**Лабораторная работа №16**

**Основы программирования на Java: проектирование иерархии классов. Наследование и полиморфизм**

**Цель работы:** Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки проектирования иерархии классов в Java, следуя принципам наследования и полиморфизма.

**Теоретические сведения**

**Рекомендации при проектировании иерархии**

При построении иерархии необходимо помнить, что отношение между классами можно выразить как «is-a», или «является». Студент «является» Человеком. Поля класса находятся с классом в отношении «has-a», или «содержит». Студент «содержит» Номер зачетной книжки. Наследование и переопределение методов используются для реализации отличий поведения. Если наследование можно заменить агрегацией, то следует так и поступить.

Нет смысла создавать подкласс Студент-заочник, если можно в подкласс Студент добавить поле Форма обучения. При наследовании в новые классы добавляются новые возможности в виде полей и методов или переопределения методов. Если новых возможностей не обнаруживается, то использование наследования, как правило, не имеет для этого оснований.

Базовая функциональность должна определяться в вершине иерархии проектируемых классов. Если в подклассе добавляются новые методы, характеризующие поведение иерархии в целом, следует заняться перепроектированием. Но нельзя учесть все возможные изменения иерархии в процессе разработки.

Избыточный функционал придется поддерживать. Лучше на поздних стадиях добавить методы в суперкласс, чем пытаться понять, зачем нужен метод, которой никто еще не использовал.

Не использовать значения переменных для характерного изменения поведения. Для этих целей следует создать подкласс и переопределить метод.

Для различных семантических сущностей не создавать общую иерархию, даже если действия для них идентичны по форме. Блокировка Теста и блокировка Студента — действия суть похожие, но отличные по своему функционалу и последствиям.

**Задание**

Изучить теоретический материал и выполнить задачи по вариантам.

**Задача**

Создать приложение, удовлетворяющее требованиям, приведенным в задании. Наследование применять только в тех заданиях, в которых это логически обосновано. Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы equals(), hashCode(), toString().

1. Создать объект класса Текст, используя классы Предложение, Слово. Методы: дополнить текст, вывести на консоль текст, заголовок текста.

2. Создать объект класса Автомобиль, используя классы Колесо, Двигатель. Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.

3. Создать объект класса Самолет, используя классы Крыло, Шасси, Двигатель. Методы: летать, задавать маршрут, вывести на консоль маршрут

4. Создать объект класса Государство, используя классы Область, Район, Город. Методы: вывести на консоль столицу, количество областей, площадь, областные центры.

5. Создать объект класса Планета, используя классы Материк, Океан, Остров. Методы: вывести на консоль название материка, планеты, количество материков.

6. Создать объект класса Звездная система, используя классы Планета, Звезда, Луна. Методы: вывести на консоль количество планет в звездной системе, название звезды, добавление планеты в систему.

7. Создать объект класса Компьютер, используя классы Винчестер, Дисковод, Оперативная память, Процессор. Методы: включить, выключить, проверить на вирусы, вывести на консоль размер винчестера.

8. Создать объект класса Квадрат, используя классы Точка, Отрезок. Методы: задание размеров, растяжение, сжатие, поворот, изменение цвета.

9. Создать объект класса Круг, используя классы Точка, Окружность. Методы: задание размеров, изменение радиуса, определение принадлежности точки данному кругу.

10. Создать объект класса Щенок, используя классы Животное, Собака. Методы: вывести на консоль имя, подать голос, прыгать, бегать, кусать.

11. Создать объект класса Наседка, используя классы Птица, Кукушка. Методы: летать, петь, нести яйца, высиживать птенцов.

12. Создать объект класса Текстовый файл, используя классы Файл, Директория. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.

13. Создать объект класса Одномерный массив, используя классы Массив, Элемент. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить).

14. Создать объект класса Простая дробь, используя класс Число. Методы: вывод на экран, сложение, вычитание, умножение, деление.

15. Создать объект класса Дом, используя классы Окно, Дверь. Методы: закрыть на ключ, вывести на консоль количество окон, дверей.

16. Создать объект класса Цветок, используя классы Лепесток, Бутон. Методы: расцвести, завять, вывести на консоль цвет бутона.

17. Создать объект класса Дерево, используя классы Лист, Ветка. Методы: зацвести, опасть листьям, покрыться инеем, пожелтеть листьям.

18. Создать объект класса Пианино, используя классы Клавиша, Педаль. Методы: настроить, играть на пианино, нажимать клавишу.

