Универсальные шаблоны

№ урока: 10 **Курс:** Java Fundamentals

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Рассмотрение универсальных типов.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу параметризированных типов.
- Использовать ограничения параметризированных типов.
- Использовать параметризированные коллекции (List<T>, Dictionary<TKey,TValue>).

Содержание урока

1. Обобщения.
2. Ковариантность и контрвариантность обобщений.
3. Ограничения обобщений.
4. Преимущества использования обобщенных коллекций.

Резюме

- Обобщение – элемент кода, способный адаптироваться для выполнения общих (сходных) действий над различными типами данных.
- Обобщения обеспечивают большую производительность, так как не происходит операции "упаковки-распаковки"(boxing-unboxing).
- Обобщения обеспечивают безопасность типов, так как могут содержать только типы, которые Вы задаете при объявлении.
- Обобщения позволяют создавать открытые (open-ended) типы, которые преобразуются в закрытые во время выполнения.
- Идентификатор <T> – это указатель места заполнения, вместо которого подставляется любой тип.
- Перегрузки обобщенных типов различаются количеством параметров типа, а не их именами. Правильная перегрузка: `MyClass<T>{ }, MyClass<T,R>{ }`
- Пример открытого типа: `MyClass<T>`
- Пример закрытого типа: `MyClass<String>`
- Каждый закрытый тип получает свою собственную копию набора статических полей.
- Общие сведения об универсальных шаблонах:
    - Используйте универсальные типы для достижения максимального уровня повторного использования кода, безопасности типа и производительности.
    - Наиболее частым случаем использования универсальных шаблонов является создание классов коллекции.
    - Можно создавать собственные универсальные интерфейсы, классы, методы и так далее.
    - Доступ универсальных классов к методам можно ограничить определенными типами данных
- Ковариантность обобщений – upcast параметров типов.
- Контрвариантность обобщений – downcast параметров типов.
- При определении универсального типа можно ограничить виды типов, которые могут использоваться клиентским кодом в качестве аргументов типа при инициализации

WebCamp
Webcamp 2015
10 Darvina Str., 13 office
Kyiv, Ukraine
t. +380 (44) 235-6117
E-mail: info@webcamp.com.ua
Site: webcamp.com.ua

соответствующего класса. При попытке клиентского кода создать экземпляр класса с помощью типа, который не допускается ограничением, в результате возникает ошибка компиляции. Это называется ограничениями.

- Ограничения определяются с помощью контекстно-зависимого ключевого слова extends.
- <T extends base_class_name> - Аргумент типа должен являться или быть производным от указанного базового класса
- Коллекция `ArrayList` - коллекция с динамическим увеличением размера до нужного значения.
- Емкость коллекции `ArrayList` — это количество элементов, которое может содержать `ArrayList`. При добавлении элементов в коллекцию `ArrayList` ее емкость автоматически увеличивается нужным образом за счет перераспределения внутреннего массива.
- Если размер коллекции можно оценить, целесообразно указать для нее начальную емкость, чтобы избавиться от необходимости выполнять операции изменения размера при добавлении элементов в коллекцию `ArrayList`.
- `Dictionary`<TKey, TValue> - класс представляет коллекцию ключей и значений.

Закрепление материала

1. Что такое обобщение?
2. Что такое закрытый тип?
3. Что такое открытый тип?
4. Объясните понятия ковариантности и контрвариантности обобщений.
5. Какие преимущества использования обобщений?
6. Какие вы знаете типы ограничений для обобщений?
7. Что такое ограничение обобщений?
8. Какие обобщенные коллекции вы знаете?

Дополнительное задание

Задание

Создайте проект, используя IntelliJ IDEA.

Создайте класс `MyClass<T>`, содержащий статический фабричный метод - `T FacrotyMethod()`, который будет порождать экземпляры типа, указанного в качестве параметра типа (указателя места заполнения типом – T).

Самостоятельная деятельность учащегося

Задание 1
Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2
Создайте проект, используя IntelliJ IDEA.

Создайте класс `MyList<T>`. Реализуйте в простейшем приближении возможность использования его экземпляра аналогично экземпляру класса `List<T>`. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления элемента, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества элементов.

Задание 3

Webcamp 2015
10 Darvina Str., 13 office
Kyiv, Ukraine

t. +380 (44) 235-6117
E-mail: info@webcamp.com.ua
Site: webcamp.com.ua

Создайте проект, используя IntelliJ IDEA.

Создайте класс `MyDictionary<TKey,TValue>`. Реализуйте в простейшем приближении возможность использования его экземпляра. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления пар элементов, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества пар элементов.

Задание 4

Зайдите на сайт Oracle.

Используя поисковые механизмы Oracle, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

Универсальные шаблоны
https://docs.oracle.com/javase/tutorial/java/generics/

Универсальные методы
https://docs.oracle.com/javase/tutorial/java/generics/methods.html

Ограничения параметров типа
https://docs.oracle.com/javase/tutorial/java/generics/boundedTypeParams.html

WebCamp

Webcamp 2015
10 Darvina Str., 13 office
Kyiv, Ukraine

t. +380 (44) 235-6117
E-mail: info@webcamp.com.ua
Site: webcamp.com.ua