java оформляется с помощью следующей конструкции:

```
public class Cat { }
```

Рассмотрим ее составляющие:

public – модификатор доступа к классу, в данном случае он нам говорит, что этот класс будет доступен не только данному классу, но и другим. В java существуют и другие модификаторы, но об этом поговорим позднее.

class – ключевое слово, говорящее о том, что это класс.

Cat– имя класса. Имена классов принято писать с заглавной буквы.

{ } – фигурные скобки, между которыми разместится тело нашего класса.

В предыдущей лабораторной работе мы упоминали, что классы в java состоят из атрибутов и методов, присвоим некоторые и нашему коту.

Атрибуты класса Cat

Атрибутами кота могут быть: имя, вес, окраска. Атрибуты это переменные, которые объявляются следующим образом:

```
private int weight; // вес кота
private String name; // имя кота
private String color; //окрас кота
```

private –здесь опять же указывает на права доступа. К переменной, в данном случае, можно обращаться только из ее класса, чтобы классы извне не могли ее изменить.

int, String – это типы данных. В данном случае, вес будет задан при помощи целого числа – int, а имя и цвет при помощи символьной строки String.

После объявления каждого атрибута должна ставиться точка с запятой;

При помощи // в коде программы оформляются комментарии, которые являются правилом хорошего тона.

Методы класса Cat

Пусть наш кот умеет есть, спать и разговаривать. Опишем это поведение с помощью методов.

```
//кот ест public void eat(){ System.out.print("Eating...\n"); }
```

public void eat()– сигнатура метода, его определение. В данном случае она нам говорит о том, что:

□ метод eat() доступен для других классов – public;

□ метод `eat()` не возвращает никаких данных – `void`.

`eat` – имя метода. Имена методов, в отличие от имен классов, принято писать с маленькой буквы. На конце каждого метода после имени всегда идут круглые скобки (), пустые или содержащие параметры (об этом позднее).

После сигнатуры метода идут фигурные скобки { }. В них содержится тело метода. Тело нашего метода содержит просто вывод сообщения о том, что кот кушает — `"Eating...\n"`. За вывод текстового сообщения отвечает метод `System.out.print("Eating...\n")`.

`\n` — символ перевода строки, чтобы при выводе наших сообщений все они не писались в одну строку.

Аналогично предыдущему методу опишем спящего кота:

```
//кот спит
public void sleep(){
System.out.print("Sleeping zz-z-z-z...\n"); }
И немного иначе опишем метод – кот говорит:
//кот говорит
public String speak(String words){
String phrase = words + "...mauu...\n"; return phrase; }
```

В отличие от предыдущих методов, этот метод возвращает значение и имеет входные параметры. Давайте подробнее рассмотрим сигнатуру метода `public String speak(String words)`:

`public`- метод `speak()` доступен для других классов;

`String` -тип значения, которое возвращает метод. В предыдущих случаях ключевое слово `void` указывало на то, что метод ничего не возвращает. В данном случае `String` указывает на то, что метод возвращает значение типа строка.

В процессе своей работы, метод выполняет определенные действия над данными, иногда необходимо, чтобы результат этих действий был передан для дальнейшей обработки другим классам, в этом случае метод передает (возвращает) этот результат. Эти возвращаемые данные, относятся к какому-либо типу, в нашем примере это тип символьной строки - `String`.

Возвращающие методы должны содержать в своем теле ключевое слово `return`, которое указывает на то, что именно возвращает данный метод. В нашем случае это переменная `phrase`.

`speak`- имя метода.

`(String words)` — входные параметры. Входные параметры, это какие-либо данные, которые передаются из других классов и, которые метод должен обработать. Наш метод получает в качестве входных данных строку в виде переменной `words`, к этой строке дописывает «...mauu...» и возвращает то, что получилось.

В итоге класс `Cat` выглядит следующим образом:

```
public class Cat {
```

```

private int weight; // вес кота
private String name; // имя кота
private String color; //окрас кота
//кот ест
public void eat(){
System.out.print("Eating...\n"); }
//кот спит
public void sleep(){
System.out.print("Sleeping zz-z-z-z...\n"); }
//кот говорит
public String speak(String words){
String phrase = words + "...mauu...\n"; return phrase; } }

```

Обращение к классу в Java

Далее рассмотрим обращение к классу Cat. И то, как его использовать в работе.

Создадим класс Hello: File -> new -> class, ввести в открывшемся окне имя класса и, в этот раз, обязательно установить галочку public static void main(String[] args).

public static void main(String[] args) – сигнатура метода main(). Программа состоит из нескольких классов, но только один из классов может содержать метод main(). Метод main(), это так называемая точка входа в программу. Без этого метода мы не сможем стартовать программу. Теперь, вместо приветствия миру, этот метод будет содержать код для работы с классом Cat.

