БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем и технологий»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине «Программирование на языке Java»

Выполнила:

студентка группы 10702121

Алюнина П.А.

Проверил:

Сидорик В.В.

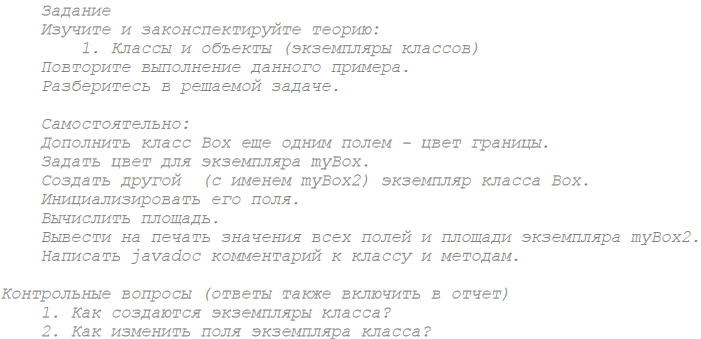
Минск, 2023

Цель: изучить синтаксис создания объектов и классов, работу с объектами, написание конструкторов.

Задачи лабораторной работы:

1. Изучить и законспектировать теорию: классы, объекты (экземпляры классов) и конструкторы.
2. Повторить выполнения примеров заданий для закрепления знаний по написанию классов и конструкторов.
3. Выполнить задания, подставив индивидуальные данные, или изменив функциональность методов.
4. Оформить отчёт с теорией и скриншотами для каждого индивидуального задания.
5. Составить глоссарий по необходимым теориям.
6. Ответить на контрольные вопросы, заданные в каждом задании:
7. Как создаются экземпляры класса?
8. Как изменить поля экземпляра класса?
9. Что такое конструктор по умолчанию?
10. Какие модификаторы доступа рекомендуется использовать для полей, метолов?
11. Что такое инкапсуляция?

**Задание №40 BoxDemo**



Java является объектно-ориентированным языком, поэтому такие понятия как "класс" и "объект" играют в нем ключевую роль. Любую программу на Java можно представить как набор взаимодействующих между собой объектов.

Шаблоном или описанием объекта является класс, а объект представляет экземпляр этого класса. Можно еще провести следующую аналогию. У нас у всех есть некоторое представление о человеке - наличие двух рук, двух ног, головы, туловища и т.д. Есть некоторый шаблон - этот шаблон можно назвать классом. Реально же существующий человек (фактически экземпляр данного класса) является объектом этого класса.

Класс определяется с помощью ключевого слова сlass.

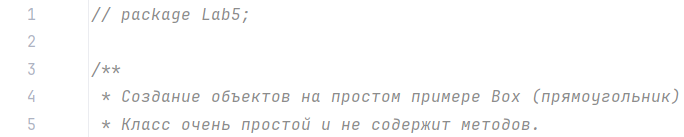
Любой объект может обладать двумя основными характеристиками: состояние - некоторые данные, которые хранит объект, и поведение - действия, которые может совершать объект.

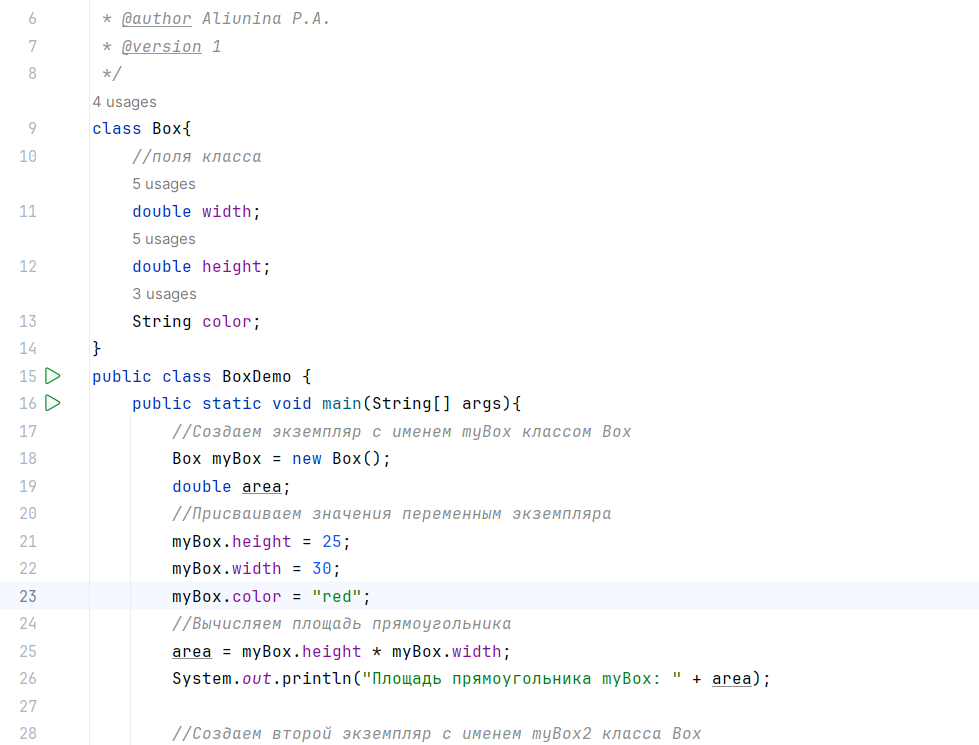
Для хранения состояния объекта в классе применяются поля или переменные класса. Для определения поведения объекта в классе применяются методы.

