**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего**

**образование**

**Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

**Кафедра «МКиИТ»**

**дисциплина «Объектно-ориентированное программирование»**

### [ПРАКТИЧЕСКИЕ ЗАНЯТИЯ](https://lms.mtuci.ru/lms/course/view.php?id=416#section-2)

## Задачи для практических занятий. Уровень сложности: 4/6

Подготовила студентка

группы БВТ1901: Нкурикийе Х

Проверил: Мосева М.С.

2021

import java.util.\*;

import java.lang.\*;

public class task04 {

public static int isSpace(String a, String b) { // 4.1

String withSpaces = a + " " + b;

int count = 0;

for (int i = 0; i < withSpaces.length(); i++) {

char c = withSpaces.charAt(i);

if (c == ' ') {

count++;

}

}

return withSpaces.length() - count;

}

public static void essay(int n, int k, String str) {

String[] arrStr = str.split(" ");

for (int i = 1; i < n; i++) {

if (isSpace(arrStr[i-1], arrStr[i]) <= k) {

arrStr[i] = arrStr[i-1] + " " + arrStr[i];

arrStr[i-1] = "";

}

}

for (int i = 0; i < n; i++) {

if (arrStr[i].length() > 0) {

System.out.println(arrStr[i]);

}

}

}

public static String[] splitBkt(String a) { // 4.2

boolean checkBkt = true;

char[] aArray = a.toCharArray();

String[] resArray = new String[0];

for (char i : aArray) {

if (!(i == (int) '(' || i == (int) ')')) {

checkBkt = false;

break;

}

}

if (checkBkt) {

int count0 = 0, count1 = 0, preI = 0;

for (int i = 0; i < aArray.length-1; i++) {

if (aArray[i] == '(') count0++;

if (aArray[i] == ')') count1++;

if (count0 == count1) {

resArray = Arrays.copyOf(resArray, resArray.length + 1);

resArray[resArray.length - 1] = a.substring(preI, i+1);

preI = i + 1;

}

}

resArray = Arrays.copyOf(resArray, resArray.length + 1);

resArray[resArray.length - 1] = a.substring(preI);

}

return resArray;

}

public static String toCamelCase(String a) { // 4.3

while (a.contains("\_")) {

int find = a.indexOf("\_");

if (find == 0) {

a = a.substring(1);

}

else if (find < a.length() - 2) {

a = a.substring(0, find) + a.substring(find+1, find + 2).toUpperCase() + a.substring(find + 2);

}

else if (find < a.length() - 1) {

a = a.substring(0, find) + a.substring(find+1, find + 2).toUpperCase();

}

else if (find < a.length()) {

a = a.substring(0, find);

}

}

return a;

}

public static String toSnakeCase(String a) {

for (int i = 0; i < a.length(); i++) {

if (a.charAt(i) >= 'A' && a.charAt(i) <= 'Z') {

if (i == 0) {

a = a.substring(0, 1).toLowerCase() + a.substring(1);

}

else if (i < a.length() - 1) {

a = a.substring(0, i) + "\_" + a.substring(i, i + 1).toLowerCase() + a.substring(i + 1);

}

else {

a = a.substring(0, i) + "\_" + a.substring(i).toLowerCase();

}

}

}

return a;

}

public static String overTime(float[] array) { // 4.4

float overtimeMorning = 0.0f, overtime = 0.0f;

if (array[0] < 9) {

overtimeMorning = 9.0f - array[0];

}

if (array[1] > 17) {

overtime = (array[1] - 17.0f) + overtimeMorning;

}

float result = (array[1] - array[0] - overtime) \* array[2] + overtime \* array[2] \* array[3];

String res = String.valueOf(result);

res += "00";

res = res.substring(0, res.indexOf(".") + 3);

return "$" + res;

}

public static String BMI(String a, String b) { // 4.5

a = a.trim();

b = b.trim();

int indexA = a.indexOf(' ');

int indexB = b.indexOf(' ');

String weightStr = a.substring(0, indexA);

String weightStep = a.substring(indexA + 1);

String heightStr = b.substring(0, indexB);

String heightStep = b.substring(indexB + 1);

float weight = Float.parseFloat(weightStr);

float height = Float.parseFloat(heightStr);

if (weightStep.contains("pounds")) {

weight /= (2.205f);

}

if (heightStep.contains("inches")) {

height /= (39.37f);

}

if (!(weightStep.contains("kilos") || weightStep.contains("pounds"))) {

return "Некорректный ввод";

}

else if (!(heightStep.contains("meters") || heightStep.contains("inches"))) {

return "Некорректный ввод";

}

else {

String resConclusion;

double bmi = weight / (height \* height);

if (bmi < 18.5) {

resConclusion = " Недостаточный вес";

}

else if (bmi < 25) {

resConclusion = " Нормальный вес";

}

else {

resConclusion = " Избыточный вес";

}

String res = String.valueOf(bmi);

res += "00";

res = res.substring(0, res.indexOf(".") + 2);

return res + resConclusion;

}

}

public static int bugger(int a) { // 4.6

int count = 0;

while (a >= 10) {

int mlt = 1, mod;

while (a != 0) {

mod = a % 10;

mlt \*= mod;

a = a / 10;

}

a = mlt;

count++;

}

return count;

}

public static String toStarShorthand(String a) { // 4.7

int i = 1, count = 1;

String res = "";

while (i < a.length()) {

if (a.charAt(i) == a.charAt(i - 1)) {

count++;

if (a.length() - 1 == i) {

res += a.charAt(i) + "\*" + count;

break;

}

else

i++;

}

else {

if (count == 1) {

res += a.charAt(i - 1);

if (i == a.length() - 1) {

res += a.charAt(i);

}

}

else {

res += a.charAt(i - 1) + "\*" + count;

}

count = 1;

i++;

}

}

return res;

}

public static boolean doesRhyme(String s, String s2) { // 4.8

String c = "";

String k = "";

int a = s.lastIndexOf(" ");

int b = s2.lastIndexOf(" ");

String subs = s.substring(a);

String subs2 = s2.substring(b);

for (char l:subs.toCharArray()) {

if ((l=='a')||(l=='e')||(l=='i')||(l=='o')||(l=='u')||(l=='y')||(l=='A')||(l=='E')||(l=='I')||(l=='O')||(l=='U')||(l=='Y')){

c += l;

}

}

for (char l:subs2.toCharArray()){

if ((l=='a')||(l=='e')||(l=='i')||(l=='o')||(l=='u')||(l=='y')||(l=='A')||(l=='E')||(l=='I')||(l=='O')||(l=='U')||(l=='Y')){

k += l;

}

}

return c.equalsIgnoreCase(k);