Пишем в тело метода main() (между его фигурными скобками) следующие строки:

```

Cat ourcat = new Cat();
ourcat.eat();
ourcat.sleep();
String say = ourcat.speak("Play with me");
System.out.println(say);

```

Разберем подробнее, что это значит.

Прежде чем вызывать, созданные нами методы в классе Cat, и заставить нашего кота есть, спать и говорить, сперва нужно создать экземпляр класса(инстанцию).

```
Cat ourcat = new Cat();
```

Данная строчка нам говорит о том, что в памяти создан экземпляр объекта Cat, а переменная ourcat типа Cat (такого же, как и наш объект) указывает на то место в памяти, где был этот объект создан.

Переменную ourcat теперь можно использовать для вызова методов класса Cat, например:

```
ourcat.eat(); ourcat.sleep();
```

При вызове этих методов в программе Eclipse удобно пользоваться комбинацией клавиш

Ctrl + пробел, после введения имени переменной и точки (рис 5.4). Программа подскажет, какие можно использовать методы для данной переменной.

Если метод возвращает какое-либо значение, например, как наш метод `speak()` возвращает значение типа `String`, то его можно вызывать следующим образом:

□ объявить переменную такого же типа, что и возвращаемое значение (в нашем случае `String`)

□ присвоить ей вызванный метод, например:

```
String say = ourcat.speak("Play with me");
```

Вспомним, что при описании нашего метода он содержал параметры `speak(String words)`. Теперь, при вызове в качестве параметра выступила фраза "Play with me", метод `speak()` ее обработал и вернул "Play with me...mauu...". Именно это значение он присвоил переменной `say`.

Мы это можем проверить, выведя `say` на печать при помощи команды:

```
System.out.println(say);
```

Итак, наш класс `Hello` теперь выглядит следующим образом:

```
public class Hello {
    /** * @param args */
    public static void main(String[] args) {
        Cat ourcat= new Cat();
        ourcat.eat();
        ourcat.sleep();
        String say = ourcat.speak("Play with me");
        System.out.println(say); } }
```

Теперь сохраним и запустим нашу программу. При запуске Eclipse может предложить выбрать `Java Applet` или `Java Application`. Нужно выбрать `Java Application`.

В качестве результата внизу в консоле мы получаем следующие строки.

```
Eating...Sleeping zz-z-z-z...Play with me...mauu...
```

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучите теоретические сведения.

Задание 1.

Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса. Пример:

Дан массив `[1, 2, 3]`, и массив весов `[1, 2, 10]`. В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».

Задание 2. По вариантам.

1. Создать класс Dog, в котором описать характеристики и поведение собаки. Вызвать его.
2. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.
3. Создать класс Car в котором описать основные характеристики и действия машины. Вызвать его.
4. Создать класс Tree в котором описать основные характеристики дерева. Вызвать его.
5. Создать класс Person в котором описать основные характеристики и действия человека. Вызвать его.
6. Создать класс Student в котором описать основные характеристики и действия студента. Вызвать его.
7. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.
8. Создать класс Tree в котором описать основные характеристики дерева. Вызвать его.
9. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.
10. Создать класс Person в котором описать основные характеристики и действия человека. Вызвать его.
11. Создать класс Car в котором описать основные характеристики и действия машины. Вызвать его.
12. Создать класс Dog, в котором описать характеристики и поведение собаки. Вызвать его.
13. Создать класс Student в котором описать основные характеристики и действия студента. Вызвать его.
14. Создать класс Tree в котором описать основные характеристики дерева. Вызвать его.
15. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.
16. Создать класс Car в котором описать основные характеристики и действия машины. Вызвать его.
17. Создать класс Person в котором описать основные характеристики и действия человека. Вызвать его.
18. Создать класс Tree в котором описать основные характеристики дерева. Вызвать его.
19. Создать класс Dog, в котором описать характеристики и поведение собаки. Вызвать его.
20. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.
21. Создать класс Student в котором описать основные характеристики и действия студента. Вызвать его.
22. Создать класс Person в котором описать основные характеристики и действия человека. Вызвать его.

23. Создать класс Car в котором описать основные характеристики и действия машины. Вызвать его.

24. Создать класс Tree в котором описать основные характеристики дерева. Вызвать его.

25. Создать класс Dog, в котором описать характеристики и поведение собаки. Вызвать его.

26. Создать класс Hotel в котором описать основные характеристики отеля. Вызвать его.

27. Создать класс Student в котором описать основные характеристики и действия студента. Вызвать его

2. Выполненные задания сдайте на проверку в папку, указанную преподавателем.

3. ДОМАШНЕЕ ЗАДАНИЕ

стр. 136-155

4. ЛИТЕРАТУРА

Альфред В., Ахо Компиляторы. Принципы, технологии и инструментарий, Вильямс, 2015.

Преподаватель

А.Н.Воронцова

Рассмотрено на заседании цикловой комиссии
программного обеспечения информационных
технологий

Протокол № _____ от «___» _____ 2020

Председатель ЦК _____ В.Ю.Михалевич