

Оценочные материалы при формировании рабочих программ дисциплин (модулей)

Направление подготовки / специальность: Прикладная математика и информатика
Профиль / специализация: Математическое и информационное обеспечение экономической
Дисциплина: Java-программирование интернет приложений

Формируемые компетенции: ПК-1
ПК-3

1. Описание показателей, критериев и шкал оценивания компетенций.

Показатели и критерии оценивания компетенций

Объект оценки	Уровни сформированности компетенций	Критерий оценивания результатов обучения
Обучающийся	Низкий уровень Пороговый уровень Повышенный уровень Высокий уровень	Уровень результатов обучения не ниже порогового

Шкалы оценивания компетенций при сдаче экзамена или зачета с оценкой

Достигнутый уровень результата обучения	Характеристика уровня сформированности компетенций	Шкала оценивания Экзамен или зачет с оценкой
Низкий уровень	Обучающийся: -обнаружил пробелы в знаниях основного учебно-программного материала; -допустил принципиальные ошибки в выполнении заданий, предусмотренных программой; -не может продолжить обучение или приступить к профессиональной деятельности по окончании программы без дополнительных занятий по соответствующей дисциплине.	Неудовлетворительно
Пороговый уровень	Обучающийся: -обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебной и предстоящей профессиональной деятельности; -справляется с выполнением заданий, предусмотренных программой; -знаком с основной литературой, рекомендованной рабочей программой дисциплины; -допустил неточности в ответе на вопросы и при выполнении заданий по учебно-программному материалу, но обладает необходимыми знаниями для их устранения под руководством преподавателя.	Удовлетворительно
Повышенный уровень	Обучающийся: - обнаружил полное знание учебно-программного материала; -успешно выполнил задания, предусмотренные программой; -усвоил основную литературу, рекомендованную рабочей программой дисциплины; -показал систематический характер знаний учебно-программного материала; -способен к самостоятельному пополнению знаний по учебно-программному материалу и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности	Хорошо

Высокий уровень	Обучающийся: -обнаружил всесторонние, систематические и глубокие знания учебно-программного материала; -умеет свободно выполнять задания, предусмотренные программой; -ознакомился с дополнительной литературой; -усвоил взаимосвязь основных понятий дисциплин и их значение для приобретения профессии; -проявил творческие способности в понимании учебно- программногo материала.	Отлично
-----------------	--	---------

Описание шкал оценивания

Компетенции обучающегося оценивается следующим образом:

Планируемый уровень результатов освоения	Содержание шкалы оценивания достигнутого уровня результата обучения			
	Неудовлетворительно Не зачтено	Удовлетворительно Зачтено	Хорошо Зачтено	Отлично Зачтено
Знать	Неспособность обучающегося самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения.	Обучающийся способен самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения.	Обучающийся демонстрирует способность к самостоятельному применению знаний при решении заданий, аналогичных тем, которые представлял преподаватель, и при его консультативной поддержке в части современных проблем.	Обучающийся демонстрирует способность к самостоятельному применению знаний в выборе способа решения неизвестных или нестандартных заданий и при консультативной поддержке в части междисциплинарных связей.
Уметь	Отсутствие у обучающегося самостоятельности в применении умений по использованию методов освоения учебной дисциплины.	Обучающийся демонстрирует самостоятельность в применении умений решения учебных заданий в полном соответствии с образцом, данным преподавателем.	Обучающийся продемонстрирует самостоятельное применение умений решения заданий, аналогичных тем, которые представлял преподаватель, и при его консультативной поддержке в части современных проблем.	Обучающийся демонстрирует самостоятельное применение умений решения неизвестных или нестандартных заданий и при консультативной поддержке преподавателя в части междисциплинарных связей.
Владеть	Неспособность самостоятельно проявить навык решения поставленной задачи по стандартному образцу повторно.	Обучающийся демонстрирует самостоятельность в применении навыка по заданиям, решение которых было показано преподавателем	Обучающийся демонстрирует самостоятельное применение навыка решения заданий, аналогичных тем, которые представлял преподаватель, и при его консультативной поддержке в части современных проблем.	Обучающийся демонстрирует самостоятельное применение навыка решения неизвестных или нестандартных заданий и при консультативной поддержке преподавателя в части междисциплинарных связей

2. Перечень вопросов и задач к экзаменам, зачетам, курсовому проектированию, лабораторным занятиям. Образец экзаменационного билета.