19. Создать объект класса Фотоальбом, используя классы Фотография, Страница. Методы: задать название фотографии, дополнить фотоальбом фотографией, вывести на консоль количество фотографий.

20. Создать объект класса Год, используя классы Месяц, День. Методы: задать дату, вывести на консоль день недели по заданной дате, рассчитать количество дней, месяцев в заданном временном промежутке.

21. Создать объект класса Сутки, используя классы Час, Минута. Методы: вывести на консоль текущее время, рассчитать время суток (утро, день, вечер, ночь).

22. Создать объект класса Птица, используя классы Крылья, Клюв. Методы: летать, садиться, питаться, атаковать.

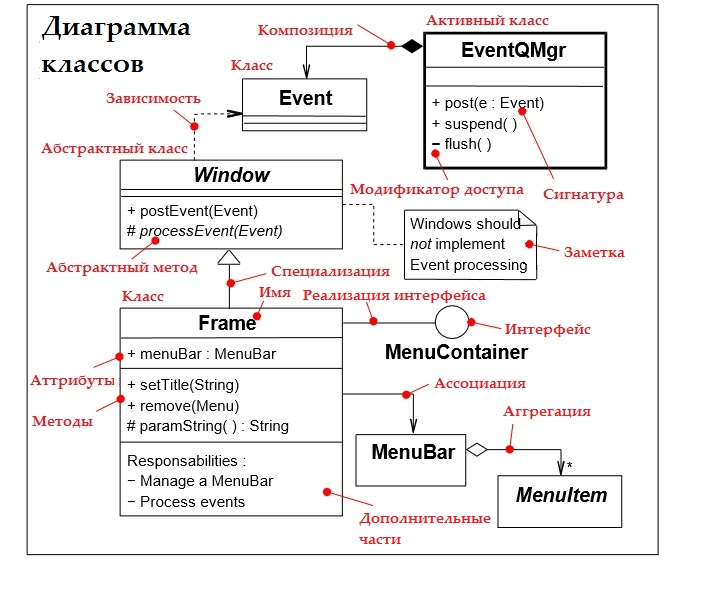
23. Создать объект класса Хищник, используя классы Когти, Зубы. Методы: рычать, бежать, спать, добывать пищу.

24. Создать объект класса Гитара, используя класс Струна, Скворечник. Методы: играть, настраивать, заменять струну.

