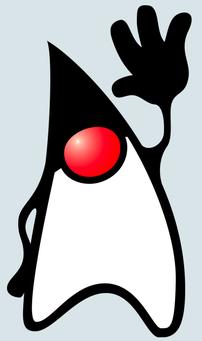
СБЕРБАНК

Корпоративный университет



???

Занятие №4

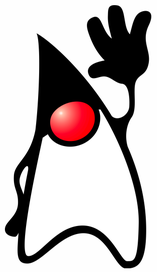
СБЕРБАНК
Корпоративный университет

Куценко Виталий

Что сможем после занятия

СБЕРБАНК
Корпоративный университет

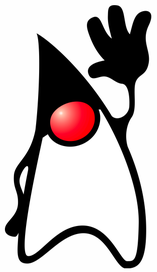
- Исключения в Java
- Основы архитектуры разработки. SOLID
- Работа с файлами в Java
- Тестирование. Mockito



Договорённости

СБЕРБАНК
Корпоративный университет

- Активно участвуем. Не стесняйтесь задавать вопрос.
- Но off-topic обсуждаем в Telegram @sb_ku_java_2019_10
- Не стесняйтесь просто спрашивать в telegram.
- В конце с Вас отзыв.
- ДЗ - доделываем текущую, начнем новую в следующий раз



Договорились?
Поехали!

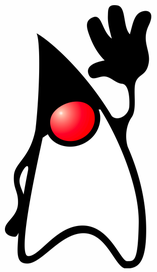
00

ДЗ. На что обратить внимание

Основные вопросы

СБЕРБАНК
Корпоративный университет

- .gitignore https://git-scm.com/docs/gitignore
- Code convention https://google.github.io/styleguide/javaguide.html



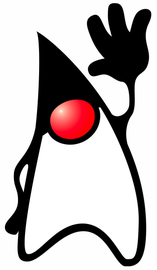
Ваши вопросы?

Если что – их можно задать потом

Что сможем после занятия

СБЕРБАНК
Корпоративный университет

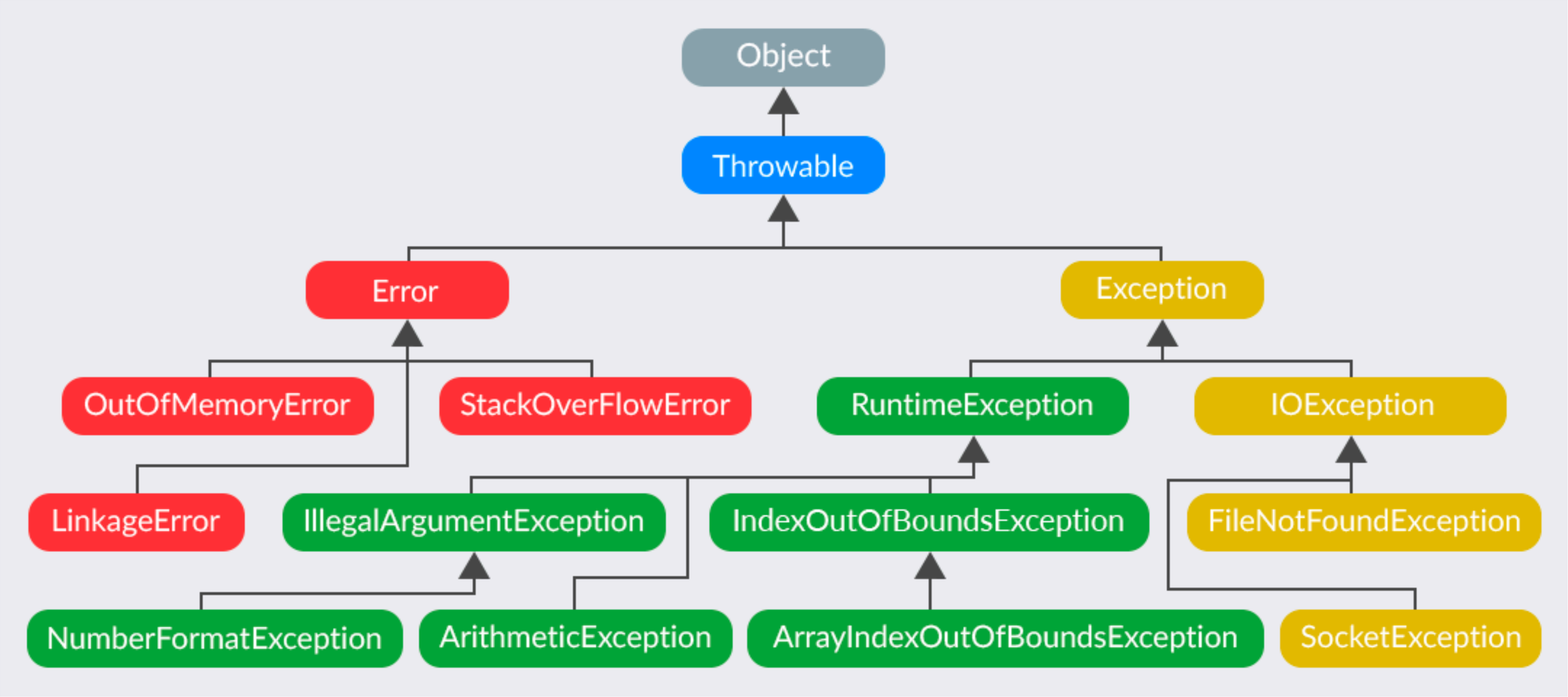
- **Исключения в Java**
- Основы архитектуры разработки. SOLID
- Работа с файлами в Java
- Тестирование. Mockito