}

public static boolean trouble(String a, String b) { // 4.9

char[] aChar = a.toCharArray();

char[] bChar = b.toCharArray();

boolean result = false;

int[] num1 = new int[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

int[] num2 = new int[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

for (int i = 1; i < aChar.length; i++) {

if (aChar[i] == aChar[i-1]) {

num1[Character.getNumericValue(aChar[i])] += 1;

}

}

for (int i = 1; i < bChar.length; i++) {

if (bChar[i] == bChar[i-1]) {

num2[Character.getNumericValue(bChar[i])] += 1;

}

}

for (int i = 0; i < 10; i++) {

if (num1[i] == 2 && num2[i] == 1) {

result = true;

break;

}

}

return result;

}

public static int countUniqueBooks(String a, String b) { // 4.10

if (b.length() == 1) {

String partA = a;

String[] parts = new String[0];

while (partA.length() != 0) {

int beginPart = partA.indexOf(b);

if (beginPart == -1) {

break;

}

partA = partA.substring(beginPart + 1);

int endPart = partA.indexOf(b);

parts = Arrays.copyOf(parts, parts.length + 1);

parts[parts.length - 1] = partA.substring(0, endPart);

partA = partA.substring(endPart + 1);

}

String resStr = "";

for (String i: parts) {

resStr += i;

}

String lowerRes = resStr.toLowerCase();

boolean[] isItThere = new boolean[Character.MAX\_VALUE];

for (int i = 0; i < lowerRes.length(); i++) {

isItThere[lowerRes.charAt(i)] = true;

}

int count = 0;

for (boolean value: isItThere) {

if (value) {

count++;

}

}

return count;

}

else {

return -1;

}

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in).useLocale(Locale.ENGLISH);

Scanner sc = new Scanner(System.in);

System.out.println("Введите номер задачи (от 31 до 40):");

int n = in.nextInt();

switch (n) {

case 31 -> {

System.out.println("essay()");

System.out.println("Введите n:");

int n1 = in.nextInt();

System.out.println("Введите k:");

int k1 = in.nextInt();

System.out.println("Введите строку:");

String c1 = sc.nextLine();

essay(n1, k1, c1);

}

case 32 -> {

System.out.println("split()");

System.out.println("Введите строку:");

String a2 = sc.nextLine();

String[] b2 = splitBkt(a2);

System.out.println("Результат:");

for (String s : b2) {

System.out.println(s);

}

}

case 33 -> {

System.out.println("toCamelCase() - 1");

System.out.println("toSnakeCase() - 2");

int n3 = in.nextInt();

if (n3 == 1) {

System.out.println("toCamelCase()");

String a3 = sc.nextLine();

System.out.println("Результат: " + toCamelCase(a3));

}

else if (n3 == 2) {

System.out.println("toSnakeCase()");

String a3 = sc.nextLine();

System.out.println("Результат: " + toSnakeCase(a3));

}

}

case 34 -> {

System.out.println("overTime()");

float[] a4 = new float[4];

System.out.println("Введите начало рабочего дня:");

a4[0] = in.nextFloat();

System.out.println("Введите конец рабочего дня:");

a4[1] = in.nextFloat();

System.out.println("Введите почасовую ставку:");

a4[2] = in.nextFloat();

System.out.println("Введите множитель сверхурочных работ:");

a4[3] = in.nextFloat();

System.out.println("Результат: " + overTime(a4));

}

case 35 -> {

System.out.println("BMI()");

System.out.println("Введите вес:");

String a5 = sc.nextLine();

System.out.println("Введите рост:");

String b5 = sc.nextLine();

System.out.println("Результат: " + BMI(a5, b5));

}

case 36 -> {

System.out.println("bugger()");

int a6 = in.nextInt();

System.out.println("Результат: " + bugger(a6));

}

case 37 -> {

System.out.println("toStarShorthand()");

String a7 = sc.nextLine();

System.out.println("Результат: " + toStarShorthand(a7));

}

case 38 -> {

System.out.println("doesRhyme()");

System.out.println("Первая строка:");

String a8 = sc.nextLine();

System.out.println("Вторая строка:");

String b8 = sc.nextLine();

System.out.println("Результат: " + doesRhyme(a8, b8));

}

case 39 -> {

System.out.println("trouble()");

System.out.println("Введите первое число:");

String a9 = sc.nextLine();

System.out.println("Введите второе число:");

String b9 = sc.nextLine();

System.out.println("Результат: " + trouble(a9, b9));

}

case 40 -> {

System.out.println("countUniqueBooks()");

System.out.println("Введите строку:");

String a10 = sc.nextLine();

System.out.println("Введите символ:");

String b10 = sc.nextLine();

System.out.println("Результат: " + countUniqueBooks(a10, b10));

