

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

Практический пример №1
Создание простого консольного приложения на языке Java

Составил: Старший преподаватель кафедры «Информатика»
Черниговский Алексей Сергеевич

Красноярск 2020

Содержание

1 IDE.....3

2 Создание нового проекта.....3

3 Считывание данных с консоли.....4

4 Запуск приложения.....6

1 IDE

Для работы с Spring Framework (в дальнейшем), нам, естественно, понадобится IDE. Вы вольны выбирать IDE на свой вкус, но в данном примере и далее будет использоваться IDE Eclipse. Данная IDE является бесплатной и широко используемой.

Многие отдают предпочтение IDE IntelliJ Idea, но его возможностей Community версии недостаточно для разработки на Spring.

Скачать IDE можно по ссылке <https://www.eclipse.org/downloads/packages/>, выбрав Eclipse IDE for Enterprise Java Developers (Рисунок 1).



Рисунок 1 — Скачивание Eclipse IDE

Выберите необходимую вам операционную систему. Здесь и далее примеры будут приводиться для ОС Windows (в частности Windows 10).

Обратите внимание, что по данной ссылке (Рисунок 1) вы скачиваете не установочный файл, а готовую к работе IDE, которую достаточно распаковать в необходимую папку и приступить к работе. Кому это необходимо, может скачать файлы установки и установить Eclipse на свой ПК.

2 Создание нового проекта

Создаем новый проект. File → New Project → Java Project

Задайте имя проекту, в моем случае это будет firstApp.

(Ожидается что слушатель уже установил JDK и JRE и в процессе создания IDE Eclipse предложила в качестве JRE уже установленную версию)

В папке src создаем класс Main, в нем прописываем main метод.

```
public class Main {
    public static void main(String[] args){

    }
}
```

Заготовка проекта готова. Продемонстрируем простой пример.

3 Считывание данных с консоли

Продемонстрируем простой пример. Создадим новый класс Person с одним числовым полем.

```
public class Person {
    private int age;
}
```

Ожидается что слушатель понимает, что в одном java-файле может содержаться только один public класс и разместит данный класс не в том же файле, где и класс Main.

Для автоматического создания сеттеров и геттеров в Eclipse IDE нажмите меню Source → Generate Getters and Setters (Рисунок 2) и выберите необходимые методы.

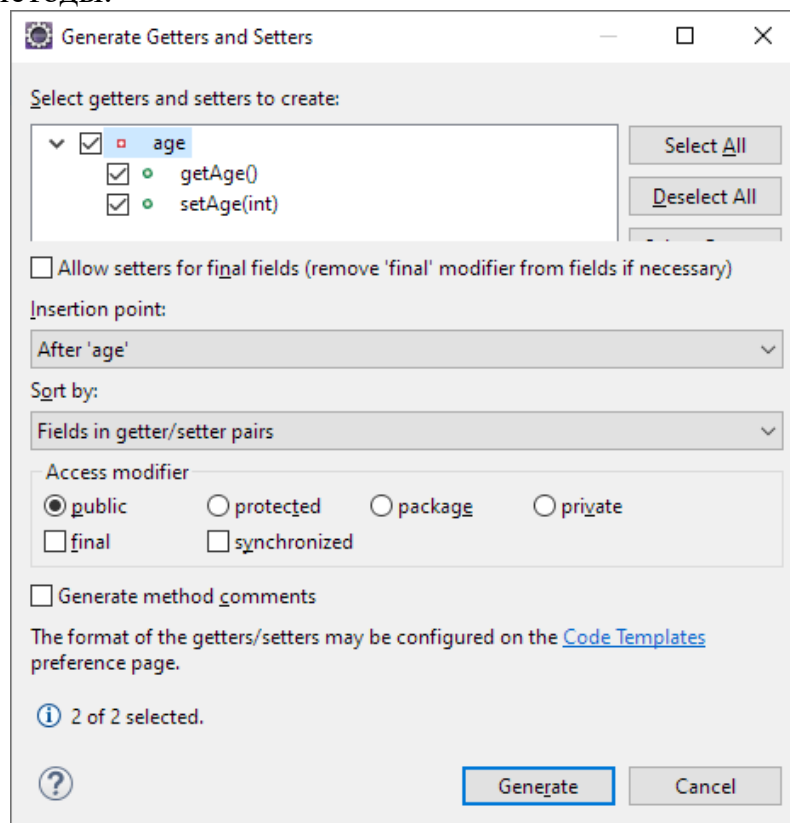


Рисунок 2 — Автогенерация сеттеров и геттеров

Для удобного считывания данных нам понадобится класс Scanner:

```
Scanner scanner = new Scanner(System.in);
```

Для удобного добавления необходимых пакетов (сначала IDE не понимает что такое класс Scanner, необходимо осуществить импорт) можно кликнуть в правой части области кода на иконку лампочки и выбрать import необходимого пакета (Рисунок 3).

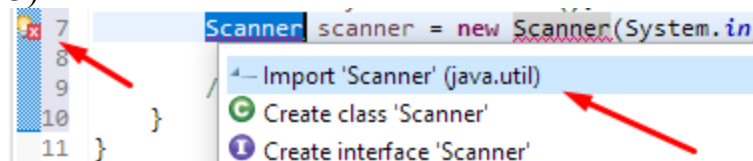


Рисунок 3 — Импорт с помощью IDE

Далее прокомментируем полный код примера:

```
public class Main {
    private static final int MIN_AGE = 0;
    private static final int MAX_AGE = 100;

    public static void main(String[] args){
        Person vasya = new Person();
        Scanner scanner = new Scanner(System.in);

        int age = 0;
        while(true) {
            if(!scanner.hasNextInt()) {
                System.out.println("Некорректные данные,
                                   попробуйте снова!");
                scanner.next(); // считываем все данные, сбрасывая ввод
            }
            else {
                age = scanner.nextInt();
                if(age > MIN_AGE && age < MAX_AGE){
                    break; // в случае,
                           когда возраст удовлетворяет условиям
                           // выходим из цикла
                }
                else {
                    System.out.println("Некорректный диапазон,
                                       попробуйте снова!");
                    age = 0; // устанавливаем age в 0 для повторного ввода
                }
            }
        }

        vasya.setAge(age);
        System.out.println("Возраст Васи - " + vasya.getAge() + " лет.");
    }
}
```

Первым делом прописываем константы, которые означают границы вводимого возраста, для того чтобы в коде не было «магических чисел», так как это является примером плохого тона.

Организовываем бесконечный цикл, который завершится при корректном вводе. Используем метод `hasNextInt()` для проверки имеется ли в консоли целое число, для проверки формата. В случае некорректного ввода выводим

сообщение. Если было введено число, проверяем его диапазон. Если диапазон верен — завершаем цикл, в ином случае «зануляем» и выполнение перейдет в начало цикла.

В конце выводим сообщение о том, что возраст был введен корректно.

4 Запуск приложения

Для того чтобы запустить приложение запустите Main.java нажав на кнопку «Play» в верхней части экрана или через Run→ Run As → Java Application.

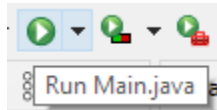


Рисунок 4 — Запуск Main.java

Естественно вы можете реализовать и другие способы ввода, это всего лишь пример для тех, кто совсем не знает с чего начать.