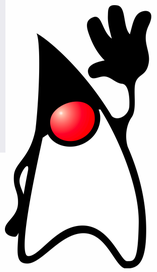


01

Исключения

Основные вопросы



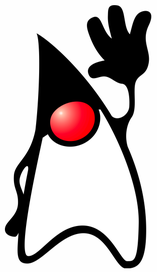


Исключения в Java. Объявление

СБЕРБАНК
Корпоративный университет

```
public class TestCheckedException extends Exception {
}


public class TestUncheckedException extends RuntimeException {
}
```



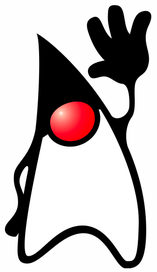
Исключения в Java. Использование

СБЕРБАНК
Корпоративный университет

```
public class HelloWorld{
    public static void main (String args[]){
        HelloWorld hw = new HelloWorld();
        try {
            hw.checkedException();
        } catch ( TestCheckedException e ) {
            e.printStackTrace();
        }
        hw.uncheckedException();
    }

    public void checkedException() throws TestCheckedException {
        throw new TestCheckedException();
    }

    public void uncheckedException(){
        throw new TestUncheckedException();
    }
}
```



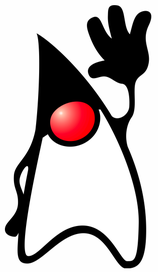
Ваши вопросы?

Если что – их можно задать потом

Что сможем после занятия

СБЕРБАНК
Корпоративный университет

- Исключения в Java
- **Основы архитектуры разработки. SOLID**
- Работа с файлами в Java
- Тестирование. Mockito



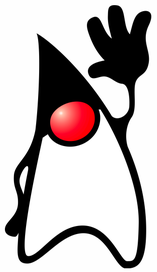
02

Основы архитектуры разработки. SOLID

Идеальная архитектура. Технический подход

СБЕРБАНК
Корпоративный университет

- Оптимальные алгоритмы. Достаточно быстрые, но не заумные
- Классы, переменные, методы названы понятно
- Отдельные классы и методы можно переиспользовать
- Классы и методы лаконичны и понятны
- Хорошее покрытие тестами (Unit и интеграционные)

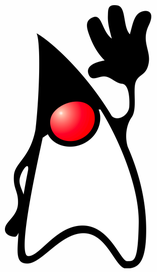


Идеальная архитектура. Бизнес подход

СБЕРБАНК
Корпоративный университет

- Сам код ничего не стоит и никому не нужен. Нужен только **функционал**

- Два критерия:
  - готов к нужной дате
  - делает то, что требуется

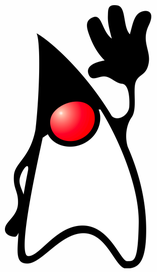


СБЕРБАНК
Корпоративный университет

В этом мире не всё всегда и везде,
а кое-что, иногда и местами.

Максим Дорофеев.
Врач-прокрастинолог

Если нужен функционал «прямо щас» для разовой акции,
то и copy-past - подходящее решение.



Couplig и Cohesion

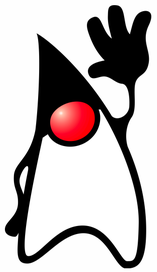
СБЕРБАНК
Корпоративный университет

- **Coupling** (зацепление, сцепление, сопряжение) — способ и степень взаимозависимости между программными модулями; сила взаимосвязей между модулями; мера того, насколько взаимозависимы разные подпрограммы или модули.