Примерный перечень вопросов к экзамену.

Компетенция ПК1

1. Язык Java. Основные операторы и методы.
2. Установка пакета разработчика Java.
3. Работа с графикой.
4. звуки и операции с мышью.
5. HTTP протокол
6. CGI интерфейсы
7. Использование параметров HTML.
8. Использование функций random и translate
9. Работа с изображениями.

Компетенция ПК3:

10. Использование мыши по принципу drag & drop
11. Java servlets API.
12. Структура Java servlets API.
13. JSP. Архитектура JSP.
14. Синтаксис JSP.
15. Шаблон MVC и его использование.
16. Пакет Struts.

Образец экзаменационного билета

Дальневосточный государственный университет путей сообщения		
Кафедра (к902) Высшая математика 3 семестр, учебный год	Экзаменационный билет № по дисциплине Java-программирование интернет приложений для направления подготовки / специальности 01.04.02 Прикладная математика и информатика профиль/специализация Математическое и информационное обеспечение экономической деятельности	«Утверждаю» Зав. кафедрой Виноградова П.В., д-р физ.- мат. наук, доцент «__» _____ 20__ г.
1. Работа с графикой. звуки и операции с мышью. HTTP протокол (ПК1, ПК3)		
2. Шаблон MVC и его использование. Пакет Struts. (ПК1, ПК3)		

Примечание. В каждом экзаменационном билете должны присутствовать вопросы, способствующих формированию у обучающегося всех компетенций по данной дисциплине.

3. Тестовые задания. Оценка по результатам тестирования.

Примерные задания теста

Задание 1 (ПК1, ПК3)

1.Скомпилируется ли данный код и если да — то каким будет вывод.....

```
public class Main{  
  
    public static void main(String[] args) {  
  
        System.out.println(new Main().$_$_()[2]);  
  
    }  
  
    ;short $_$_()[] {{{  
  
        return new short[007];  
  
    }}};  
  
};
```

Правильный вариант ответа: 0

2. Выберите верный вариант ответа. Вы пишете код для класса, который будет в пакете по умолчанию, и будете использовать сервисные классы в пакете java.util.

Выберите самый разумный фрагмент кода, для начала файла исходного текста.

(+) import java.util.* ;

(-) package default ; import java.util.* ;

(-) import java.util.* ; package default ;

(-) import java.lang.* ; import java.util.* ;

3. Выберите верный вариант ответа. Что случится, когда Вы попытаете откомпилировать и выполнить класс, содержащий этот метод?

```
public void testX() {  
  
    Integer nA = new Integer( 4096 ) ;  
  
    Long nB = new Long( 4096 ) ;  
  
    if( nA.equals( nB )) System.out.println("equals") ;  
  
    if( nA.intValue() == nB.longValue() ){  
  
        System.out.println("EQ") ;  
  
    }  
  
}
```

Варианты ответа:

(-)Компилятор выдаст ошибку в строке 4 потому что типы объекта nA и nB не соответствуют.

(+)Программа откомпилируется и выполнится, выведя " EQ ".

(-) Программа откомпилируется и выполнится, выведя " equalsEQ ".

(-)Программа откомпилируется и выполнится, выведя " equals ".

Полный комплект тестовых заданий в корпоративной тестовой оболочке АСТ размещен на сервере УИТ ДВГУПС, а также на сайте Университета в разделе СДО ДВГУПС (образовательная среда в личном кабинете преподавателя).

Соответствие между балльной системой и системой оценивания по результатам тестирования устанавливается посредством следующей таблицы:

Объект оценки	Показатели оценивания результатов обучения	Оценка	Уровень результатов обучения
Обучающийся	60 баллов и менее	«Неудовлетворительно»	Низкий уровень
	74 – 61 баллов	«Удовлетворительно»	Пороговый уровень
	84 – 75 баллов	«Хорошо»	Повышенный уровень
	100 – 85 баллов	«Отлично»	Высокий уровень

4. Оценка ответа обучающегося на вопросы, задачу (задание) экзаменационного билета, зачета, курсового проектирования.

