

ПРАКТИЧЕСКАЯ РАБОТА 1-1

ЗАДАНИЕ:

1. Создайте проект *PracticalWork36* с пакетом *com.alpha*, в котором создайте класс **Task36** с методом *public static void main(String[] args)*.
2. Добавьте в пакет *com.alpha* пакет **work1**, в котором опишите класс **Runner** с методом *run()*, а также добавьте в код метода *main()* создание экземпляра класса **Runner** и вызов на нем метода *run()*.
3. Добавьте в класс **Runner** метод *findMax(int... array)*, который получает параметр переменной арности и возвращает максимальное значение.
4. Добавьте в метод *run()* код отображения в консоль результата вызова метода *findMax()* с различными аргументами.
5. Запустите приложение на исполнение.
6. Добавьте к описанию метода *findMax()* аннотацию `@Deprecated` и прокомментируйте возникшую ситуацию.
7. Добавьте к описанию метода *run()* аннотацию `@SuppressWarnings("deprecation")` и прокомментируйте возникшую ситуацию.
8. Добавьте в класс **Runner** обобщенный метод *findMax(T... array)*, который получает параметр переменной арности типа **T** и возвращает максимальное значение. Прокомментируйте возникшую ситуацию.
9. Добавьте к описанию метода *findMax()* аннотацию `@SafeVarargs` и прокомментируйте возникшую ситуацию.

ТРЕБОВАНИЯ

1. Приложение должно быть написано на языке Java.
2. Используйте *System.out.println()* для вывода результата.

ПРАКТИЧЕСКАЯ РАБОТА 1-2

1. Откройте проект *PracticalWorkB36* и добавьте в пакет *com.alpha* пакет *work2*, в котором опишите класс **Runner** с методом *run()* и сделайте так, чтобы в методе *main()* класса **Task36** вызывался именно этот метод *run()*.
2. Добавьте в класс **Runner** пакета *work2* описание перечисления **PermissionAction**, которое определяет операции пользователя с файлами (например, *USER_READ*, *USER_CHANGE*, ...).
3. Добавьте в пакет *com.alpha.work2* описание класса **User** (поля: *name* типа *String* и *permissions* типа **List<PermissionAction>**; конструктор с параметром *name*; метод *toString()*, который отображает имя пользователя; геттер для получения списка разрешений и метод добавления нового разрешения).
4. Добавьте в пакет *com.alpha.work2* описание аннотации **MyPermission** с политикой удержания *RetentionPolicy.RUNTIME* и методом *value()*, который возвращает значение типа **PermissionAction**.
5. Добавьте в пакет *com.alpha.work2* описание класса **Action** с методами для чтения и записи в файл. Методы должны быть аннотированы *@MyPermission* с соответствующими правами доступа, а также получать пользователя, который хочет их выполнить (метод принимает решение: разрешить доступ к файлу и выполнить работу или отказать в доступе и выдать сообщение).
6. Добавьте в метод *run()* класса **Runner** пакета *com.alpha.work2* код создания массива пользователей с различными правами и экземпляра класса **Action**. Затем вызовите на экземпляре класса **Action** методы записи и чтения из файла для различных пользователей.
7. Запустите приложение на исполнение.

ТРЕБОВАНИЯ

1. Приложение должно быть написано на языке Java.
2. Используйте метод *getAnnotation()* для получения аннотации метода.