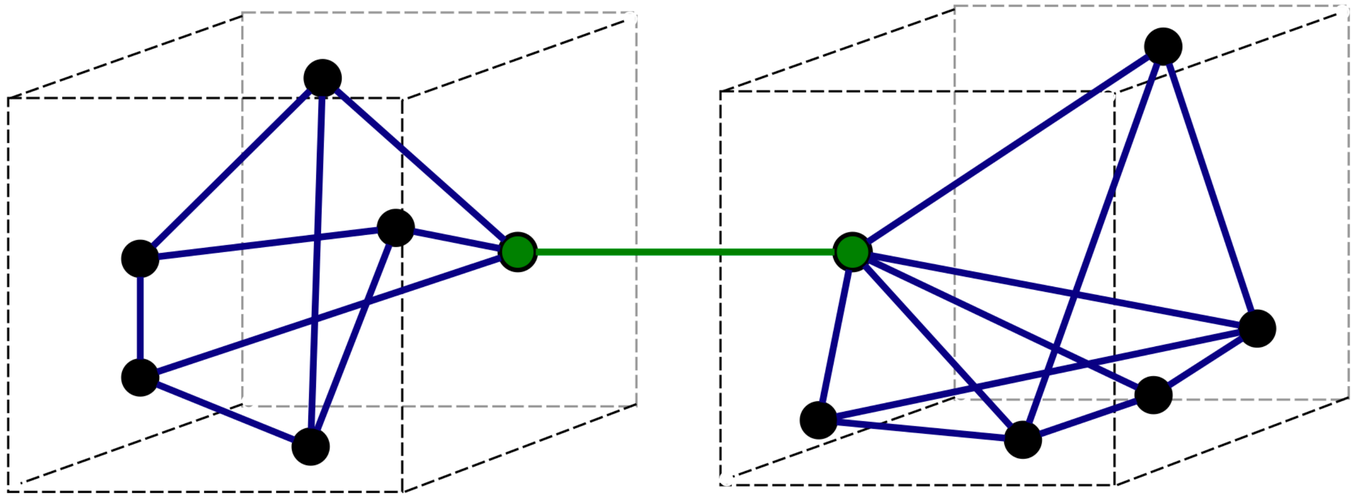
  Чем меньше - тем лучше

- **Cohesion** (связность, или прочность) — мера силы взаимосвязанности элементов внутри модуля; способ и степень, в которой задачи, выполняемые некоторым программным модулем, связаны друг с другом.