Оценка ответа обучающегося на вопросы, задачу (задание) экзаменационного билета, зачета

Элементы оценивания	Содержание шкалы оценивания			
	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично
	Не зачтено	Зачтено	Зачтено	Зачтено
Соответствие ответов формулировкам вопросов (заданий)	Полное несоответствие по всем вопросам	Значительные погрешности	Незначительные погрешности	Полное соответствие
Структура, последовательность и логика ответа. Умение четко, понятно, грамотно и свободно излагать свои мысли	Полное несоответствие критерию.	Значительное несоответствие критерию	Незначительное несоответствие критерию	Соответствие критерию при ответе на все вопросы.
Знание нормативных, правовых документов и специальной литературы	Полное незнание нормативной и правовой базы и специальной литературы	Имеют место существенные упущения (незнание большей части из документов и специальной литературы по названию, содержанию и т.д.).	Имеют место несущественные упущения и незнание отдельных (единичных) работ из числа обязательной литературы.	Полное соответствие данному критерию ответов на все вопросы.
Умение увязывать теорию с практикой, в том числе в области профессиональной работы	Умение связать теорию с практикой работы не проявляется.	Умение связать вопросы теории и практики проявляется редко	Умение связать вопросы теории и практики в основном проявляется.	Полное соответствие данному критерию. Способность интегрировать знания и привлекать сведения из различных научных сфер
Качество ответов на дополнительные вопросы	На все дополнительные вопросы преподавателя даны неверные ответы.	Ответы на большую часть дополнительных вопросов преподавателя даны неверно.	1. Даны неполные ответы на дополнительные вопросы преподавателя. 2. Дан один неверный ответ на дополнительные вопросы преподавателя.	Даны верные ответы на все дополнительные вопросы преподавателя.

Примечание: итоговая оценка формируется как средняя арифметическая результатов элементов оценивания.

Задание на РГР.

Образец выполнения 1. «Создание игры на Java для двух игроков».

Задание. Создание игры на Java для двух игроков.

Общие требования:

- Подсчет очков и вывод их по завершению игры.
 - Ведение таблицы победителей Cookie. Отображение таблицы и просмотр результатов.
 - Подсчет времени игры.
 - Изменение различных настроек игры.
 - Изменение уровней сложности.
 - Все варианты игр не включают в себя написание интеллектуального программного кода.
- Если игра для двух игроков, то предполагается, что будут играть два человека, а не человек с компьютером.

Варианты заданий:

1. Тетрис.
2. Lines.
3. Пятнашки.
4. Крестики-нолики.
5. Червяк и яблоки. По игровому полю ползает червяк, который при съедании яблока увеличивается в размерах. Сложность игры определяется количеством препятствий и разными объектами поедания.
6. Ралли. Внизу игрового поля влево, вправо двигается машина, а на нее надвигается дорога с препятствиями.
7. Охота (на попадание мышкой по мишеням). В пределах игрового поля в разных местах появляется мишень, а игрок должен по ней попасть курсором. Сложность определяется скоростью появления и количеством мишеней.
8. Однорукий бандит.
9. Виселица. Должна быть введена база начальных слов. При неудачном угадывании строится виселица.
10. Galaxian. По экрану двигается группа объектов, они перемещаются влево, вправо и вниз. Внизу игрового поля двигается пушка, выстрелами которой нужно сбить все объекты. Сложность определяется скоростью перемещения и приближения объектов.
11. Поле чудес. Должна быть заполнена база вопросов и ответов.
12. Морской бой.
13. Балда. Необходима начальная база слов. При введении слова, которое отсутствует в базе, должен задаваться запрос о разрешении добавить его в базу.
14. Минер.левой кнопкой мыши открываем клетку. Правой - помечаем мину.
15. Тетрис с шариками. В стакан падают группы из трех шариков одного цвета.
16. Pacman. Герой двигается и поедает точки. Ему должны мешать препятствия и другие герои.
17. Уголки.
18. Пасьянс.
19. TOPPLER. Отправлять кружки с пивом по 2-8(сложность) стойкам, по которым приближаются жаждущие (скорость движения и количество нарастает).
20. FILLER. Начальное квадратное поле заполнено разноцветными квадратиками. Поочередно два(четыре) игрока выбирают цвет. Все примыкающие к соответствующему углу квадратики прошлого цвета становятся выбранного. Побеждает тот, кто захватил поле больше других.
21. Морской бой.
22. Rally.