**Форма отчета**

1. Тема, цель, индивидуальное задание

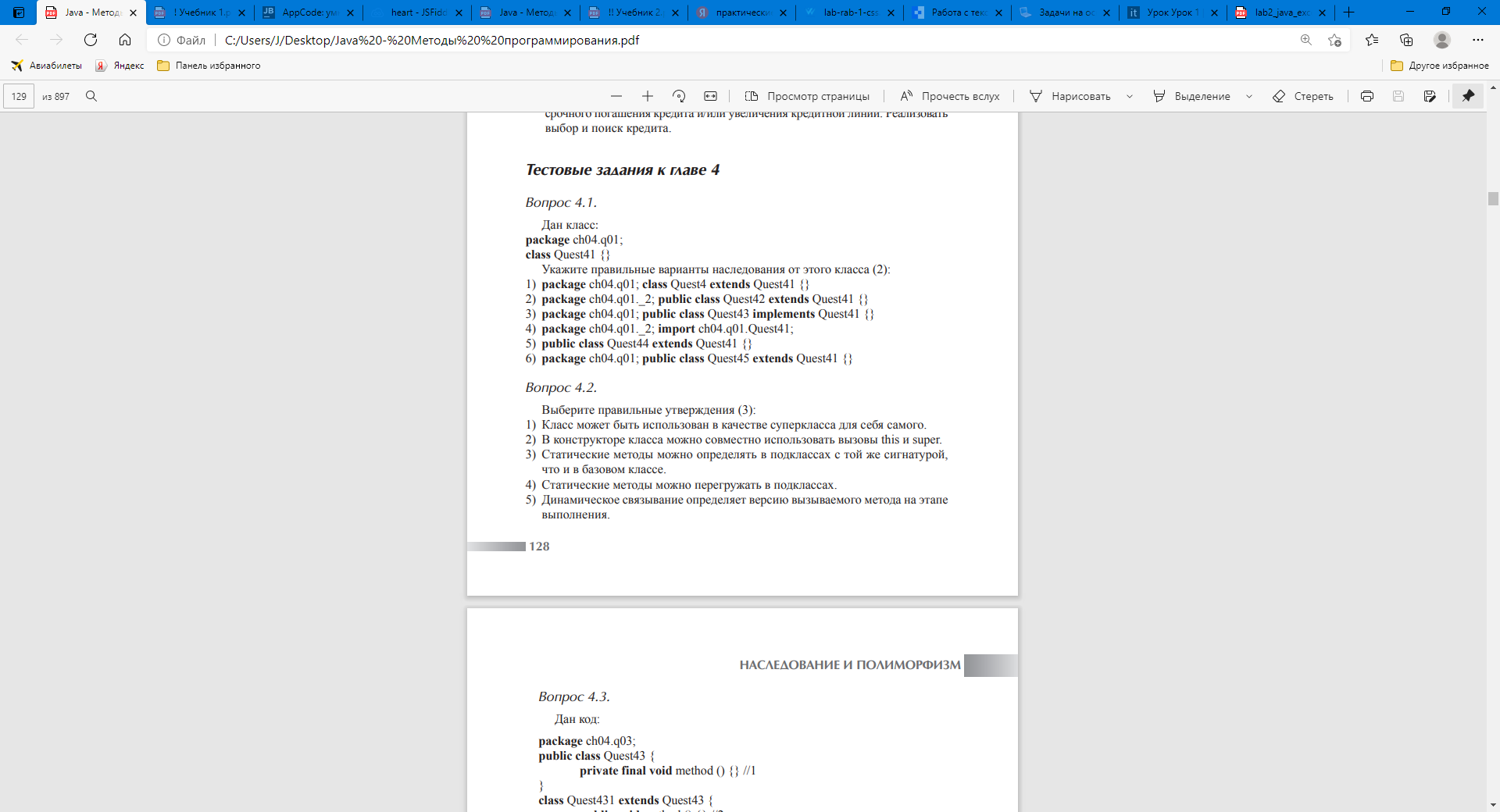