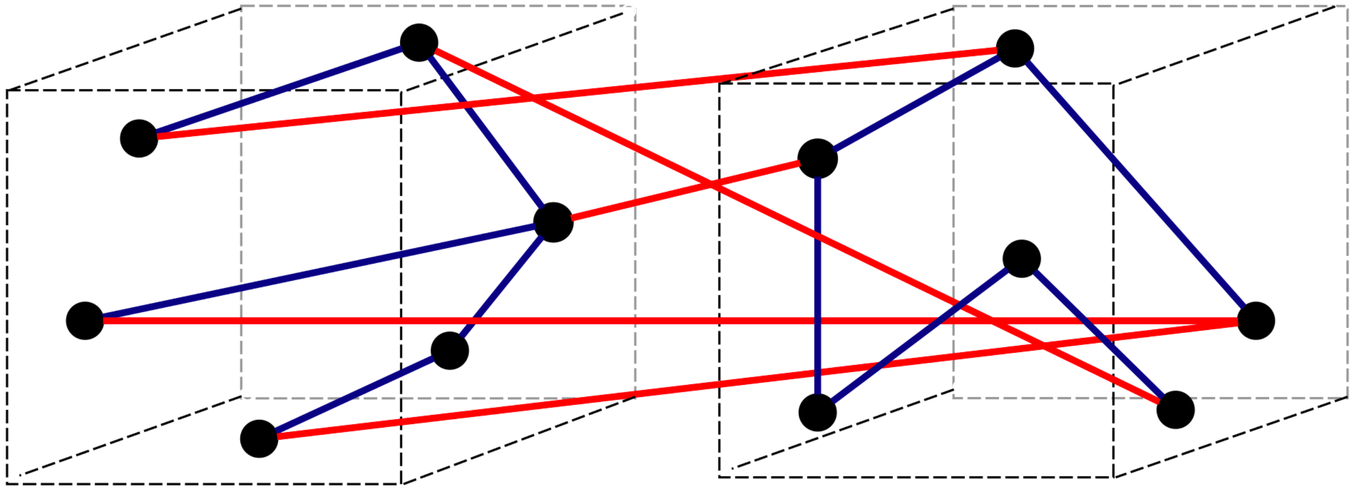
  Чем больше - тем лучше



Couplig и Cohesion



a) Слабое зацепление, сильная связность



b) Сильное зацепление, слабая связность

Полиморфизм

СБЕРБАНК
Корпоративный университет

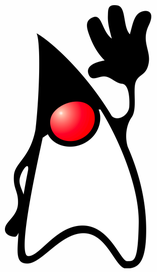
Три кита ООН:

- Наследование
- Инкапсуляция
- **Полиморфизм**

https://github.com/kataus/sbrf_cu_2019_l04_solid см. poly

В языках программирования и теории типов полиморфизмом называется способность функции обрабатывать данные разных типов.

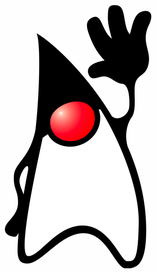
https://ru.wikipedia.org/wiki/Полиморфизм_(информатика)



Принципы SOLID

СБЕРБАНК
Корпоративный университет

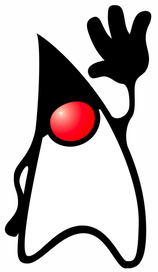
- **S**ingle Responsibility
- **O**pen/Close
- **L**iskov Substitution
- **I**nterface Segragation
- **D**ependency Inversion



Single Responsibility

СБЕРБАНК
Корпоративный университет

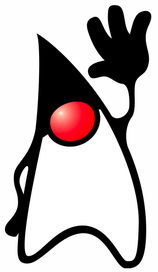
- Должна быть только одна причина для изменения класса
- Класс должен выполнять одну функцию, т.е. решать одну проблему



Open/Close

СБЕРБАНК
Корпоративный университет

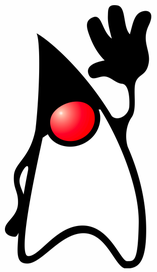
- Открыт для расширения / Закрыт для изменений
- Для добавления нового функционала не надо менять существующий код. Править можно только баги.



Liskov Substitution

СБЕРБАНК
Корпоративный университет

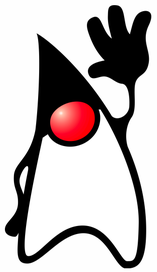
- Класс потомок должен вести себя как родитель
- Предопределенные методы не должны бросать новые исключения или выполнять дополнительные действия



Interface Segragation

СБЕРБАНК
Корпоративный университет

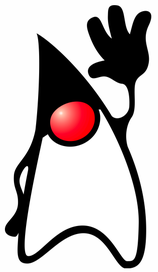
- Интерфейсы должны быть максимально «узкие» и специфичные
- Класс, имплементирующие интерфейс, не должны принуждаться имплементировать методы интерфейса, которые ему не нужны
- Универсальный, «широкий» интерфейс должен быть разделен на несколько более «узких», специфичных



Dependency Inversion

СБЕРБАНК
Корпоративный университет

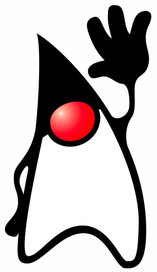
- Высокоуровневые модули не должны зависеть от низкоуровневых
- Высокоуровневые модули должны зависеть от абстракций



Другие аббревиатуры

СБЕРБАНК
Корпоративный университет

- **DRY** - don't repeat yourself
- **KISS** - keep it simple stupid
- **YAGNI** - You Aren't Gonna Need It

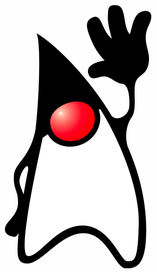


Что сможем после занятия

СБЕРБАНК
Корпоративный университет

- Исключения в Java
- Основы архитектуры разработки. SOLID

- **Работа с файлами в Java**
- Тестирование. Mockito

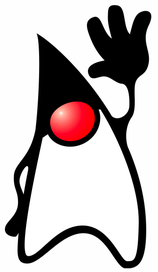


Что сможем после занятия

СБЕРБАНК
Корпоративный университет

- Исключения в Java
- Основы архитектуры разработки. SOLID
- Работа с файлами в Java

- **Тестирование. Mockito**



Ваши вопросы?

Спасибо за внимание!