Образец выполнения 2. «Разработка приложений на Java».

Задание: Разработайте своё приложение Java. Рекомендуется разработка Rich-UI приложения либо Web-приложения.

Реализовать программу, реализующую интерполирование исходных данных, отобразив их на графике, используя:

1. Интерполирование многочленом Лагранжа
2. Интерполирование многочленом Ньютона
3. Интерполирование кубическими сплайнами
4. Аппроксимация методом наименьших квадратов

Варианты исходных данных.

	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
	0.0	0.01	0.2	1.2	1.4	2.0	2.5	3.0	3.2	3.21
	0	2	4	6	8	10	12	14	16	18
	12.5	8.9	7.5	5.3	3.8	2.3	1.5	1.2	1.05	1.0
	0	2	4	6	8	10	12	14	16	18
	0.0	1.2	6.7	12.6	9.8	5.8	3.6	1.5	1.1	0.7
	0.55	0.8	0.93	1.16	1.31	1.48	1.78	2.06	2.58	2.89
	0.44	0.8	1.63	1.26	1.07	1.11	0.83	0.37	0.21	0.15
	1.0	1.6	2.2	2.8	3.4	4.0	4.6	5.2	5.8	6.4
	145.67	136.53	128.34	90.56	30.76	15.3	12.76	11.89	11.15	10.89
	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
	1.0	0.65	0.2	0.05	0.02	0.05	0.02	0.65	0.9	1.0
	1.0	1.6	2.2	2.8	3.4	4.0	4.6	5.2	5.8	6.4
	12.8	13.1	13.67	14.2	30.8	50.76	62.45	65.43	66.1	66.67
	1.0	1.8	2.6	3.4	4.2	5.0	5.8	6.6	7.4	8.2
	31.5	5.3	2.1	1.12	0.07	-0.1	-2.5	-7.8	-12.5	-27.2
	-2.8	-2.7	-2.0	-1.0	0.0	0.5	1.0	1.5	2.0	2.5
	-11.4	-5.4	-1.2	-0.2	0.0	0.25	2.0	6.5	12.8	23.6
	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5

		0.0	0.35	0.8	0.95	0.98	0.95	0.8	0.35	0.1	0.0
		0.0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7
		-1.0	0.1	1.3	1.7	0.3	0.7	2.5	3.0	2.9	2.7
		-5.0	-4.0	-3.0	-2.0	-1.0	0.0	1.0	2.0	3.0	4.0
		12.5	7.8	2.3	0.4	-4.1	0.2	1.9	4.8	9.4	10.5
		-3.0	-2.5	-2.0	-1.5	-1.0	-0.5	0.0	0.5	1.0	1.5
		4.5	4.6	4.4	3.5	1.3	0.2	0.1	0.3	0.2	0.2
		-1.5	-1.25	-1.0	-0.75	-0.5	-0.25	0.0	0.25	0.5	0.75
		0.1	0.12	0.7	1.4	2.9	1.6	0.6	0.2	0.1	0.05
		0	3	6	9	12	15	18	21	24	27
		0.0	2.7	9.3	10.4	10.6	10.7	10.6	9.8	9.1	8.7
		0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
		0.01	0.03	0.28	1.9	1.5	1.3	1.1	1.6	2.2	2.28
		0	2	4	6	8	10	12	14	16	18
		1.5	1.9	2.5	5.7	4.8	3.3	2.5	1.6	1.0	0.8
		0.0	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
		0.0	1.3	6.9	11.6	8.7	6.2	4.1	1.6	1.3	0.5
		0.5	0.8	1.1	1.4	1.7	2.0	2.3	2.6	2.9	3.1
		0.4	0.85	1.67	1.16	1.67	1.21	0.74	0.28	0.19	0.13
		1.0	1.6	2.2	2.8	3.4	4.0	4.6	5.2	5.8	6.4
		45.6	36.5	28.3	9.5	3.7	5.3	2.7	1.8	1.1	1.8

Тематика контрольных работ

1. Тема контрольной работы №1 «Базовые операции приведения типов, чтение из стандартного входного и записи в стандартный выходной потоки»
2. Тема контрольной работы №2 «Работа с классами и объектами Java»