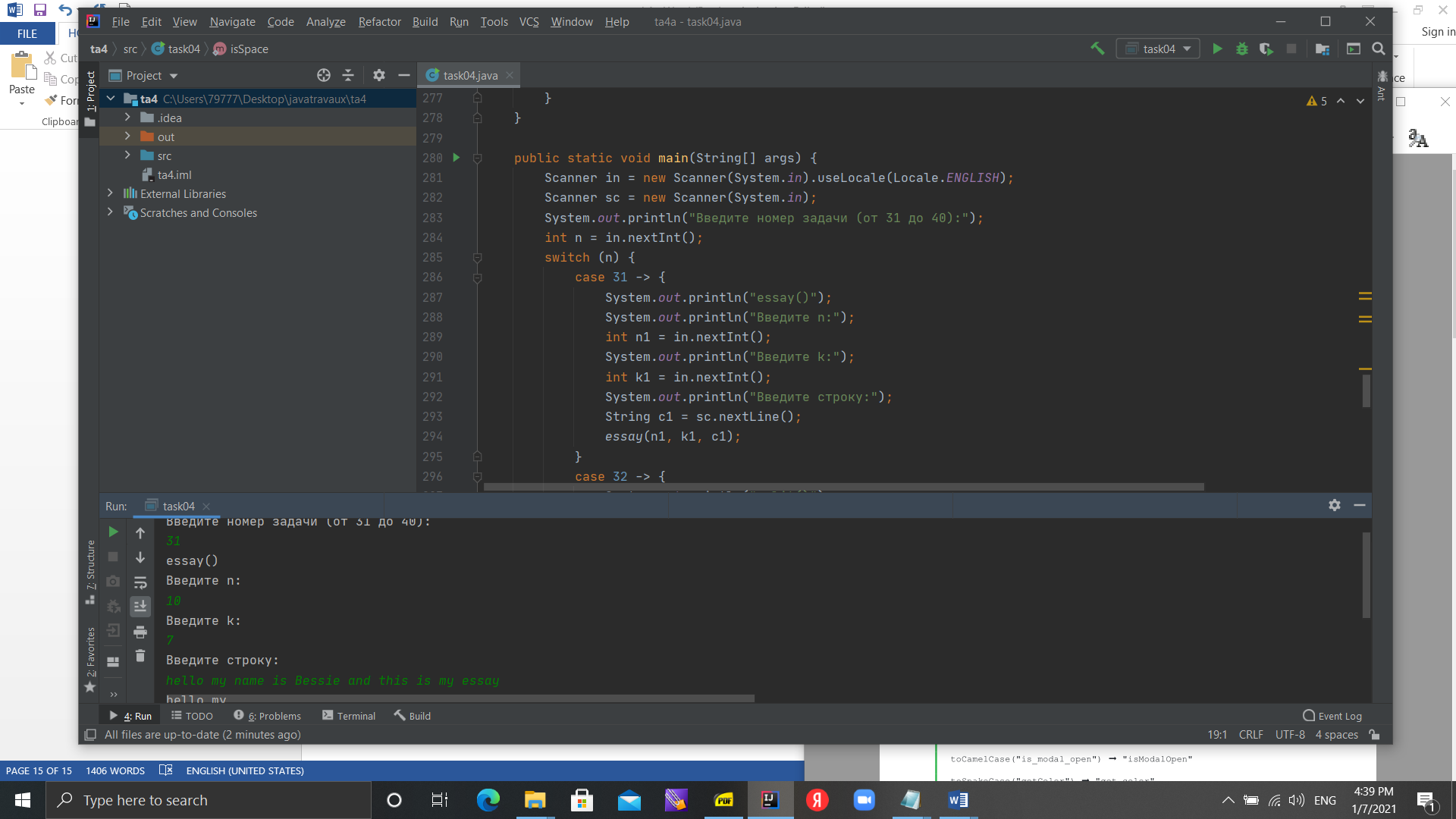
}

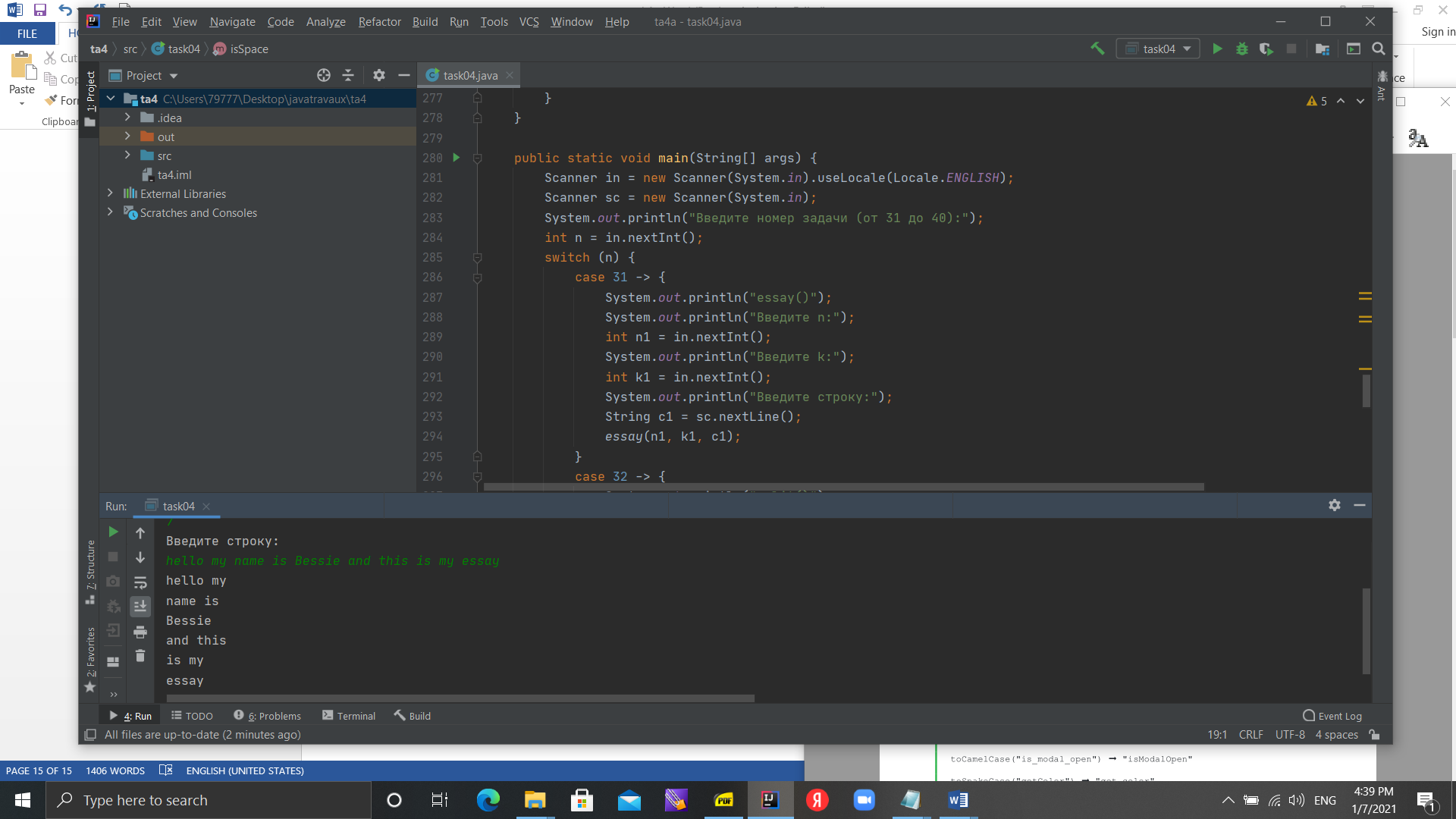
}

}