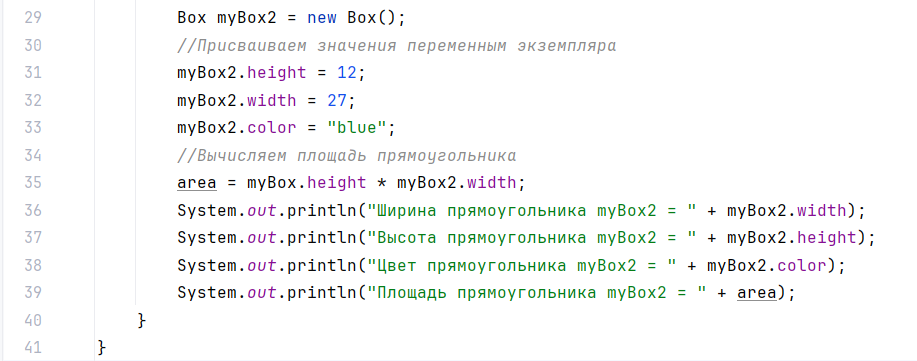
Новый объект (или экземпляр) создаётся из существующего класса при помощи ключевого слова new:

Person boy = new Person(); //создали мальчика из класса Person

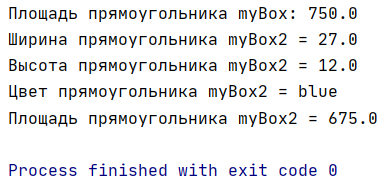
1. **Код программы**







1. **Скриншот выполнения программы**



1. **Ответы на контрольные вопросы**
2. Как создаются экземпляры класса?

Новый объект (или экземпляр) создаётся из существующего класса при помощи ключевого слова **new**:



Слева от оператора присваивания = определяется имя переменной и его тип **Cat**. В правой части выражения происходит выделение памяти для нового экземпляра класса **Cat** и инициализируется экземпляр. Оператор присваивания присваивает переменной ссылку на только что созданный объект. Имена объектов не нужно начинать с большой буквы, как у класса. Так вы будете различать, где класс, а где экземпляр класса.

Если имя экземпляра класса состоит из нескольких слов, то используется верблюжья нотация, когда все первые буквы слов, кроме первой, пишутся с большой (пр. – superBlackCat.)

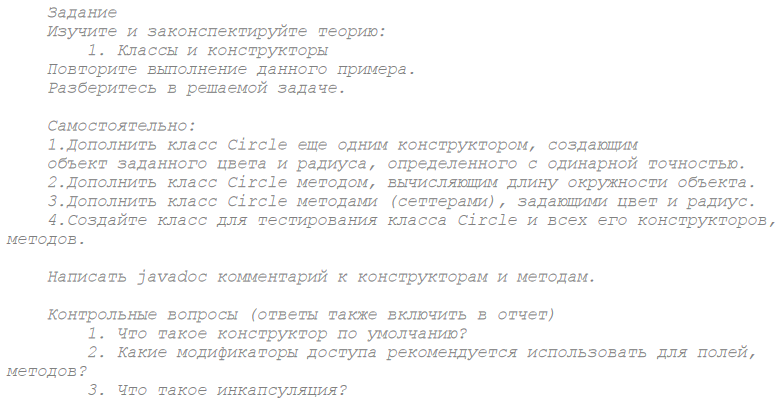
1. Как изменить поля экземпляра класса?

