

Задание 8.
Работа с рефлексией

8.1. Напишите консольное приложение, которое:

- создает и инициализирует переменную типа **String**;
- использует рефлексия для изменения значения созданной строки (через доступ к закрытому полю *char[] value* класса **String**).

ТРЕБОВАНИЯ.

1. Приложение должно быть написано на языке Java.
2. Использовать только стандартные компиляторы и библиотеки.
3. При кодировании должны быть использованы соглашения об оформлении кода для языка Java.
4. Строка должна быть создана и проинициализирована двумя способами:
 - ✓ как строковый литерал;
 - ✓ введена набором с клавиатуры.
5. Значение для замены может быть задано программно или введено набором с клавиатуры.
6. Строка должна быть выведена до изменения и после.

8.2. Напишите консольное приложение, которое:

- описывает класс "Унарная арифметическая операция":
 - закрытое поле вещественного типа;
 - несколько конструкторов;
 - методы доступа (геттеры и сеттеры);
 - метод *toString()*;
 - методы инкремента, декремента, изменения знака.
- создает экземпляр описанного класса через механизм рефлексии (любым конструктором);
- отображает информацию о состоянии созданного экземпляра через рефлексия (экземпляр класса **Field**);
- изменяет состояние экземпляра динамическим вызовом метода доступа.

ТРЕБОВАНИЯ.

1. Приложение должно быть написано на языке Java.
2. Использовать только стандартные компиляторы и библиотеки.
3. При кодировании должны быть использованы соглашения об оформлении кода для языка Java.
4. Информация об экземпляре должна быть выведена в консоль после создания и изменения.

Работа с лямбда-выражениями

8.3. Напишите консольное приложение, которое:

- создает массив типа `Integer` и заполняет его со случайными значениями, затем упорядочивает значения массива в порядке убывания, используя лямбда-выражения для указания порядка сортировки;
- создает список строк и сортирует его в порядке обратном алфавитному, используя лямбда-выражения для указания порядка сортировки.

Примечание: необходимо вывести массив и список до и после сортировки.

8.4. Откройте приложение, разработанное по заданию 8.3, и внесите следующие изменения:

- добавьте метод, который получает массив, условие выборки значений из массива, указанное с помощью лямбда-выражения, и возвращает результат некоторой обработки выбранных элементов массива;
- добавьте метод, который получает список, условие выборки строк с помощью лямбда-выражения – строки, начинающиеся с указанной буквы, и возвращает выбранные строки.

ТРЕБОВАНИЯ.

1. Приложение должно быть написано на языке Java.
2. Использовать только стандартные компиляторы и библиотеки.
3. При кодировании должны быть использованы соглашения об оформлении кода для языка Java.
4. Применить ссылки на методы, где возможно.

Работа с лямбда-выражениями и функциональными интерфейсами

8.5. Откройте приложение, разработанное по заданию 8.4, и внесите следующие изменения:

- добавьте описание функционального интерфейса с абстрактным методом конвертации строки и методом проверки строки на существование;
- метод, который преобразует строки списка: все символы в верхний регистр, используя лямбда-выражение для реализации интерфейса.

ТРЕБОВАНИЯ.

1. Приложение должно быть написано на языке Java.
2. Использовать только стандартные компиляторы и библиотеки.
3. При кодировании должны быть использованы соглашения об оформлении кода для языка Java.
4. Применить ссылки на методы, где возможно.

Работа с поточной обработкой данных

8.6. Напишите консольное приложение, которое:

- создает и выводит список целых чисел с использованием методов **Stream**: сначала создать бесконечную последовательность чисел, начиная с 10 и каждое следующее увеличением на 10, затем отобрать первые 10 чисел, которые уменьшить в 2 раза каждое, и собрать их в список;
- описывает класс **Person** с полями: имя, возраст и пол, методами: *getter()* и *setter()*, конструктором и переопределенным методом *toString()*;
- создает коллекцию объектов типа **Person** и, используя **Stream**, выбирает и выводит на консоль всех военнообязанных мужчин;
- определяет для коллекции объектов типа **Person** средний возраст женщин с использованием **Stream**.

ТРЕБОВАНИЯ.

1. Приложение должно быть написано на языке Java.
2. Использовать только стандартные компиляторы и библиотеки.
3. При кодировании должны быть использованы соглашения об оформлении кода для языка Java.
4. Применить ссылки на методы, где возможно.