Министерство высшего образования и науки РФ

Санкт–Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчет**

по дисциплине «Системный подход в разработке ПО»

**Абстрактные классы, исключения и ассерты**

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

доцент, к.т.н \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

Санкт-Петербург

2025

Оглавление

[2. Лекция 2.3 3](#_Toc190780471)

[2.1. Задание 1 3](#_Toc190780472)

[2.2. Пример instanceof 4](#_Toc190780473)

[2.3. Задание 2 6](#_Toc190780474)

[2.4. Пример на кастинг 8](#_Toc190780475)

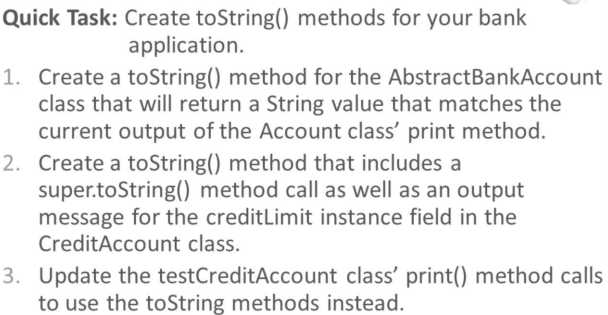
[3. Практика 2.3 9](#_Toc190780476)

[4. Лекция 2.4 14](#_Toc190780477)

[4.1. Обработка ошибок 14](#_Toc190780478)

# Лекция 2.3

## Задание 1



1. toString() в AbstractBankAccount

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. toString() в CreditAcccount

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Вызов печатающих методов

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

Результат:

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Контент, сгенерированный ИИ, может содержать ошибки.

## Пример instanceof

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## Задание 2

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Контент, сгенерированный ИИ, может содержать ошибки.

1,2. Создание метода

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

3. Вызов метода

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Результат:

Изображение выглядит как текст, меню, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## Пример на кастинг

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

# Практика 2.3

## Лексика

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. DownCasting
2. Virtual Method Invocation
3. Instanceof
4. Casting
5. Interface
6. UpCasting
7. Private constructor

## JavaBank

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

1. Определение массива



2, 3. Изменение метода

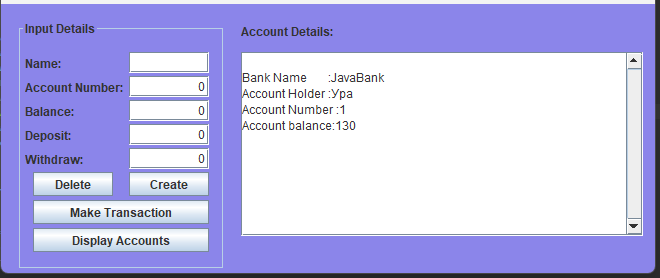
Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

4. Изменение вызова метода



Результат



## Bike

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Контент, сгенерированный ИИ, может содержать ошибки.

1. Необходимость объявить свойства объектов, необходимость иметь конструкторы
2. Сделаем класс абстрактным



1. Удалим создание 4 байка
2. Изменение в классе Bike

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

MountainBike

Изображение выглядит как снимок экрана

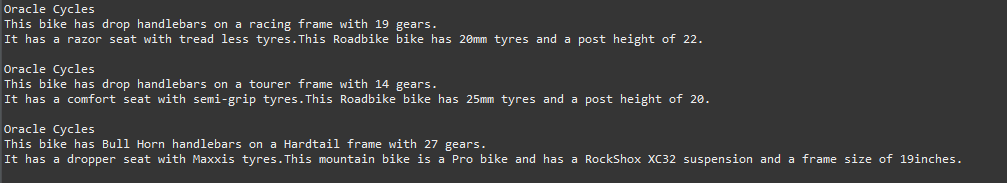
Контент, сгенерированный ИИ, может содержать ошибки.

RoadBike

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Результат



## Animal

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Изображение выглядит как текст, Шрифт, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.**

Кастинг примитивных типов и объектов:

Кастинг примитивных типов меняет занимаемое место в памяти и значения, которые ограничиваются в зависимости от памяти.

Кастинг объектов:

Даункастинг – объект приобретает поля и методы дочернего класса, а также если есть переопределенные методы, то будет использоваться реализация дочернего метода.

Апкастинг – объект теряет все поля и методы, которых нет в родительском классе. При этом будет использоваться родительская реализация всех методов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

# Лекция 2.4

## Обработка ошибок

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

## Try with recources

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

## Multi-catch

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## Method throwing exception

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

## Custom Exception

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## Asserts

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

# Практика 2.4

## Лексика

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Try-with-resources
2. Class Invariant
3. Assertions
4. Try-catch-finally
5. Loop Invariant
6. Conditional Invariant
7. Multi-catch Block
8. Finally Block
9. Checked and Unchecked Exceptions

## JavaBank

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Конспект**

**Методы и объекты**

**Механизм передачи аргументов методам**

Базовое положение состоит в том, что при передаче аргументов методам на самом деле передается техническая, автоматически создаваемая копия аргументов.

**Передача аргументом объекта**

Если посмотреть на ситуацию шире, то легко понять, что при вызове метода нельзя изменить значения аргументов, которые передаются методу. Причина — в механизме передачи аргументов методу: аргументы передаются по значению, поэтому на самом деле метод для обработки получает копии аргументов.

**Объект как результат метода**

Нередко результатом метода должен возвращаться объект. Схема, по которой результатом метода возвращается объект, сводится к тому, что при выполнении метода создается объект, а результатом метода возвращается ссылка на этот объект, которая «технически» реализуется через объектную переменную.