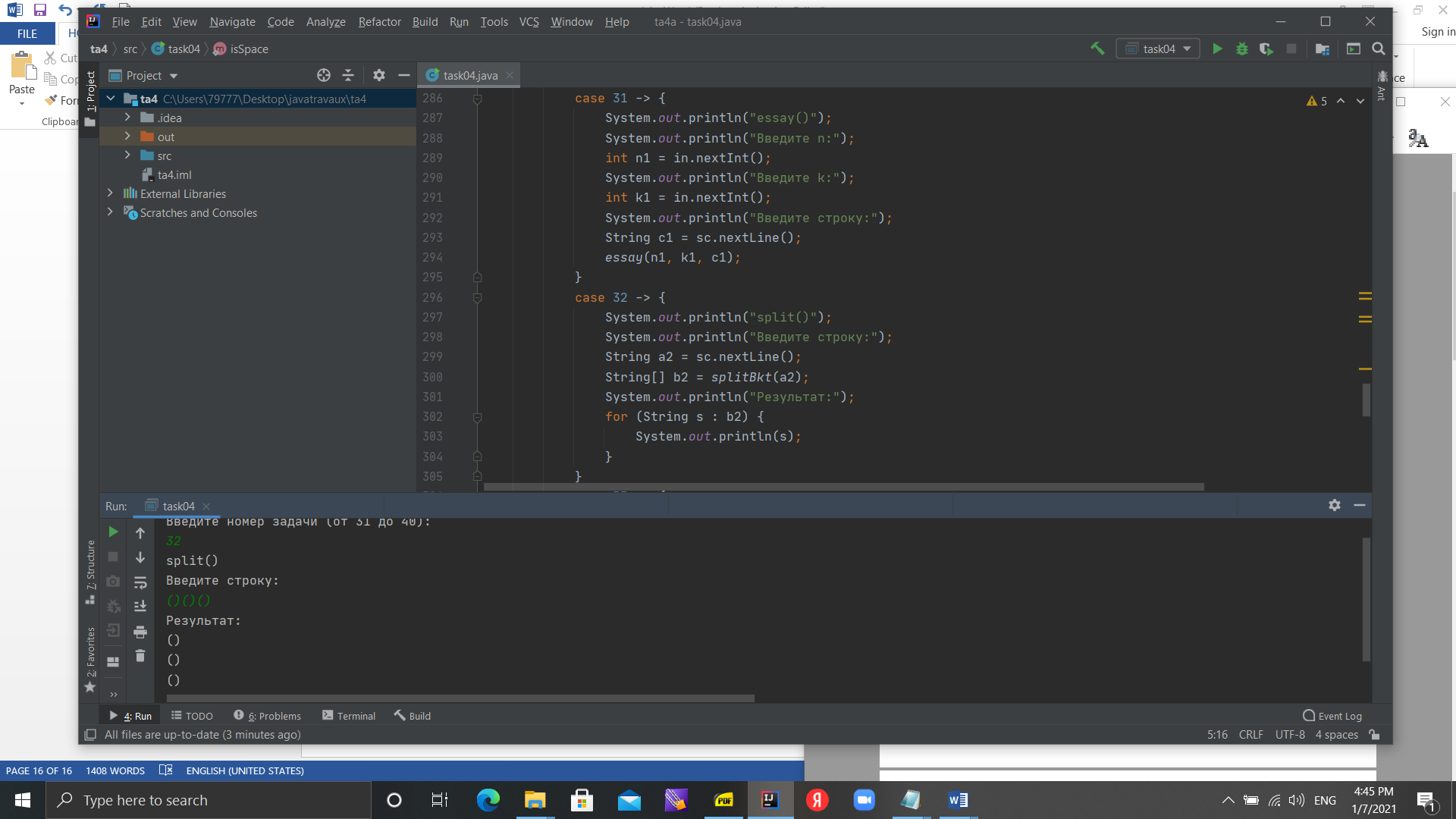
}

4.1

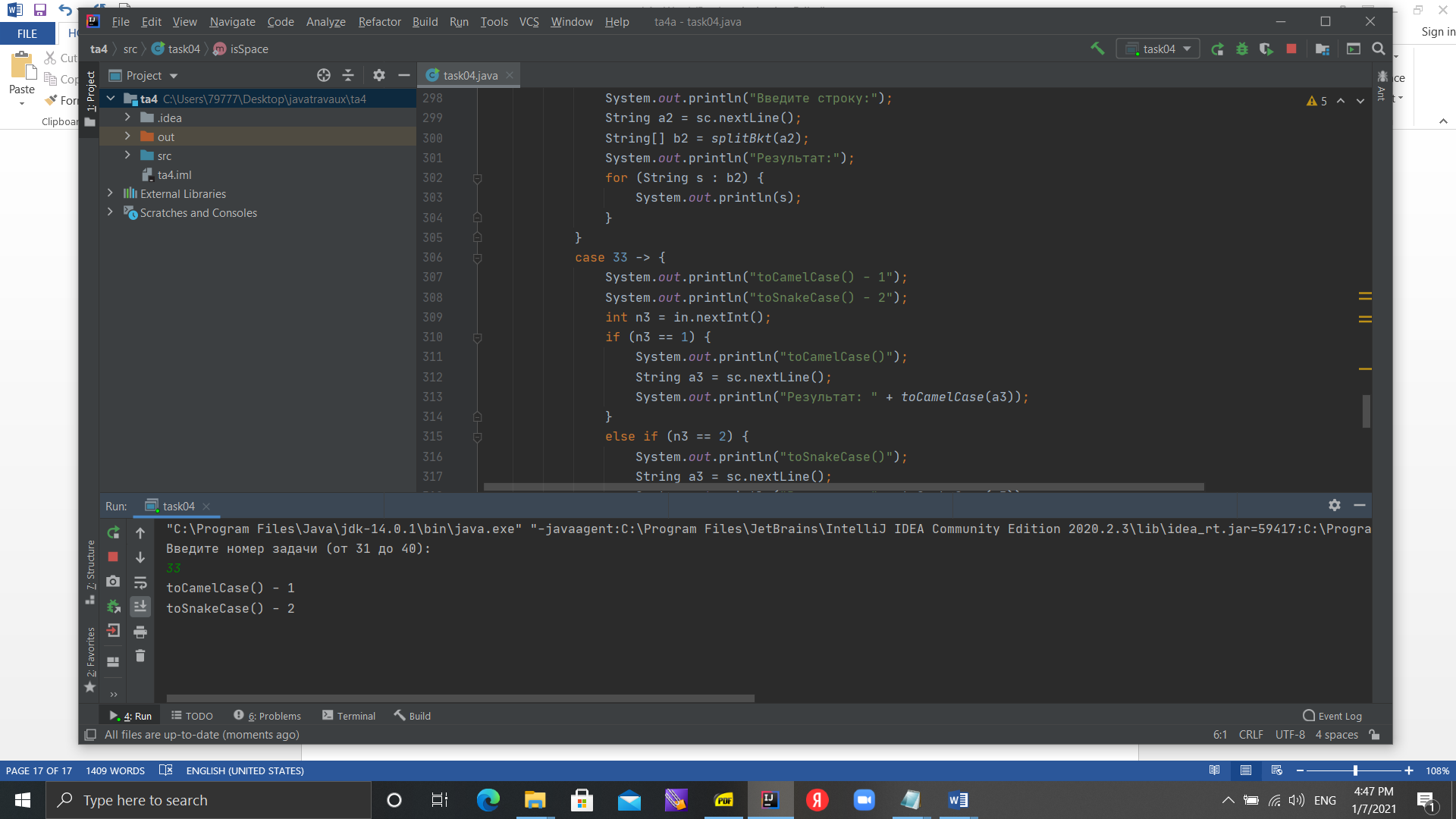