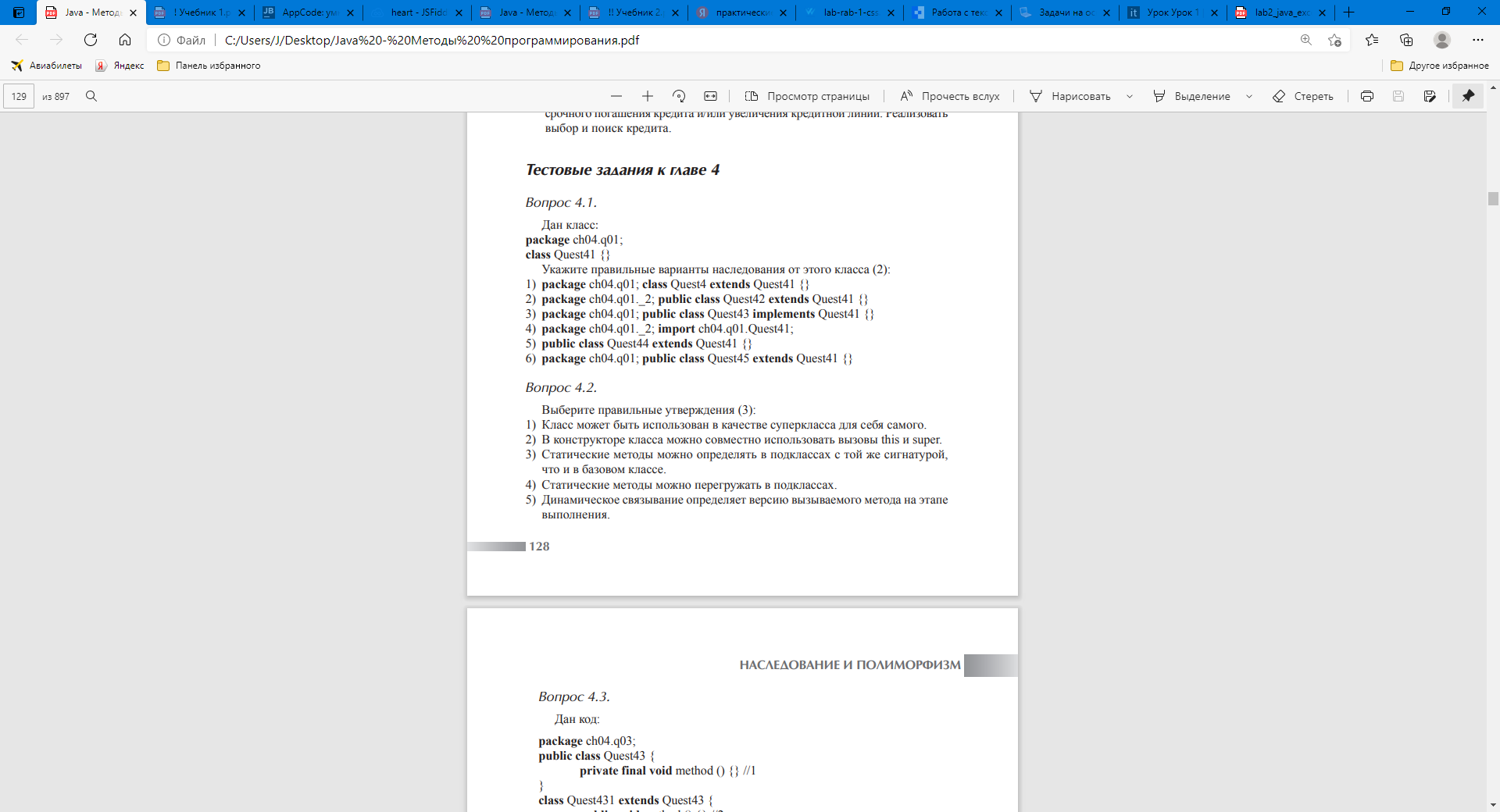
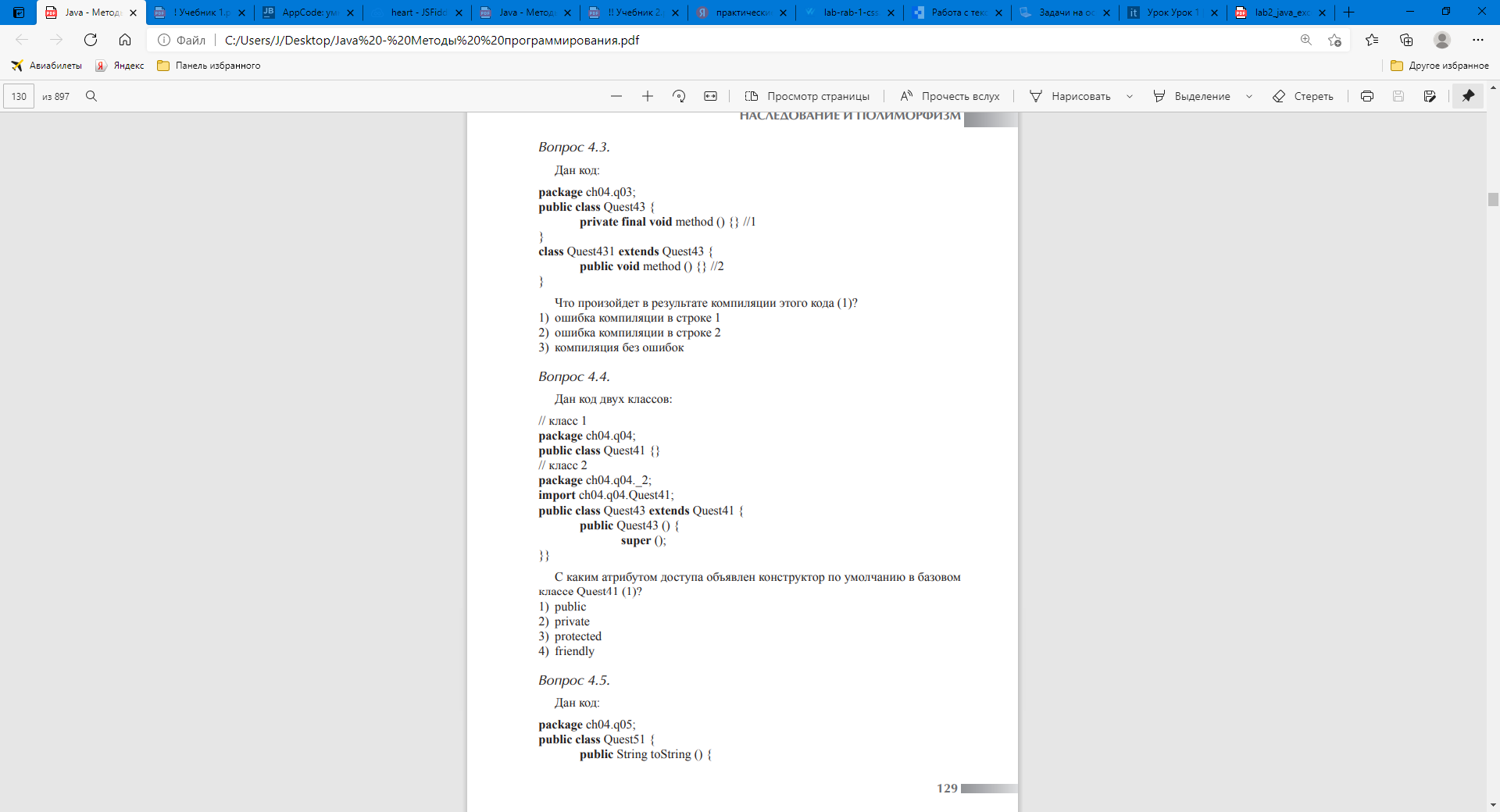