Во-первых, все переменные экземпляра (поля) необходимо помечать модификатором private.

Для изменения поля экземпляра класса необходимо использовать “сеттеры” и “геттеры”— методы для работы с полями класса.

Название происходит от английского “get” — “получать” (т.е. “метод для получения значения поля”) и set — “устанавливать” (т.е. “метод для установки значения поля”).

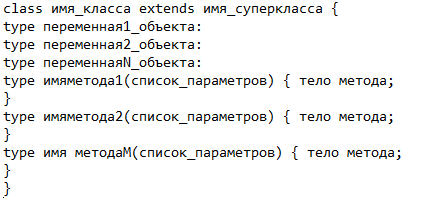
**Задание №41 TestCircle**



Базовым элементом объектно-ориентированного программирования в языке Java является класс.

Класс — это шаблон для создания объекта. Класс определяет структуру объекта и его методы, образующие функциональный интерфейс. В процессе выполнения Java-программы система использует определения классов для создания представителей классов. Представители являются реальными объектами. Термины «представитель», «экземпляр» и «объект» взаимозаменяемы.

Ниже приведена общая форма определения класса:



Ключевое слово extends указывает на то, что «*имя\_класса*» — это подкласс класса «*имя\_суперкласса*».

Конструктор — это специальный метод, который вызывается при создании нового объекта. Не всегда удобно инициализировать все переменные класса при создании его экземпляра. Иногда проще, чтобы какие-то значения были бы созданы по умолчанию при создании объекта. По сути, конструктор нужен для автоматической инициализации переменных.

Конструктор инициализирует объект непосредственно во время создания. Имя конструктора совпадает с именем класса, включая регистр, а по синтаксису конструктор похож на метод без возвращаемого значения.

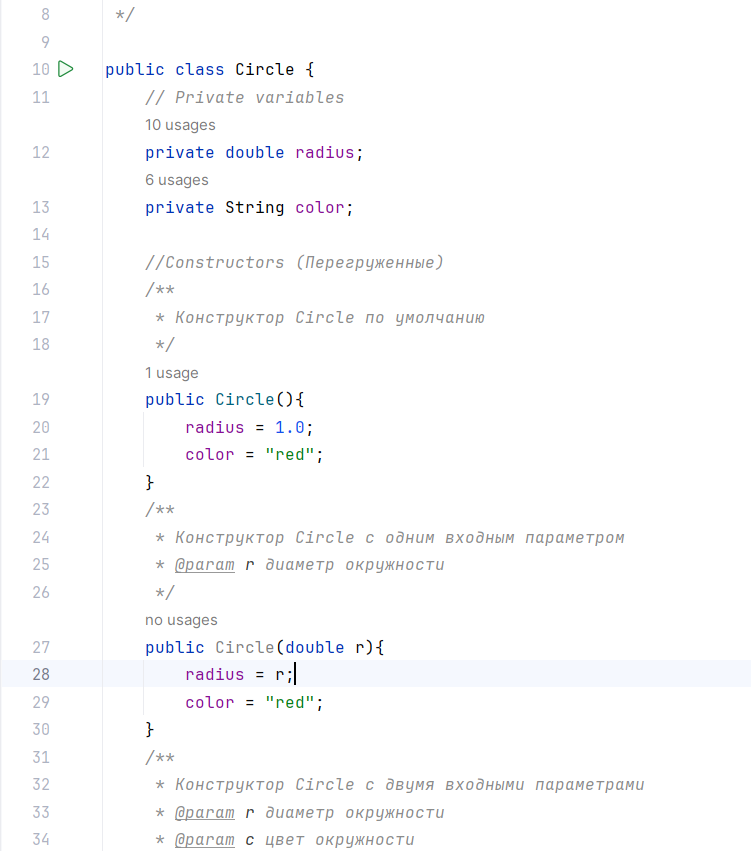
private int Cat(); *// так выглядит метод по имени Cat*Cat(); *// так выглядит конструктор класса Cat*