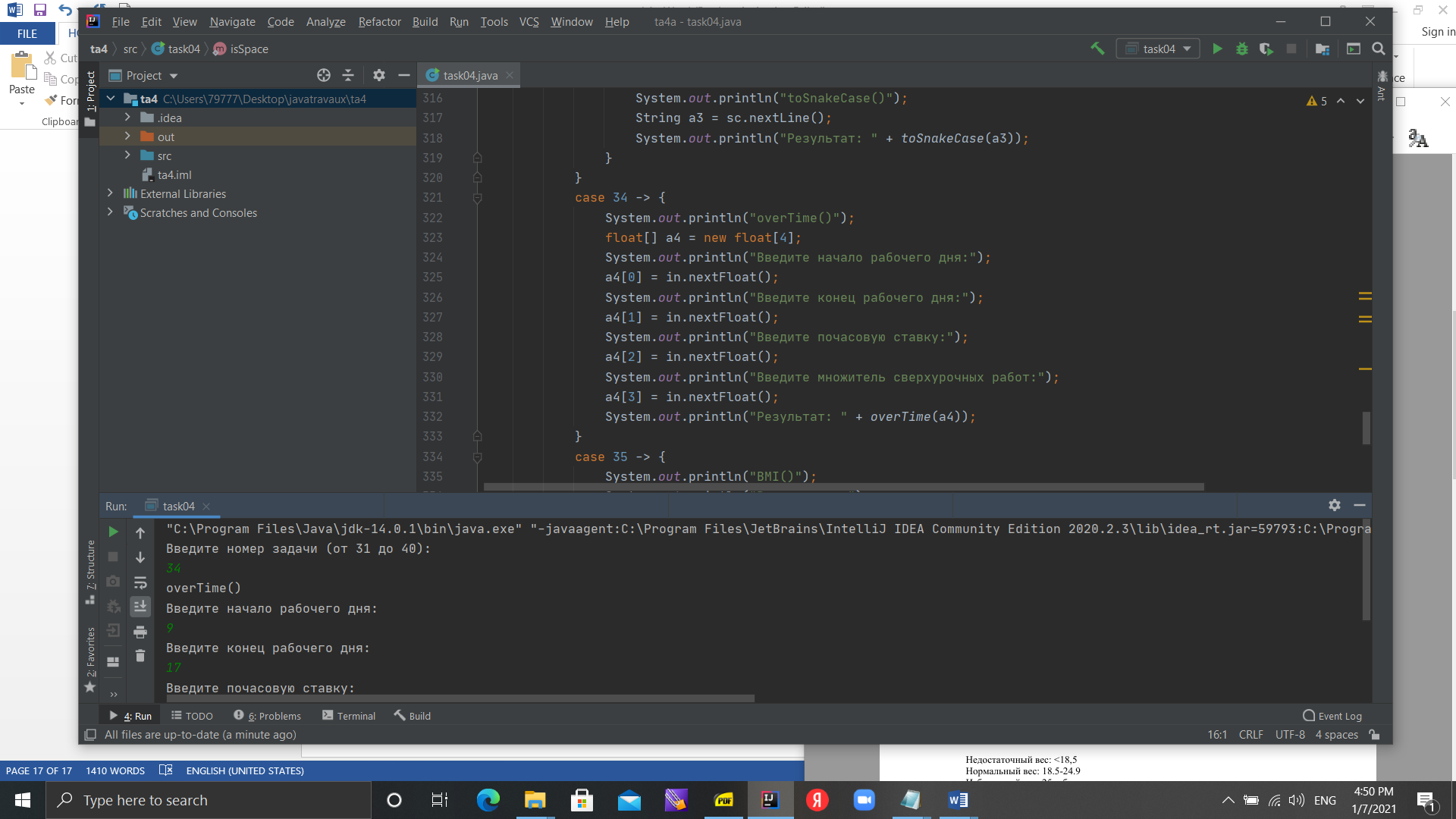
32//4.2

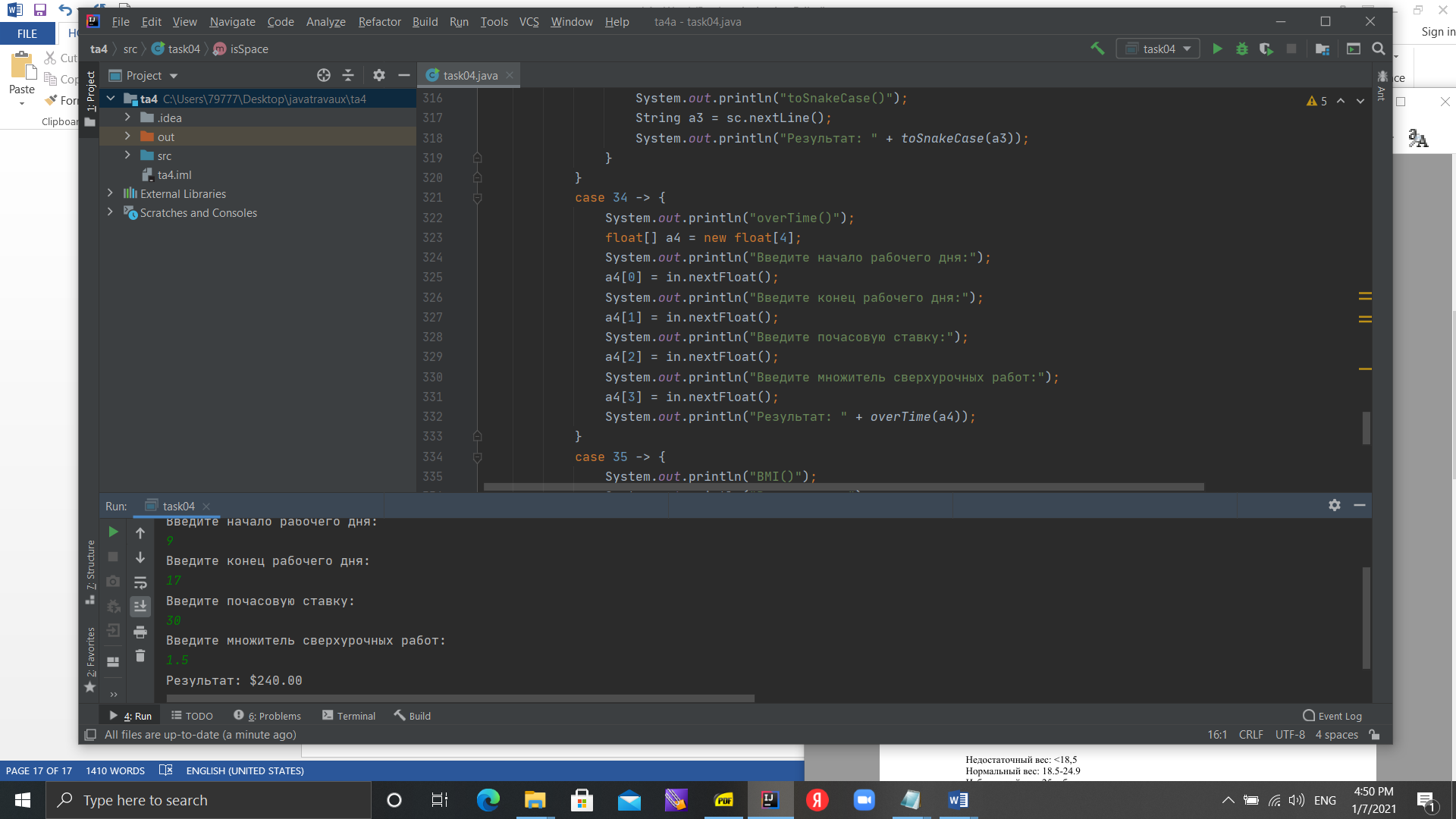


33//4.3

