



*School of  
Computer  
Science*

# ОБСУЖДЕНИЕ ПРЕДЫДУЩИХ ЗАНЯТИЙ



# ОПРЕДЕЛЕНИЯ РЕФЛЕКСИЯ

- **Рефлексия** – это механизм, который позволяет анализировать и модифицировать структуру и поведение программы, а также получать доступ к данным во время ее выполнения
- Применяется при тестировании, отладке, в интегрированных средах разработки (IDE), библиотеках и фреймворках Java (например Lombok, Spring, Hibernate и т.д.)
- Реализуется с помощью объектов-представлений о классах, методах, полях и т.д. – это объекты следующих классов:
  - Class
  - Method
  - Field
  - ...



## ПОЛУЧЕНИЕ МЕТОДОВ ЧЕРЕЗ РЕФЛЕКСИЮ

Имеем следующие переменные:

```
Object obj = new <любой_класс>;
```

```
Class cls = obj.getClass();
```

Какой из вызовов вернет только доступные методы объекта, включая и определенные в его родительских классах?

1. `cls.getMethods();`

2. `cls.getDeclaredMethods();`



## ПОЛУЧЕНИЕ ПОЛЕЙ ЧЕРЕЗ РЕФЛЕКСИЮ

Имеем следующие переменные:

```
Object obj = new <любой_класс>;
```

```
Class cls = obj.getClass();
```

Какой из вызовов вернет все поля объекта, которые определены в его классе без учета модификаторов видимости?

1. `cls.getFields();`

2. `cls.getDeclaredFields();`



# ВОПРОС

## ДОСТУП К ЗАЩИЩЕННЫМ ПОЛЯМ

Имеем следующие переменные:

```
Object obj = new <любой_класс>;
```

```
Class cls = obj.getClass();
```

```
Field field = cls.getDeclaredField("name");
```

Какая строка кода необходима, чтобы очередным вызовом `field.get()` добраться до значения `private` поля `name`?

1. `field.setAccessible(true);`
2. `field.accessSetable(true);`
3. `field.accessDenied(false);`



# ОПРЕДЕЛЕНИЯ КЛАССЫ-ОБЕРТКИ

- **Классы-обертки** – это специальные классы из стандартной библиотеки Java, которые расширяют возможности использования примитивных типов данных
- В отличие от примитивов, классы-обертки:
  - хранят не значение, а ссылку на него
  - не имеют закрепленного диапазона значений
  - в качестве значения по умолчанию используют `null`
  - обладают своими методами
- Имена классов-оберток являются производными от названий примитивов и пишутся в коде с заглавной буквы



# ОПРЕДЕЛЕНИЯ ПРИМИТИВЫ И КЛАССЫ-ОБЕРТКИ

## Соответствие примитивов и обёрток

● Примитив

● Обёртка

● Примитив

● Обёртка

byte

Byte

short

Short

int

Integer

long

Long

float

Float

double

Double

char

Character

boolean

Boolean

[Сравнение примитивов и оберток в коде](#)



# ОПРЕДЕЛЕНИЯ ИНТЕРФЕЙС

- **Интерфейс** (**interface**) – набор сигнатур методов, которые описывают функциональность объектов относящихся к интерфейсу, но не реализуют её
- Интерфейс имплементируется (**implements**) в его классах-реализациях, которые наполняют методы конкретным поведением
- Можно воспринимать интерфейс как контракт: класс, который реализует интерфейс, обязуется выполнять методы, описанные в нём





# ВОПРОС

## СВОЙСТВА ИНТЕРФЕЙСОВ JAVA

- Что из описанного характерно для интерфейсов в Java?
  1. Все описанные методы по умолчанию имеют модификатор доступа **public**
  2. Если в интерфейсе есть поля, то они **public static final**
  3. Реализация интерфейса в классе делается с помощью ключевого слова **implements**
  4. В классе, реализующем интерфейс, нельзя сужать область доступа для методов этого интерфейса
  5. При реализации интерфейса в классе требуется определить все методы интерфейса, у которых отсутствует квалификатор **default**



# ОПРЕДЕЛЕНИЯ ПЕРЕЧИСЛЕНИЕ

- **Перечисление** (`enum`) – специальный тип данных для хранения ограниченного множества значений
- Преимущества использования перечислений в программе Java:
  - Улучшение читаемости кода
  - Безопасность типов и значений констант
  - Удобный синтаксис сравнения (`==`) и оператора `switch`



# ВОПРОС

## ГДЕ ПОДХОДИТ ПЕРЕЧИСЛЕНИЕ?

- Какие из этих значений удобнее всего хранить в перечислении, а какие нет?

Ассортимент  
магазина  
канцтоваров

Статусы  
сообщений в  
журнале событий

Список  
покупок в  
магазине

Планеты Солнечной  
системы

Текст песни

Времена года

Ноты

Порядок выступлений  
артистов концерта

Имена учеников  
в классе

**СПАСИБО ЗА ВНИМАНИЕ !**  
**ВОПРОСЫ ?**



*School of  
Computer  
Science*