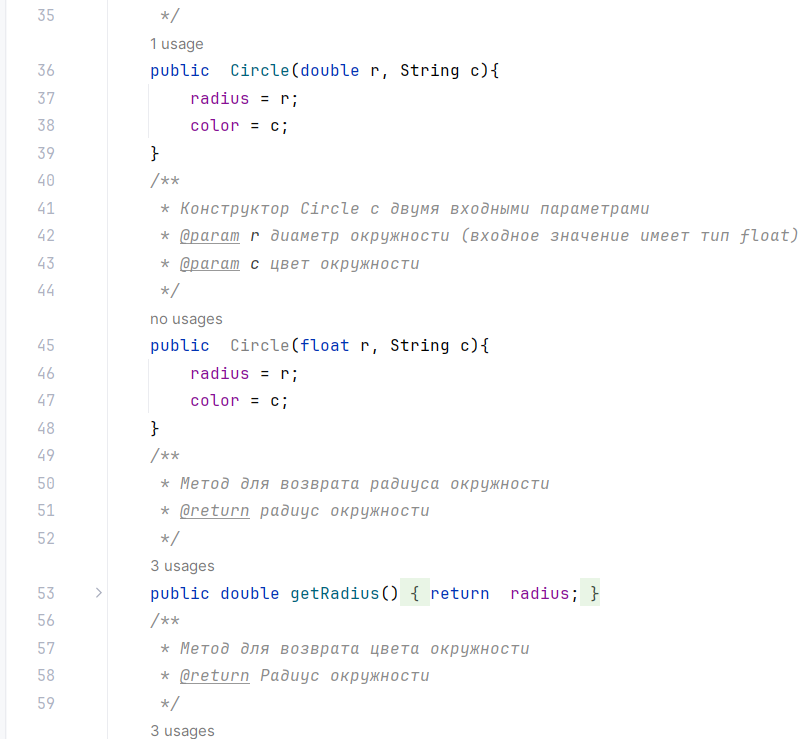
В отличие от метода, конструктор никогда ничего не возвращает.

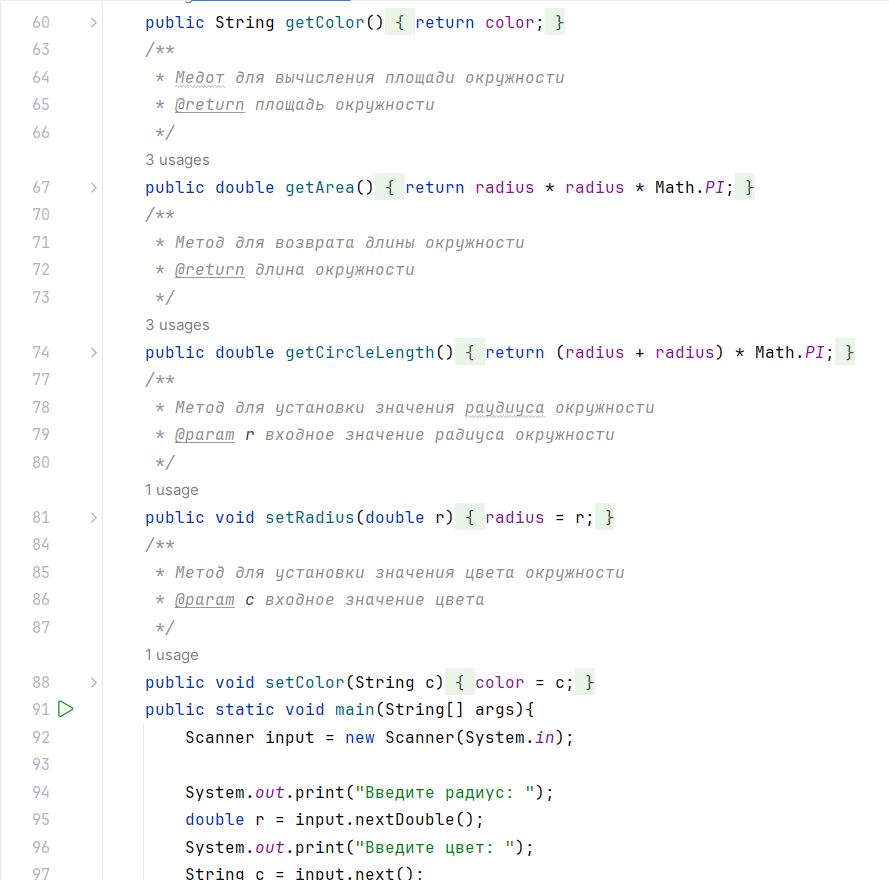
Конструктор определяет действия, выполняемые при создании объекта класса, и является важной частью класса. Как правило, программисты стараются явно указать конструктор. Если явного конструктора нет, то Java автоматически создаст его для использования по умолчанию.