Образцы контрольных работ
Контрольная работа № 1

Задание: Определить расстояние на плоскости между двумя точками с заданными координатами $M1(x1,y1)$ и $M2(x2,y2)$

Этапы решения задачи

1. Математическая модель: расстояние на плоскости между двумя точками $M1(x1,y1)$ и $M2(x2,y2)$ высчитывается по формуле
2. Составим схему алгоритма
3. Создадим новый Java-файл в директории labs с именем Lab01.java

```
package labs;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import static java.lang.Double.parseDouble;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;
import static java.lang.System.in;
import static java.lang.System.out;

/**
 * @author: Ilya Sadykov (mailto: smeccia@yandex-team.ru)
 */
public class Lab01 {
```

```

public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

    out.println("Эта программа вычисляет расстояние между двумя
точками на плоскости");

    out.println("Введите координаты двух точек:");
    out.print("X1=");
    double x1 = parseDouble(reader.readLine());
    out.print("Y1=");
    double y1 = parseDouble(reader.readLine());
    out.print("X2=");
    double x2 = parseDouble(reader.readLine());
    out.print("Y2=");
    double y2 = parseDouble(reader.readLine());
    double distance = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
    out.println("d=" + distance);
    out.println("Нажмите Enter для завершения работы программы");
    reader.readLine();
}
}

```

Откомпилируем файл с помощью команды:

```
javac labs/Lab01.java
```

4. Запустим файл с помощью команды:

```
java labs.Lab01
```

Варианты заданий:

1. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.

2. Определить время падения камня на поверхность земли с высоты h .
3. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
4. Треугольник задан координатами своих вершин. Найти: периметр треугольника; площадь треугольника.
5. Вычислить высоту треугольника, опущенную на сторону a , по известным значениям длин его сторон a, b, c .
6. Вычислить объем цилиндра с радиусом основания r и высотой h .
7. Определить координаты вершины параболы $y=ax^2+bx+c$ ($a \neq 0$). Коэффициенты a, b, c заданы.
8. По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали.

Контрольная работа № 2

Задание: Реализовать иерархию классов, необходимую для отрисовки различных типов фигур: Треугольник, Окружность, Квадрат. При реализации учесть схожие уникальные особенности каждой фигуры. Схожие черты вынести в интерфейсы и абстрактные классы, уникальные реализовать непосредственно в классе каждой фигуры. Разработать класс-отрисовщик фигур (вывод параметров на консоль).

Этапы решения задачи

1. Реализуем следующую иерархию классов
2. Создадим набор классов в пакете lab02:

```
public interface Figure {  
    public void draw();  
}  
  
public class Triangle implements Figure {  
    @Override  
    public void draw() {  
    }  
}
```

```
public class Square implements Figure {  
    @Override  
    public void draw() {  
    }  
}
```

```
public class Dot implements Figure {  
    @Override  
    public void draw() {  
    }  
}
```

```
public class Rectangle extends Square {  
}
```

```
public class Circle extends Dot {  
}
```

```
public class Ellipse extends Circle {  
}
```

3. Добавим поля, характеризующие каждый тип. Например, для точки:

```
public class Dot implements Figure {  
    private int centerX;  
    private int centerY;  
  
    public Dot(int centerX, int centerY) {
```

```

        this.centerX = centerX;

        this.centerY = centerY;
    }

    public int getCenterX() {
        return centerX;
    }

    public void setCenterX(int centerX) {
        this.centerX = centerX;
    }

    public int getCenterY() {
        return centerY;
    }

    public void setCenterY(int centerY) {
        this.centerY = centerY;
    }

    @Override
    public void draw() {
    }
}

```

4. Также добавим реализацию отрисовки путём вывода на консоль для каждого типа. Например, для точки:

```

@Override
    public void draw() {

```

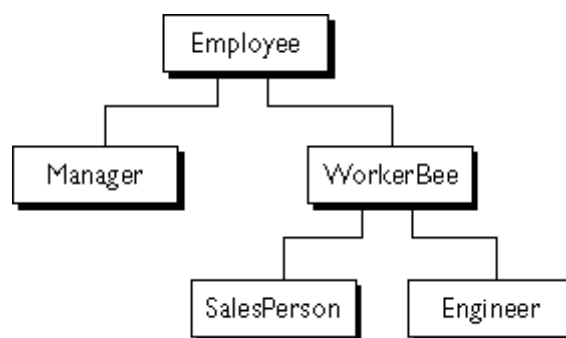
```
        System.out.println("Dot[" + centerX + ";" + centerY + "]);  
    }  
}
```