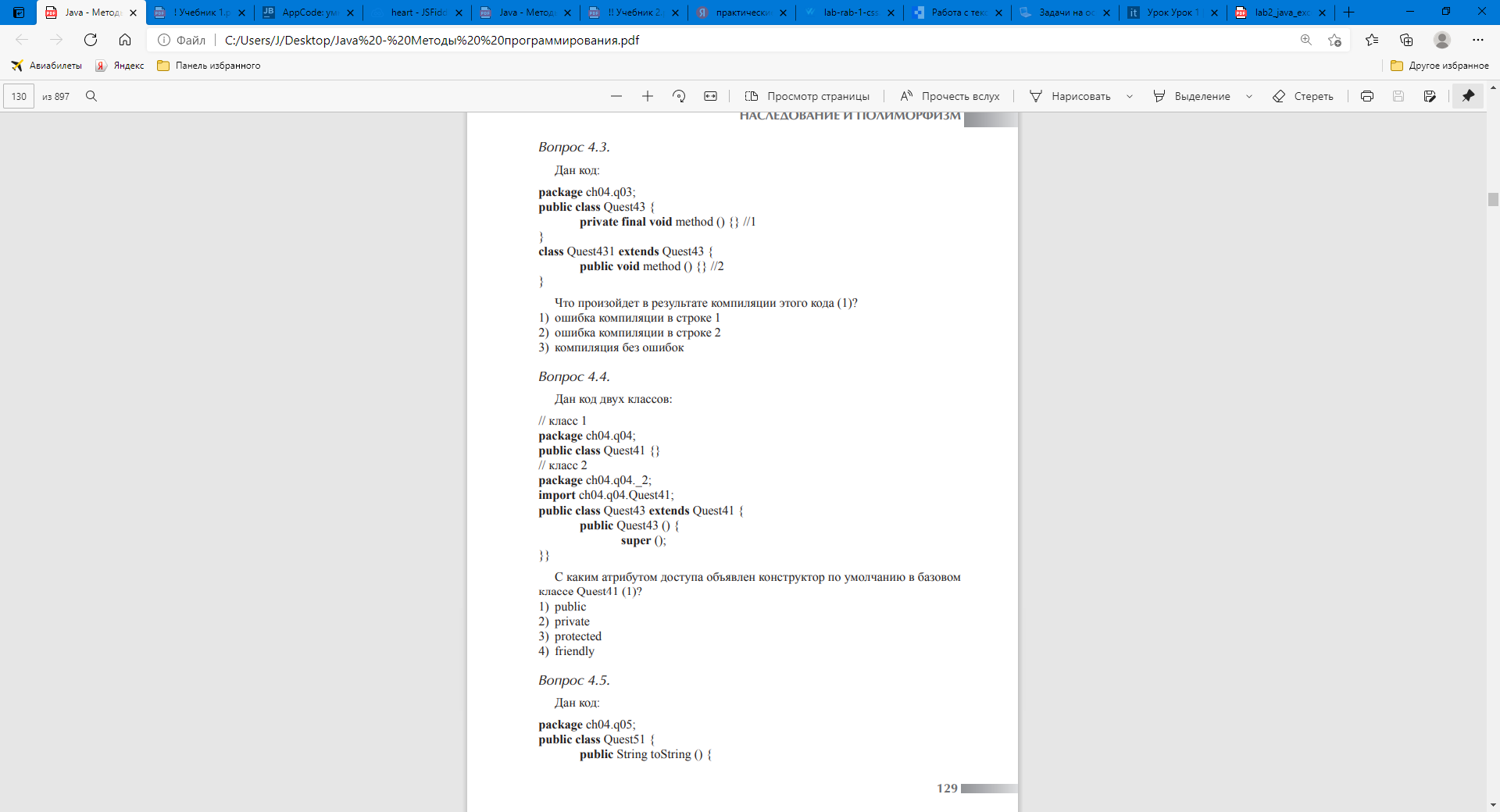
2. Код программы, ИЕРАРХИЯ классов

