**package** Assignment5;

**import** java.util.Scanner;

**public** **class** WrapperClassConvert {

**public** **static** **void** main(String[] args){

Scanner scan = **new** Scanner(System.***in***);

WrapperClassConvert converter = **new** WrapperClassConvert();

String userInput = scan.nextLine();

Object res;

**try**{

**if** (userInput.matches("-?\\d+")){

res= converter.convertToWrapper(Integer.*parseInt*(userInput));

}**else** **if**(userInput.matches("-?\\d+\\.\\d+")){

res= converter.convertToWrapper(Double.*parseDouble*(userInput));

}**else** **if**(userInput.equalsIgnoreCase("true") || userInput.equalsIgnoreCase("false")){

res= converter.convertToWrapper(Boolean.*parseBoolean*(userInput));

}**else** **if**(userInput.length() == 1){

res= converter.convertToWrapper(userInput.charAt(0));

}**else**{

res="Give proper primitive type as input";

}

}**catch**(NumberFormatException e){

res="Give proper primitive type as input";

}

System.***out***.println(res);

}

**public** Object convertToWrapper(Object input) {

**if**(input == **null**){

**return** "Give proper primitive type as input";

}

**if**(input **instanceof** Integer){

**return** Integer.*valueOf*((**int**) input);

}**else** **if** (input **instanceof** Double){

**return** Double.*valueOf*((**double**) input);

}**else** **if** (input **instanceof** Float){

**return** Float.*valueOf*((**float**) input);

}**else** **if** (input **instanceof** Long){

**return** Long.*valueOf*((**long**) input);

}**else** **if** (input **instanceof** Short){

**return** Short.*valueOf*((**short**) input);

}**else** **if** (input **instanceof** Byte){

**return** Byte.*valueOf*((**byte**) input);

}**else** **if** (input **instanceof** Character){

**return** Character.*valueOf*((**char**) input);

}**else** **if** (input **instanceof** Boolean){

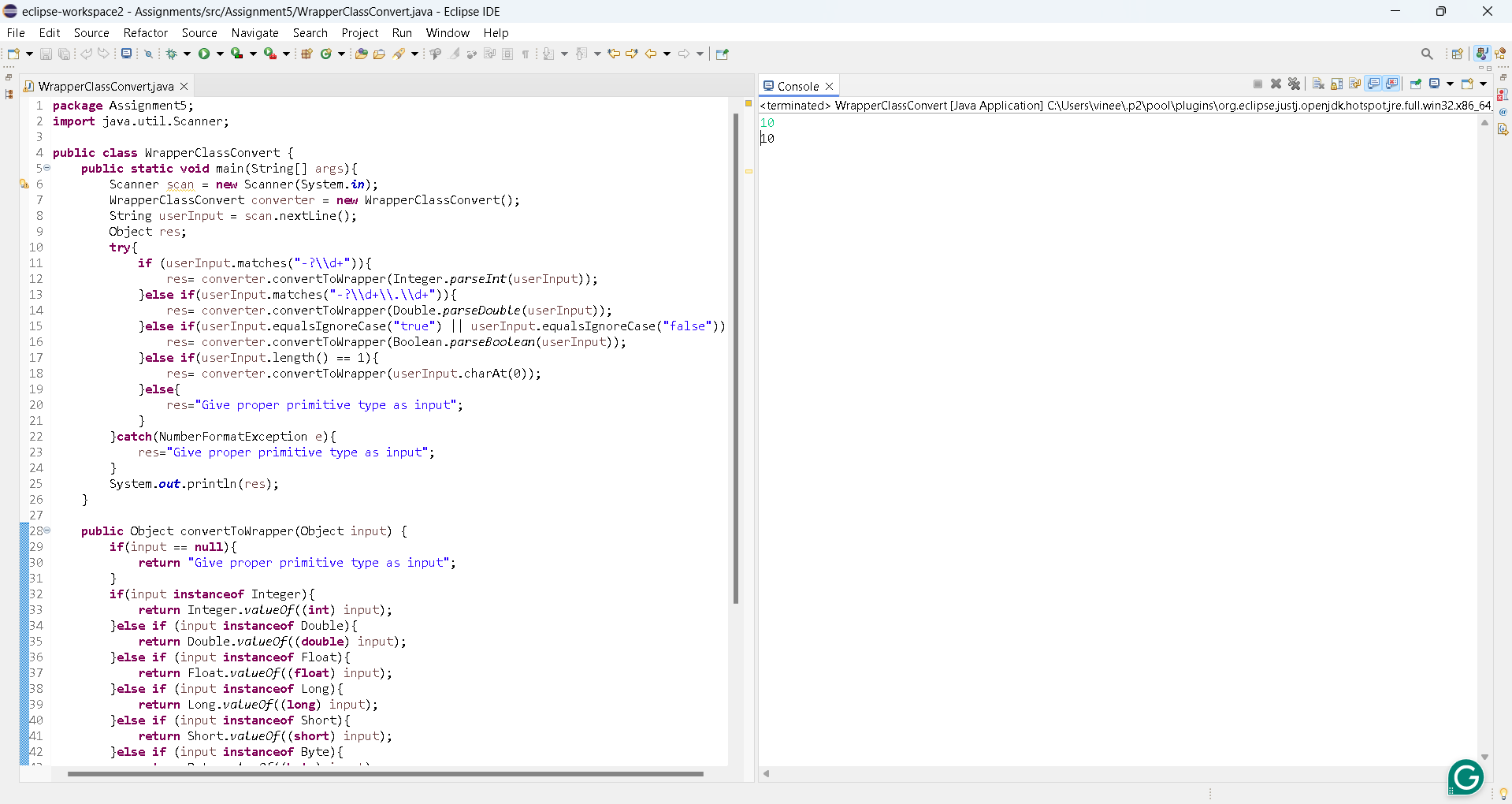
**return** Boolean.*valueOf*((**boolean**) input);

