ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт – Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

Санкт-Петербургский колледж телекоммуникаций

**Практическое занятие № 2**

**Примитивные типы данных и их операции**

МДК.03.02. Инструментальные средства

разработки программного обеспечения К581

Студент гр. 581\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Лепов

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. И. Баталов

Санкт-Петербург

2021 год

1. Постановка задачи.

* В данной практической работе нужно выполнить фрагменты программного кода, приведенного тексте «Практическое занятие № 2».
* Также выполнить фрагменты программного кода приведенного в тексте лекционного занятия № 2. Типы данных и операции.
* Посмотреть на результаты компиляции.
* Также нужно проверить свои навыки загрузки кода программ на хостинге проектов программного обеспечения github.com

1. Исходный код.

**Пример соответствия символа числу:**

public static void main(String[] args) {

char symb1=1067;

char symb2 ='Ы';

System.out.println("symb1 contains "+ symb1);

System.out.println("symb2 contains "+ symb2);

}

**Пример соответствия символа числу:**

public static void main(String[] args) {

char ch = 'J';

int intCh = (int) ch;

System.out.println("J corresponds with "+ intCh);

}

**Пример работы со строками:**

public static void main(String[] args) {

String s1 = "I have ";

String s2 = " apples ";

int num = 3;

String s = s1 + num + s2;

System.out.println(s);

}

**Пример использования умножения с разной битовой точностью и попытка деления. Но, так как переполнение происходит только в случае умножения, а присвоенное значение входит в диапазон int, то вывод при делении на ноль оборачивается ошибкой:**

package com.company;

public class Main {

public static void main(String[] args) {

int i=300000;

System.out.println(i\*i); // умножение с точностью 32 бита

long m=i;

System.out.println(m\*m); // умножение с точностью 64 бита

System.out.println(1/(m-i));

}

}

**Пример преобразования типов и переполнения стека:**

package com.company;

public class Main {

public static void main(String[] args) {

byte x=-128;

System.out.println(-x);

byte y=127;

System.out.println(++y);

}

}

**Пример влияния оператора на тип данных:**

package com.company;

public class Main {

public static void main(String[] args) {

char c='A';

System.out.println(c);

System.out.println(c+1);

System.out.println("c="+c);

System.out.println('c'+'='+c);

}

}

**Пример Overflow:**

package com.company;

public class Main {

public static void main(String[] args) {

System.out.println(1e20f\*1e20f);

System.out.println(-1e200\*1e200);

}

}

**Пример Underflow:**

package com.company;

public class Main {

public static void main(String[] args) {

System.out.println(1e-40f/1e10f); // underflow для float

System.out.println(-1e-300/1e100); // underflow для double

float f=1e-6f;

System.out.println(f);

f+=0.002f;

System.out.println(f);

f+=3;

System.out.println(f);

f+=4000;

System.out.println(f);

}

}

**Пример констант:**

package com.company;

public class Main {

public static void main(String[] args) {

String s1 = "abc";

String s2 = "abc";

String s3 = "a"+"bc";

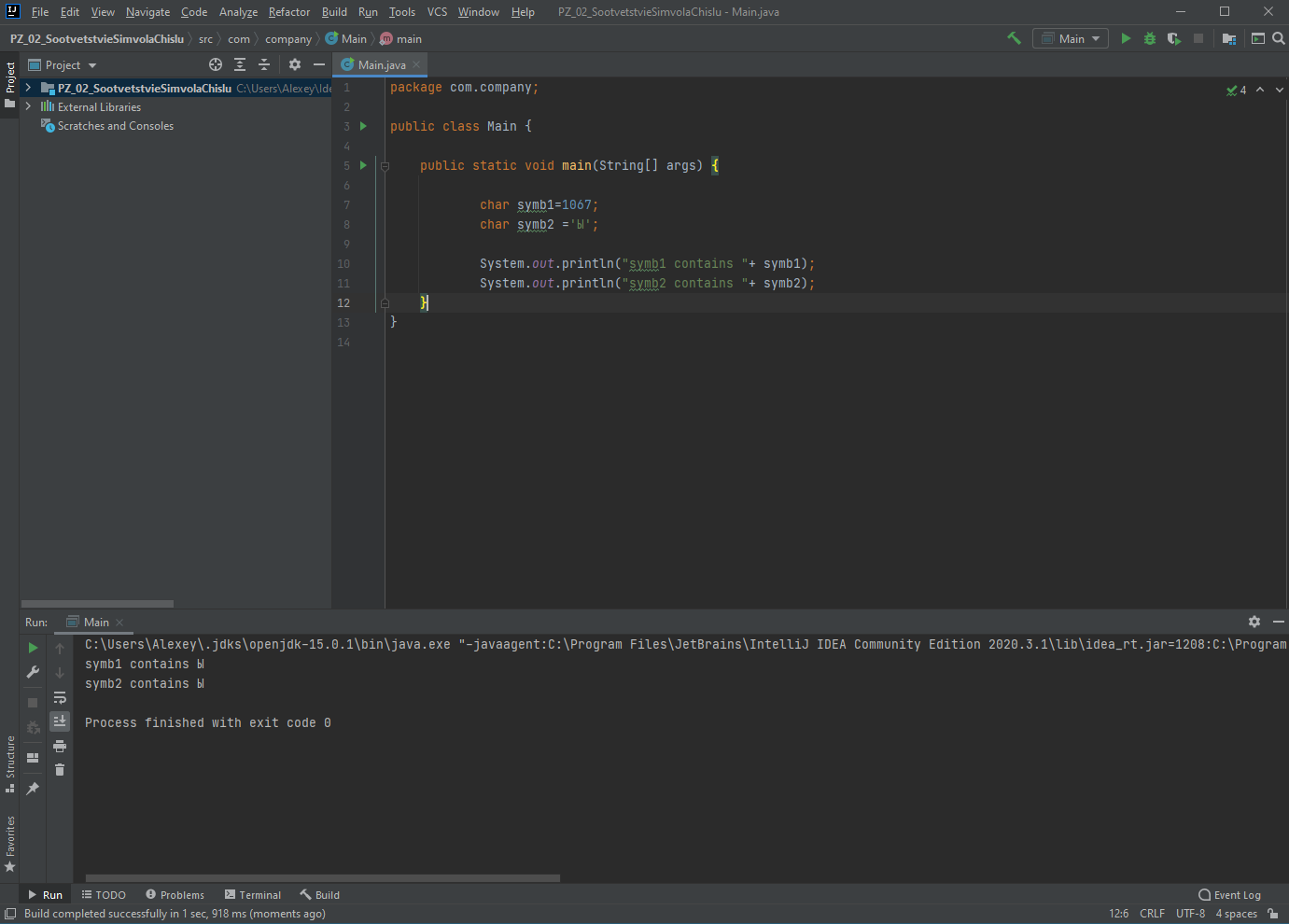
System.out.println(s1==s2);