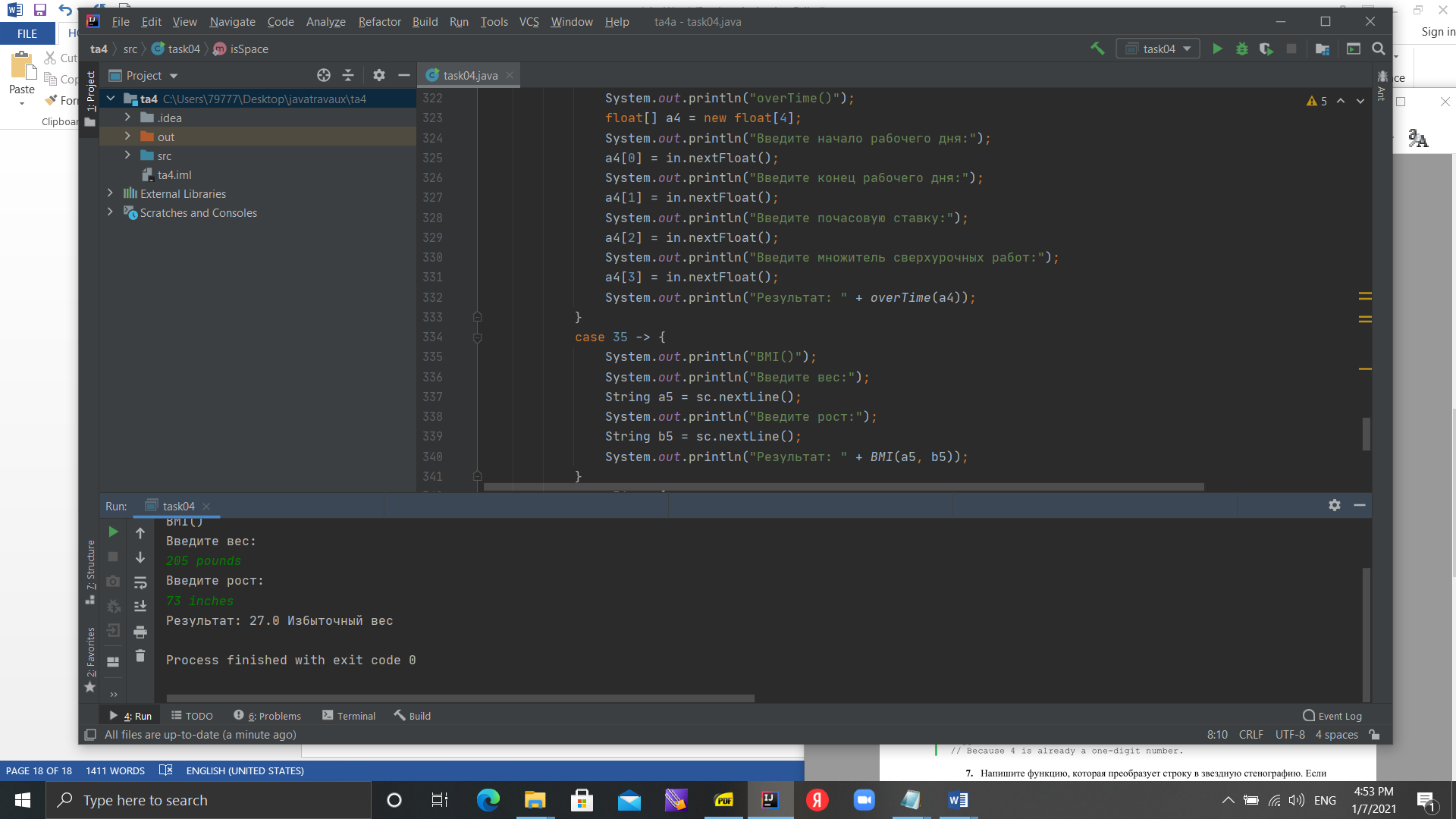


34//4.4

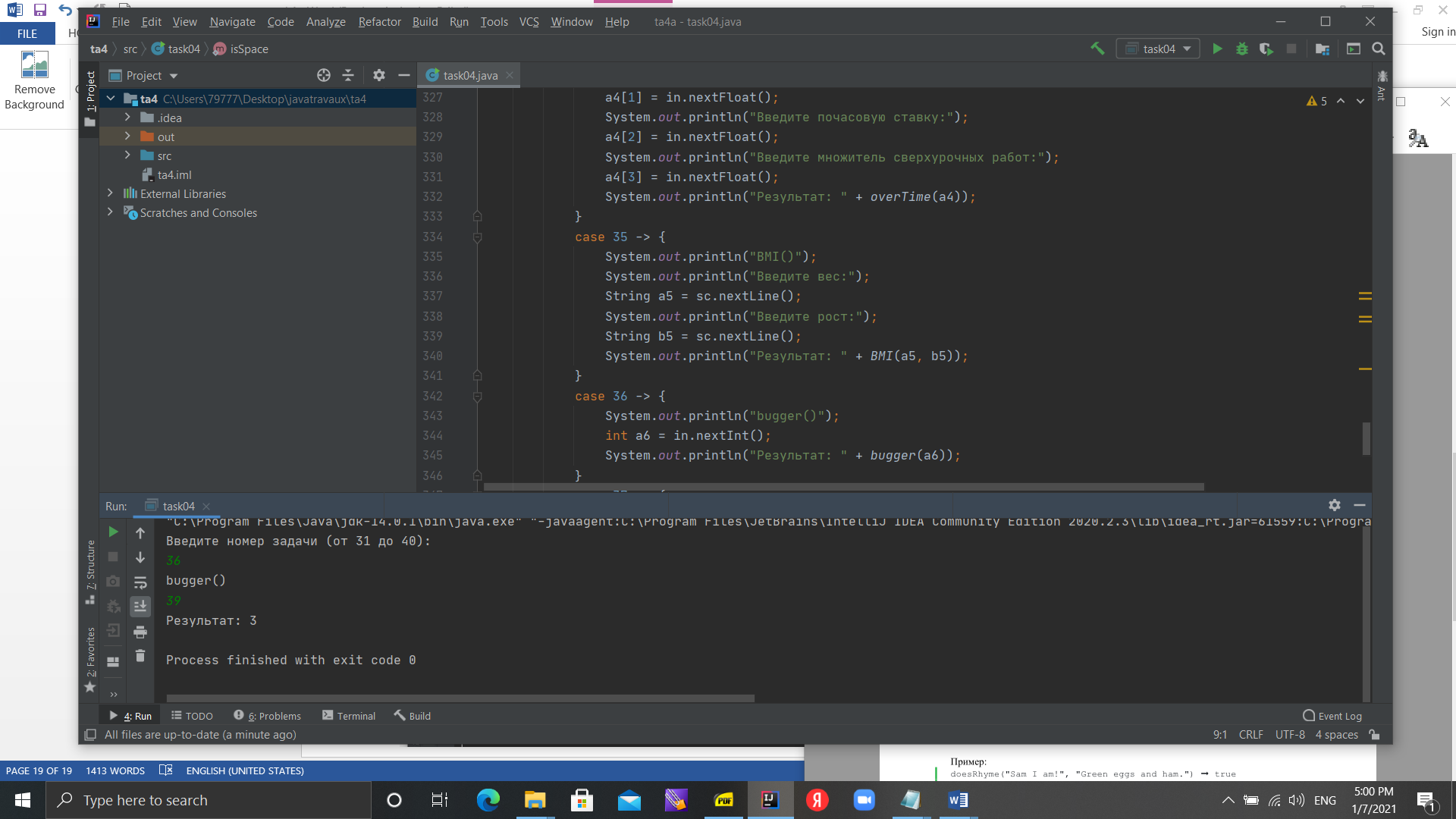