**Объекты и наследование**

**Фабрика объектов**

Напомним, что объектная переменная суперкласса может ссылаться на объект подкласса. Мы воспользуемся этим замечательным обстоятельством для того, чтобы создать метод, возвращающий, в зависимости от фактического значения своего аргумента, объекты разных классов. Ссылка-результат выполняется на объект, созданный в теле метода.

**Конструктор создания копии**

Еще один, важный с практической точки зрения, пример иллюстрирует использования в классе конструктора создания копии. Речь идет об описанном в классе конструкторе, который позволяет создавать на основе уже существующего объекта другой объект (обычно с такими же значениями полей, но это не обязательно).

**Массивы и объекты**

**Массив как поле**

Итак, допустим, что необходимо описать класс, у которого полем является массив. Как это можно сделать? В принципе, достаточно просто: в классе полем объявляется переменная массива, а создание и заполнение собственно массива обычно переносится в конструктор.

**Массив объектов**

При создании массива объектов массив создается из объектных переменных, а уже затем в каждую из переменных записывается ссылка на объект.

**Цепочка объектов**

Еще один способ организации объектов, не подразумевающий применения массива, состоит в том, что создается «цепочка» объектов: это группа объектов, в которой каждый объект (за исключением последнего) содержит ссылку на другой объект.

**Внутренние классы**

Класс может быть описан внутри другого класса. Такой класс называется внутренним . Класс, в котором описан внутренний класс, будем называть внешним или классом-контейнером. Главная и далеко не очевидная особенность внутреннего состоит в том, что он имеет доступ к полям и методам внешнего класса. При этом из внешнего класса нет прямого доступа к полям и методам внутреннего класса (однако доступ к таким полям и методам можно получить только через объект внутреннего класса).

**Анонимные классы**

Достаточно часто возникает необходимость в создании класса, который

используется в программе один-единственный раз. В таких случаях нередко используют анонимные классы — классы, у которых нет «собственного имени». Мы рассмотрим два способа создания анонимных классов: путем наследования абстрактного класса и путем реализации интерфейса.

**Создание анонимного класса путем наследования абстрактного суперкласса**

Допустим, имеется некоторый абстрактный класс, который назовем условно Суперкласс. Анонимный класс будем создавать наследование Суперкласса. Но на самом деле создание анонимного класса подразумевает, что на основе этого анонимного класса сразу же создается объект. Другими словами, нет смысла говорить об описании анонимного класса, если речь не идет о создании объекта. В принципе, объект тоже может быть анонимным. Но может быть и так, что ссылка на объект, созданный на основе анонимного класса, записывается в объектную переменную. Ссылка на объект анонимного класса может быть записана в переменную Суперкласса, на основе которого «конструируется» анонимный класс. Используется следующий шаблон:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

То есть сама конструкция создания объекта на основе анонимного класса начинается, как и положено, с инструкции new, после которой указан конструктор суперкласса, а в фигурных скобках — код, содержащий описание абстрактных методов. Если речь идет об анонимном объекте, созданном на основе анонимного класса, то, скорее всего, такой анонимный объект передается аргументом какому-то методу или из него самого вызывается метод. В любом случае используется конструкция вида:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

**Создание анонимного класса через реализацию интерфейса**

Еще один часто используемый способ создания объектов анонимного класса базируется на реализации в анонимном классе интерфейса. Подход здесь абсолютно такой же, как и в случае с наследованием класса — с поправкой на то, что имеем дело с интерфейсом. Ниже представлен шаблон создания объекта на основе анонимного класса, реализующего Интерфейс, при условии, что ссылка на созданный объект записывается в интерфейсную переменную:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Анонимный объект анонимного класса при реализации в последнем интерфейса может создаваться кодом следующего вида:

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Резюме**

При передаче аргументов методам на самом деле передаются копии аргументов, которые создаются автоматически. Поэтому в теле метода нельзя изменить значение аргумента.

• При передаче аргументом методу объекта на самом деле передается ссылка на объект. Если нужно, чтобы метод возвращал объект результатом, то в теле метода создается соответствующий объект, а ссылка на него возвращается результатом метода.

• Если необходимо описать класс с полем-массивом, то полем является переменная массива, значением которой присваивается ссылка на массив. Массив обычно создается при вызове конструктора. Аналогичным образом поступают, если необходимо описать класс с полем-объектом: полем является объектная переменная, а значением ей присваивается ссылка на объект.

• Для создания массива объектов создается массив из объектных переменных, значениями которым присваиваются ссылки на объекты.

• Класс может быть описан в классе. Такой класс называется внутренним. Внутренний класс имеет доступ к полям внешнего класса.

• Объекты могут создаваться на основе анонимных классов. Такие анонимные классы создают, как правило, или наследованием абстрактного класса, или реализацией интерфейса.

**Примеры кода**

**Пример 1**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.**

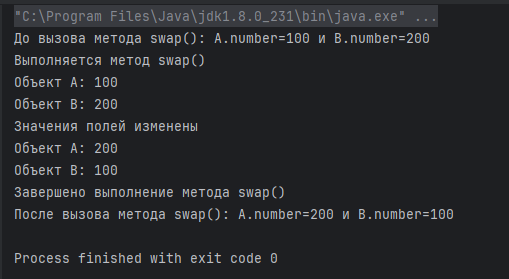
**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 2**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

****

**Пример 3**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 4**

**Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 5**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 6**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 7**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 8**

**Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

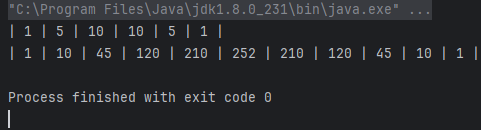
**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 9**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

****

**Пример 10**

**Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 11**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Пример 12**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**