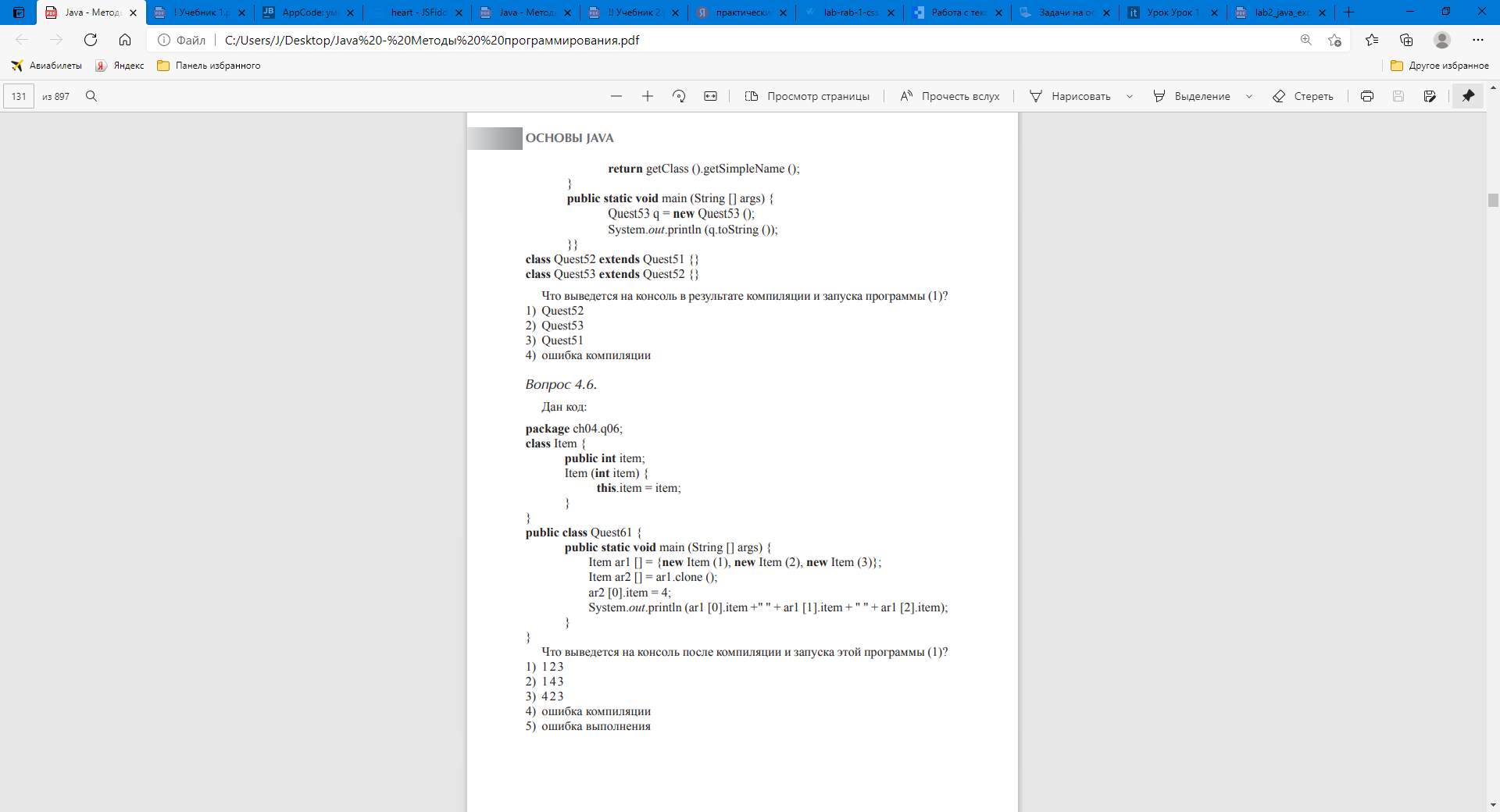
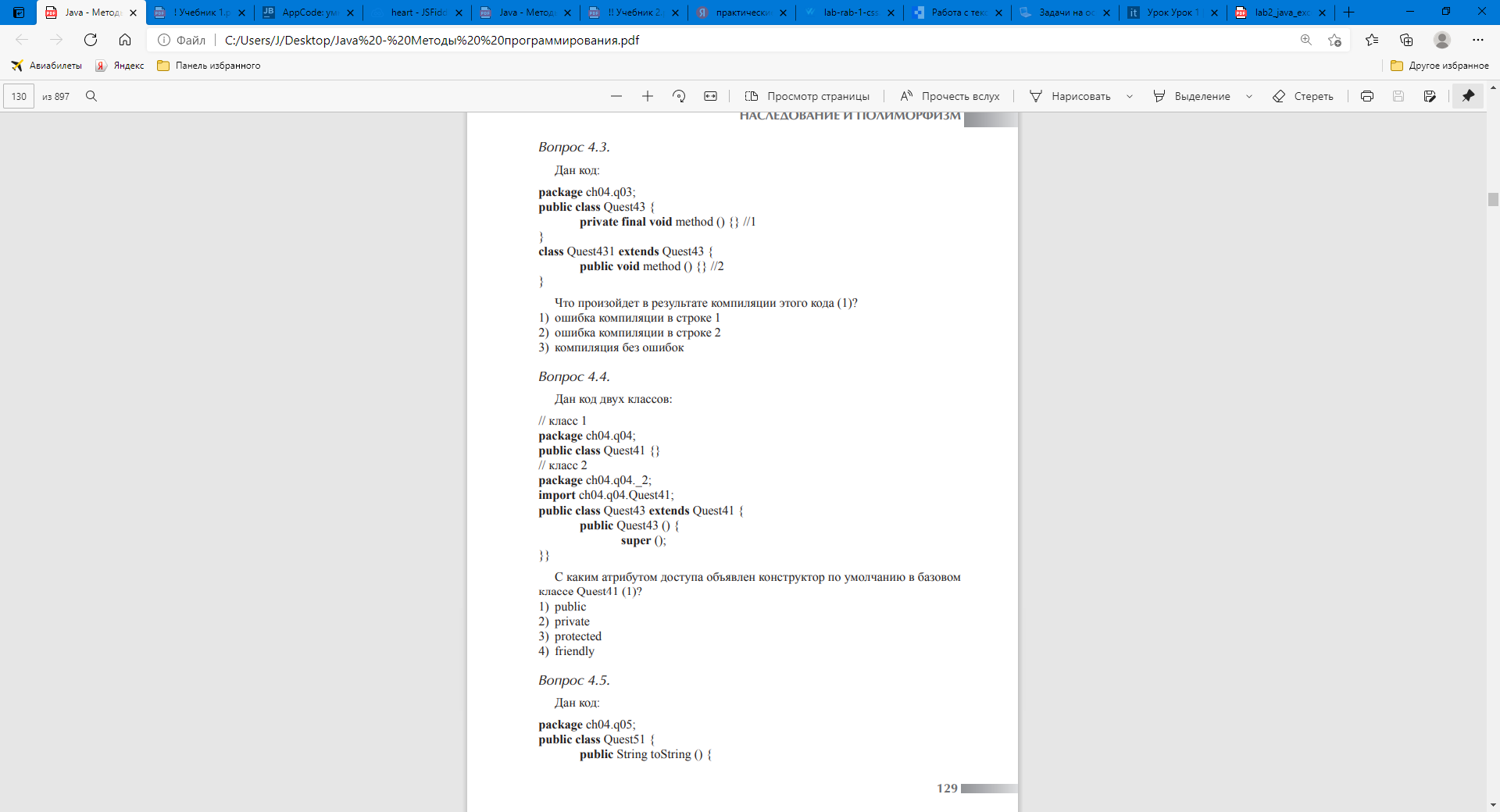


3. Скриншот результата

4. Ответы на контрольные вопросы

**Контрольные вопросы для защиты**

1. Что такое класс? Объект?
2. Что такое наследование? Полиморфизм?
3. 
4. 
5. 
6. 

7.

8. 