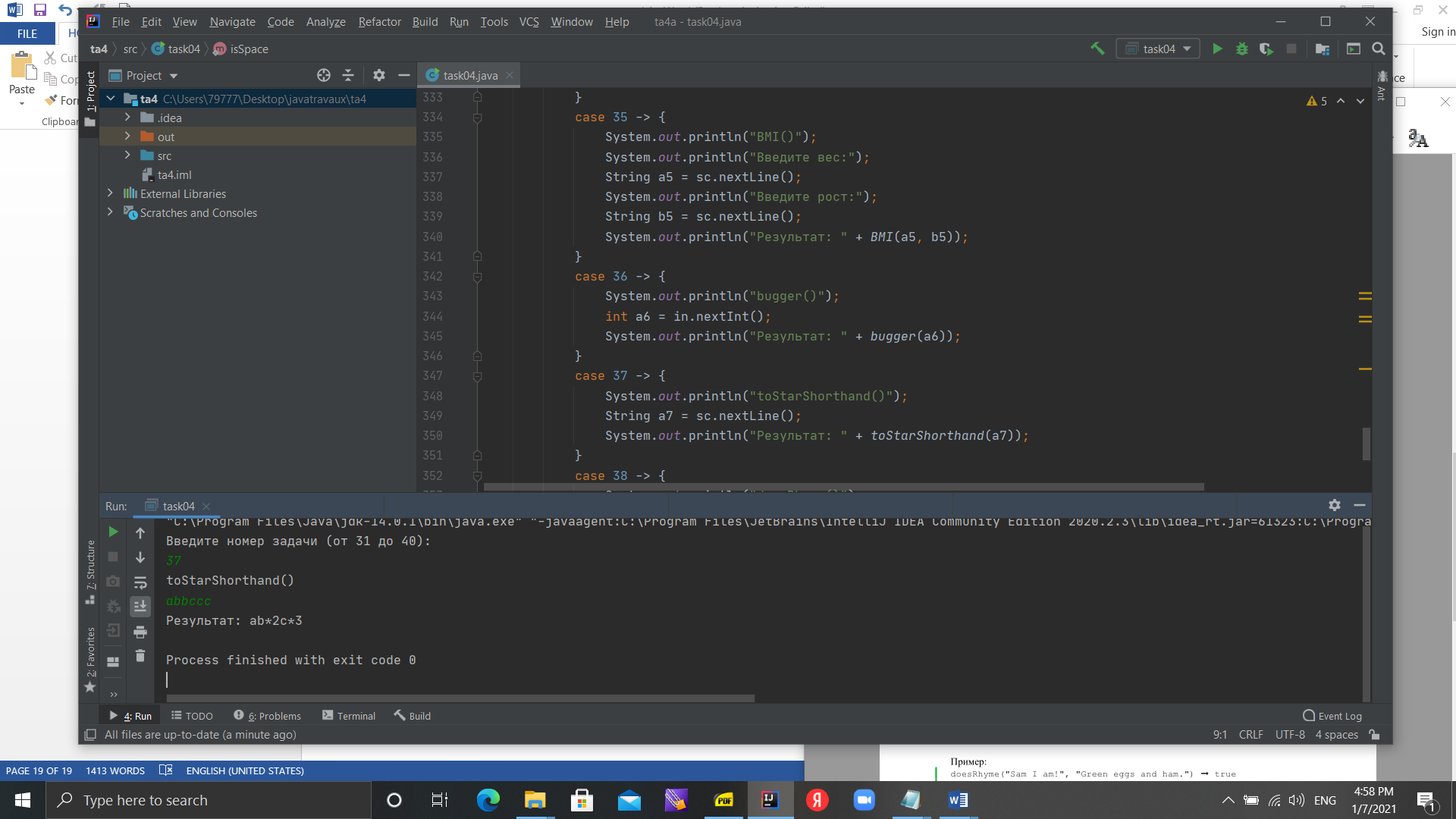
35//4.5



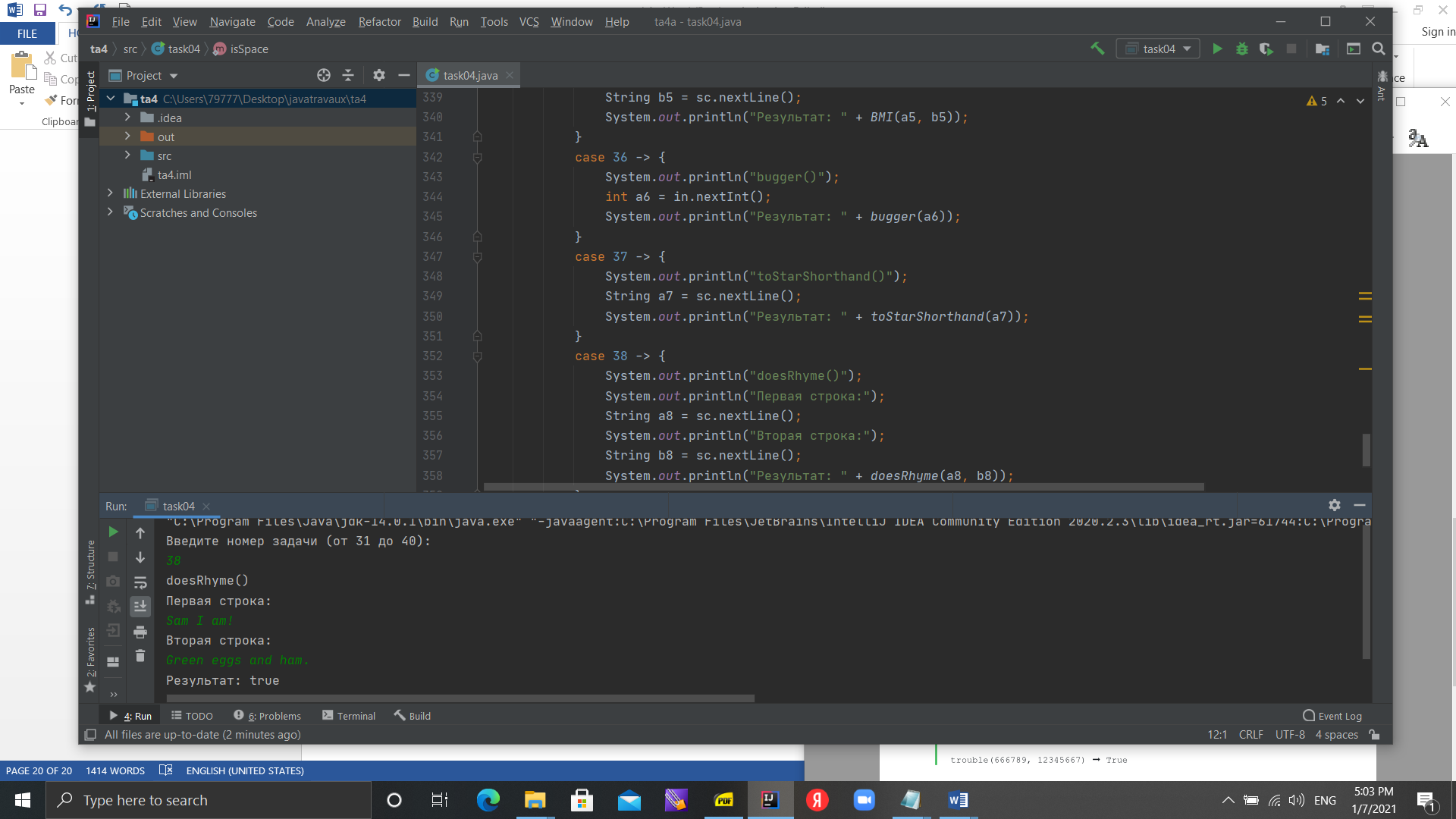