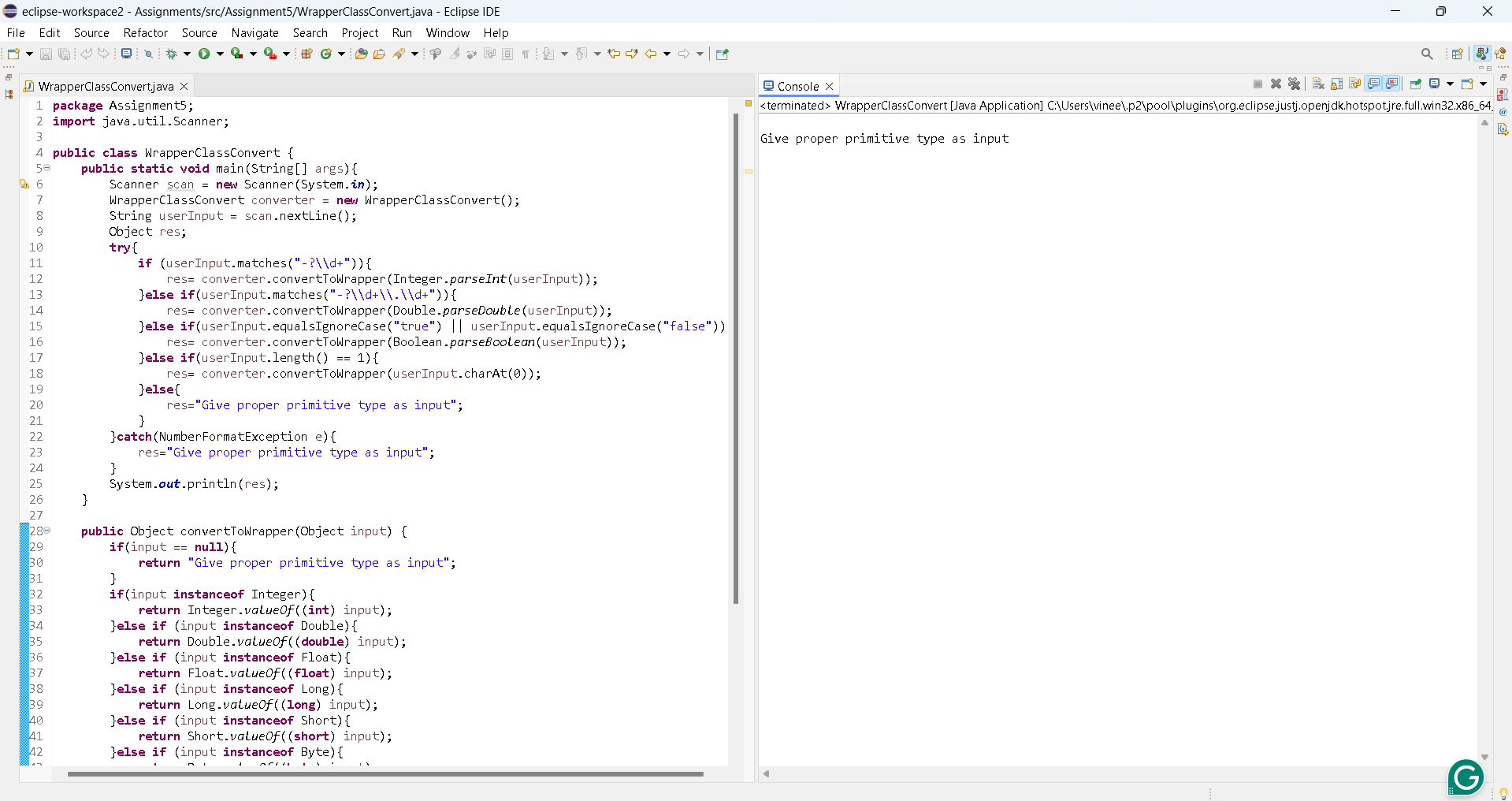
}**else**{

**return** "Give proper primitive type as input";

}

}

}



A screenshot of a computer

Description automatically generated