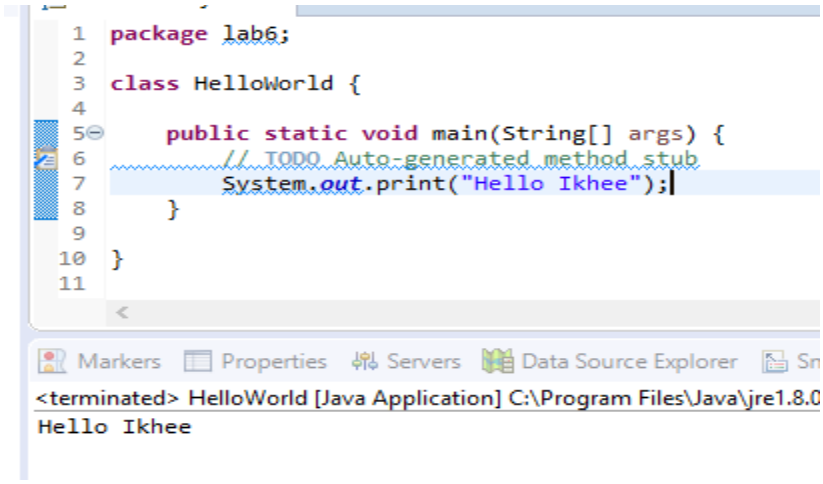
1) Файлын гарчиг өгөх
2)



3)
```
/**
 * HelloWorld class нь дэлгэцэнд
 * "Hello Ikhee!" гаргалтыг хэвлэж харуулна.
 */
class HelloWorld {
	public static void main(String[] args) {
		System.out.print("Hello Ikhee"); // Hello Ikhee String дэлгэцэнд харуулна.
	}

}
```

4)

a) Abcd bacd cbad bcad cabd acbd

b)

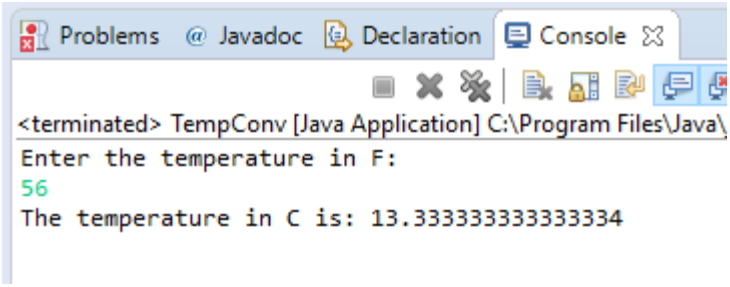
i) Таны сонгосон програмын аргумент юу вэ?
4 hello

ii) Түс аргументыг оруулан ажиллуулахад гарсан үр дүн эсвэл алдааны зурвасыг оруулж хэрхэн ажилсан талаар тайлбарла?
N тоо нь string Уртаас бага байх ёстой. Учир нь хүснэгтэн төрлийн утганд хандахад string уртаас хэтэрсэн утганд хандаж чадахгүй учир зарим үед `java.lang.ArrayIndexOutOfBoundsException` алдаа өгч байсан. Оруулсан string хувьд түүний үсэгнүүдийн байрыг оруулсан n тооноос хамааруулан сольж байрлуулж байсан.

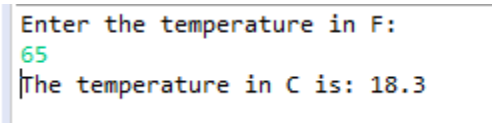
iii) Яагаад энэ аргументыг сонгосныг тайлбарла.
Аргументээр орж ирэх утгаа `permute` функлуу дамжуулах учир эхний параметр нь Integer, дараагий нь string төрөлтэй байх ёстой.

5)

a) Эхний гаралт:

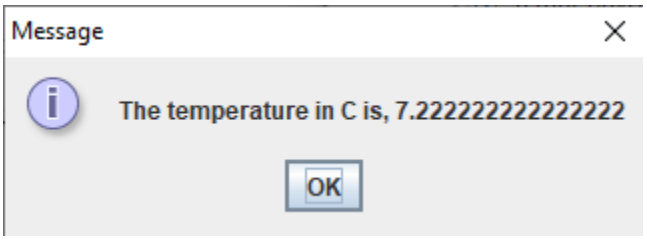


b) Дараагийн гаралт:



6)

```
/**
 * @author ikhbayar.o
 * @version 1.1
 *
 */
public class TempConvGUI {
   public static void main(String[] args) {
          String fahrString;
          double fahr, cel;
          while(true) {
                fahrString = JOptionPane.showInputDialog("Enter the
                temperature in F");
             if(!fahrString.isEmpty()) {
                fahr = Double.parseDouble(fahrString);
                cel = (fahr - 32) * 5.0 / 9.0;
                   JOptionPane.showMessageDialog(null, "The temperature in
                   C is, " + cel);
             }
          }
   }
}
```



7)

10)

```
package hucklebuckle;
```

```
/**
 * <p>
 * Second optional command-line parameter: ngames, the number of games to be
 * played on this run of the program, an integer between 0 and 9 inclusive.
 * Note: if only one command-line parameter is supplied, it is interpreted as
 * gridsize rather than as ngames, as indicated by the nested bracketing 9 on
 * the usage comment below.
 * <p>
 * Usage: hbbv1.1 [gridsize [ngames]]
 */
public class HuckleBuckle {

	public static final String VERSION = "1.1";

	public static void main(String[] args) {

		int gridSize = 5; // gridSize == 5, if there are no command-line args
		int nGames = 1; // 2 дахь аргумент байхгүй уед нэг удаа тоглоно гэдгийг зааж өгнө.
		if (args.length > 0) {
			try {
				gridSize = Integer.parseInt(args[0]);
			} catch (NumberFormatException e) {
				System.err.println("Error: first arg (gridSize) must be an integer.  ");
				System.exit(1);
			}
			if (gridSize < 1 || gridSize > 40) {
				System.err.println("Error: first arg (gridSize) must be in the range 1..40.  ");
				System.exit(1);
			}
		}
		if (args.length > 1) {
			try {
				nGames = Integer.parseInt(args[1]);
			} catch (NumberFormatException e) {
				System.err.println("Error: second arg (nGames) must be an integer. ");
				System.exit(1);
			}
			if (nGames < 0 || nGames > 10) {
				System.err.println("0 - 9 хооронд тоглолтын тоо байх ёстой.");
				System.exit(1);
			}
		}

		if (args.length > 2) {
			System.err.println("Warning: too many args. ");
		}
		Game myGame = new Game(gridSize);

		myGame.play(nGames);
```

```
	}

}
```

11)

```
package hucklebuckle;

class Seeker extends Player {

	private Hider myHider;
	private int distanceMoved; // updated after every change of position

	Seeker(String name, Game game) {
		super(name, game);
		distanceMoved = 0;
	}

	void sayHelloTo(Hider h) {
		myHider = h;
		System.out.println(getName() + " says, \"Hi, " + h.getName() + ", I'm "
+ getName() + ". Glad to meet you!\"");

	}

	private void reportLocation() {
		System.out.println(getName() + " says, \"I'm at " + getX() + ", " +
getY() + ".\"");
	}

	/**
	 * Moves this seeker to (px, py).
	 *
	 * 12
	 *
	 * @param px the x-coord of the next position of this seeker
	 * @param py the y-coord of the next position of this seeker
	 *
	 *		If the distance moved is non-zero, the seeker reports their new
	 *		location.
	 *
	 *		Constraint: this method should use Player.distance()
	 *
	 */
	void moveTo(int px, int py) {
		if (distance(px, py) > 0.0) {
			distanceMoved += (int) distance(px, py);
			// To do: confirm that truncation is appropriate here, e.g.
			// that a Seeker who moves diagonally should count "one step"
			// for distances in the range [1.0, 2.0).
			setX(px);
			setY(py);
			reportLocation();
		}
	}
```

```
	void seek() {
		reportLocation(); // I report my initial location
		boolean moveRight = true; // I move left-to-right on even-numbered rows
		for (int y = 0; y < getGame().getGridSize(); y++) {
			for (int cx = 0; cx < getGame().getGridSize(); cx++) {
				int x = (moveRight ? cx : getGame().getGridSize() - cx -
1);
				moveTo(x, y);
				if (myHider.revealTemperature(this) == 0) { // Did I find
it?
					System.out.println(getName() + " says, \"That was
fun! I walked " + distanceMoved
							+ " steps before I found it.\"");
					return; // I stop when I find the hidden object
				}
			}
			moveRight = !moveRight;
		}
		System.out.println(getName() + " says, \"I'm giving up.  I took " +
distanceMoved + "steps before quitting.\"");
	}

}
```