36//4.6



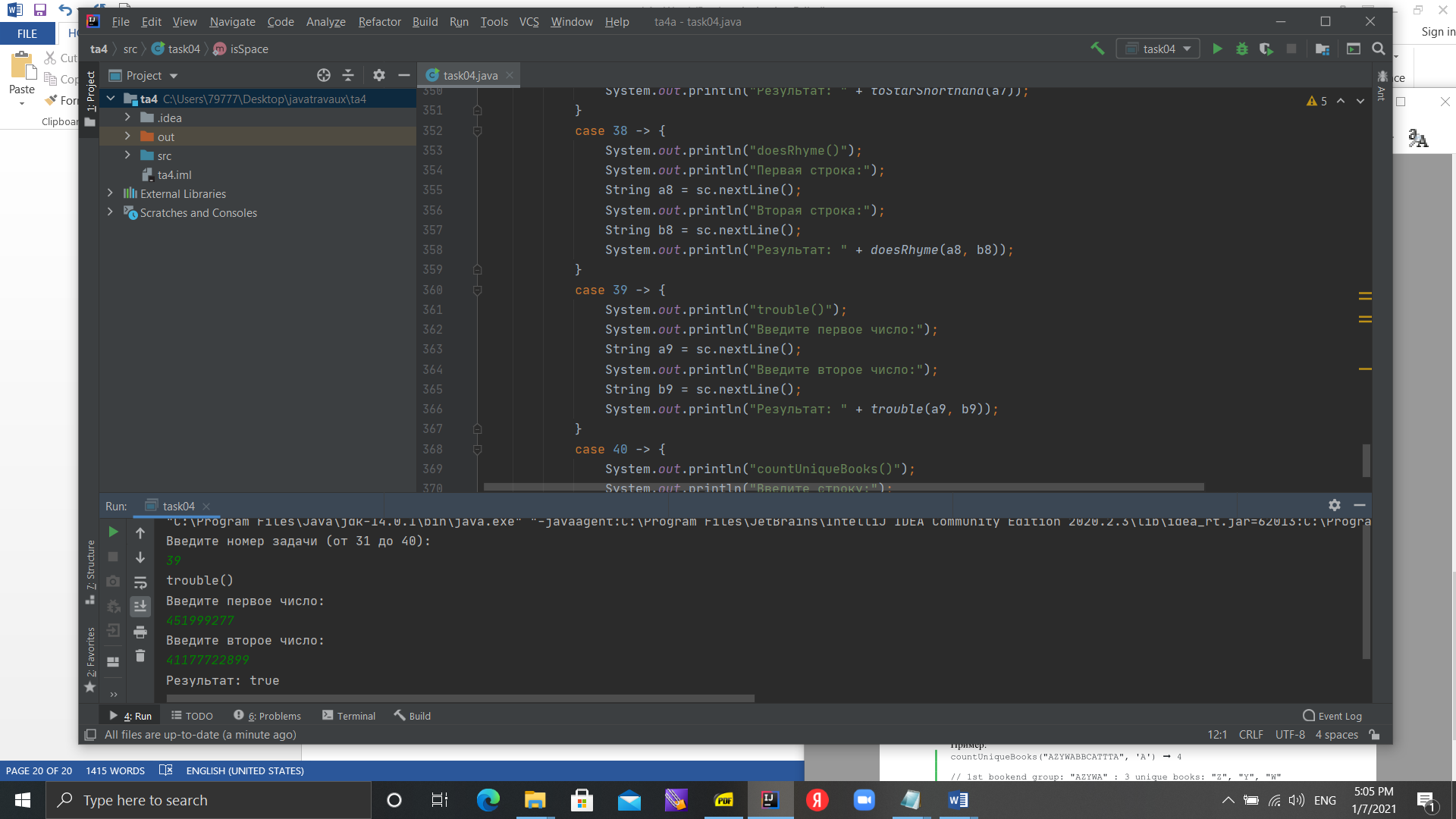
37//4.7



38//4.8



39//4.9



40//4.10