1. **Код программы**





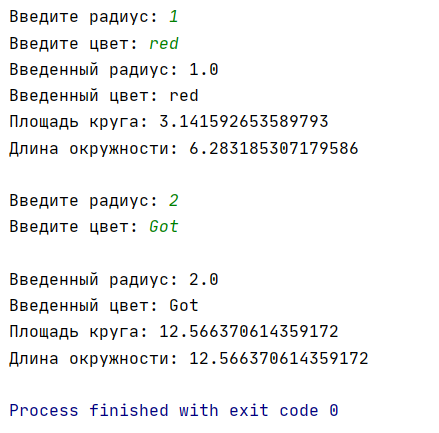








1. **Скриншот выполнения программы**

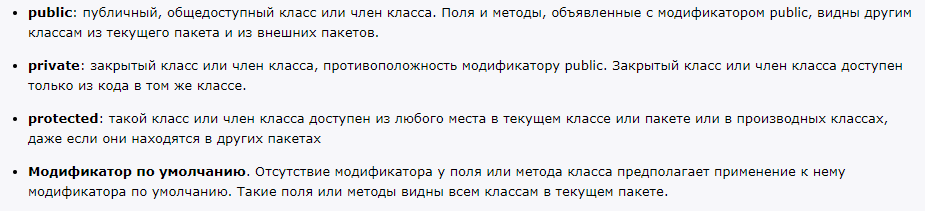


1. **Ответы на контрольные вопросы**
2. Что такое конструктор по умолчанию?

Конструктор по умолчанию (от англ. default constructor) – это конструктор, который не имеет параметров. Конструктор по умолчанию может объявляться в классе явным образом или генерироваться автоматически. Если в классе не объявить ни одного конструктора, то, будет генерироваться конструктор по умолчанию. То есть, конструктор по умолчанию генерируется в классе автоматически только в том случае, если класс не содержит реализации других конструкторов. Если класс содержит реализацию хотя бы одного конструктора с параметрами, то, чтобы объявить конструктор по умолчанию, его нужно объявлять в классе явным образом.

1. Какие модификаторы доступа рекомендуется использовать для полей, метолов?

Если другой класс имеет прямой доступ к этому полю, то есть вероятность, что в процессе работы программы ему будет передано некорректное значение, например, отрицательное число. Подобное изменение данных не является желательным. Либо же мы хотим, чтобы некоторые данные были доступны напрямую, чтобы их можно было вывести на консоль или просто узнать их значение. В этой связи рекомендуется как можно больше ограничивать доступ к данным, чтобы защитить их от нежелательного доступа извне. Для этого лучше использовать модификаторы **public** и **private**.



1. Что такое инкапсуляция?

Инкапсуляция (от англ. *encapsulation*) — в информатике размещение в одном компоненте данных и методов, которые с ними работают. Также может означать скрытие внутренней реализации от других компонентов. Например, доступ к скрытой переменной может предоставляться не напрямую, а с помощью методов для чтения (геттер) и изменения (сеттер) её значения.

Глоссарий

**Класс** (от англ. *class*) — в объектно-ориентированном программировании, представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов.

**Инкапсуляция** (от англ. *encapsulation*) — размещение в одном компоненте данных и методов, которые с ними работают. Также может означать скрытие внутренней реализации от других компонентов. Например, доступ к скрытой переменной может предоставляться не напрямую, а с помощью методов для чтения (геттер) и изменения (сеттер) её значения.

**Конструктор** — это специальный метод, который вызывается при создании нового объекта. Не всегда удобно инициализировать все переменные класса при создании его экземпляра. Иногда проще, чтобы какие-то значения были бы созданы по умолчанию при создании объекта. По сути, конструктор нужен для автоматической инициализации переменных.

**Конструктор по умолчанию** (от англ. default constructor) – это конструктор, который не имеет параметров. Конструктор по умолчанию может объявляться в классе явным образом или генерироваться автоматически. Если в классе не объявить ни одного конструктора, то, будет генерироваться конструктор по умолчанию. То есть, конструктор по умолчанию генерируется в классе автоматически только в том случае, если класс не содержит реализации других конструкторов. Если класс содержит реализацию хотя бы одного конструктора с параметрами, то, чтобы объявить конструктор по умолчанию, его нужно объявлять в классе явным образом.

**new** – ключевое слово для создания нового объекта (или экземпляра) класса.