Реализуем Main-класс и отрисует различные типы фигур:

```
public class Drawer {  
  
    public static void main(String[] drawer) {  
  
        Figure[] figures = new Figure[]{  
            new Dot(0, 0),  
            new Ellipse(0, 0),  
            new Circle(10, 20),  
            new Square(),  
            new Rectangle()  
        };  
  
        for(Figure figure : figures){  
            figure.draw();  
        }  
    }  
}
```

Варианты заданий:

1. Реализуйте следующую иерархию классов:



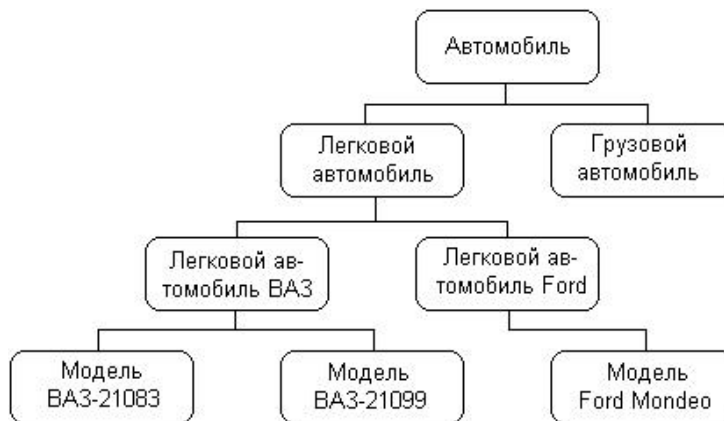
- ✓ Employee содержит свойства name (значением которого по умолчанию является пустая строка) и dept (значением которого по умолчанию является "general").

- ✓ Manager базируется на Employee. Он добавляет свойство reports (значением которого по умолчанию является пустой массив, который предполагается использовать для размещения в качестве значений массива Employee-объектов).
- ✓ WorkerBee также базируется на Employee. Он добавляет свойство projects (значением которого по умолчанию является пустой массив, который предполагается использовать для размещения в качестве значений массива строк).
- ✓ SalesPerson базируется на WorkerBee. Он добавляет свойство quota (значением которого по умолчанию является 100). Он также переопределяет свойство dept значением "sales", указывая, что все salespersons/продавцы находятся в том же department/департаменте.
- ✓ Engineer базируется на WorkerBee. Он добавляет свойство machine (значением которого по умолчанию является пустая строка), а также переопределяет свойство dept значением "engineering".

2. Классы собак, кошек и утконосов являются дочерними по отношению к классу млекопитающих, следовательно наследуют его свойства. При программной реализации этой иерархии реализуйте метод "кормление детенышей" в родительском классе, вместо того, чтобы несколько раз дублировать его в каждом из подклассов. Внутри метода сделайте вывод на консоль сообщение о запуске процесса кормления. В классе млекопитающих реализуйте метод "потомство" (родить детеныша), в классах собак и кошек этот метод будет отсутствовать (система будет искать его в родительском классе и найдет его там), а в классе утконосов нужно написать новый метод, с тем же именем, но другой реализацией (отложить яйца).



3. Классы «Легковой автомобиль» и «Грузовой автомобиль» являются подклассами класса «Автомобиль», а классы «Модель ВАЗ-21083» и «Модель ВАЗ-21099» являются подклассами класса «Легковой автомобиль производства ВАЗ». Каждая конкретная модель автомобиля имеет определённую стоимость. Реализуйте иерархию и Main-класс, в котором выведете на консоль все имеющиеся в магазине автомобили (предварительно их необходимо добавить в массив).



4. Гусь и пингвин имеют различную скорость передвижения. Реализуйте классы этих птиц и контроллер, который будет управлять их передвижением. В контроллере реализуйте набор различных птиц и их перемещение в цикле с выводом результата пройденного расстояния на консоль.