System.out.println(s1==s3);

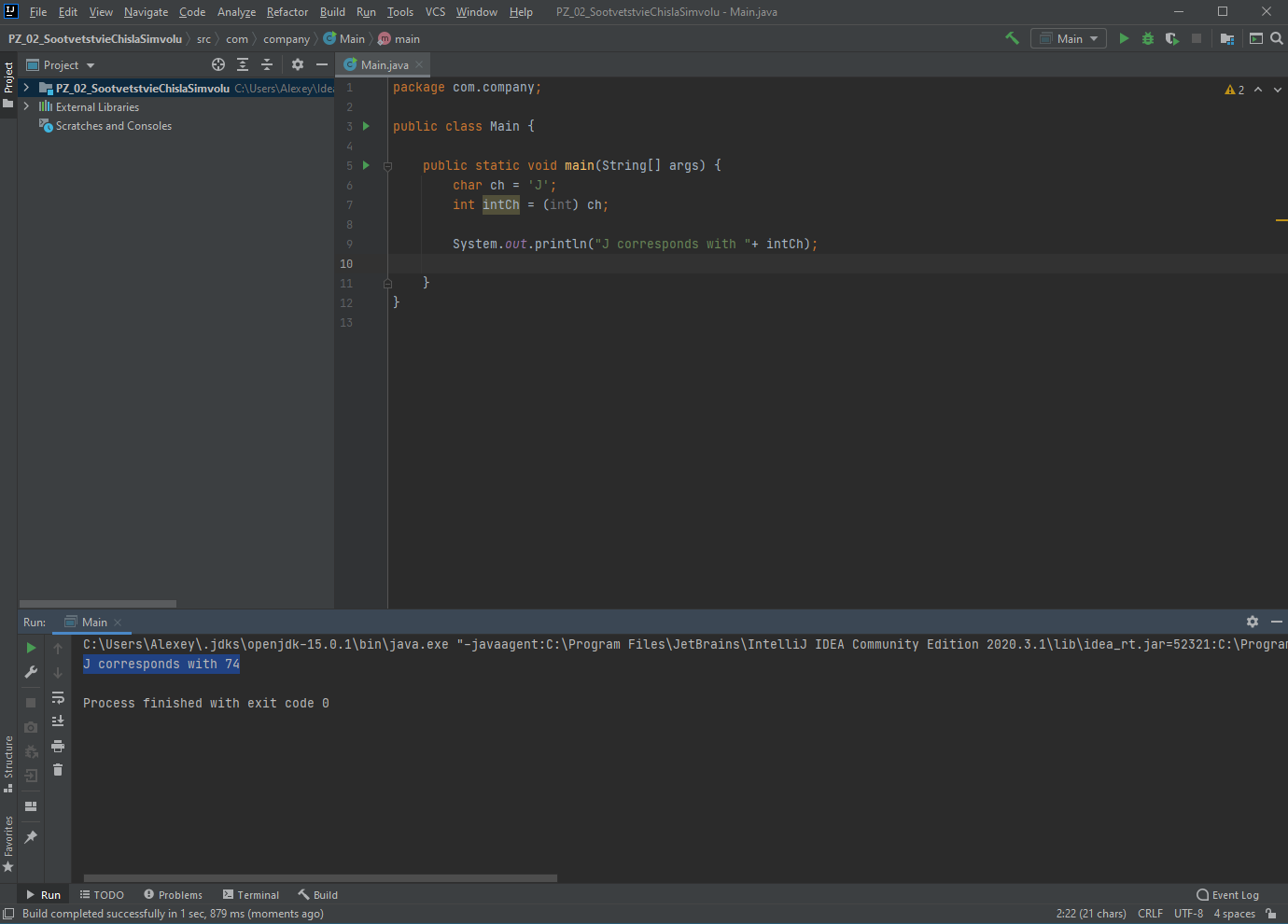
}

}

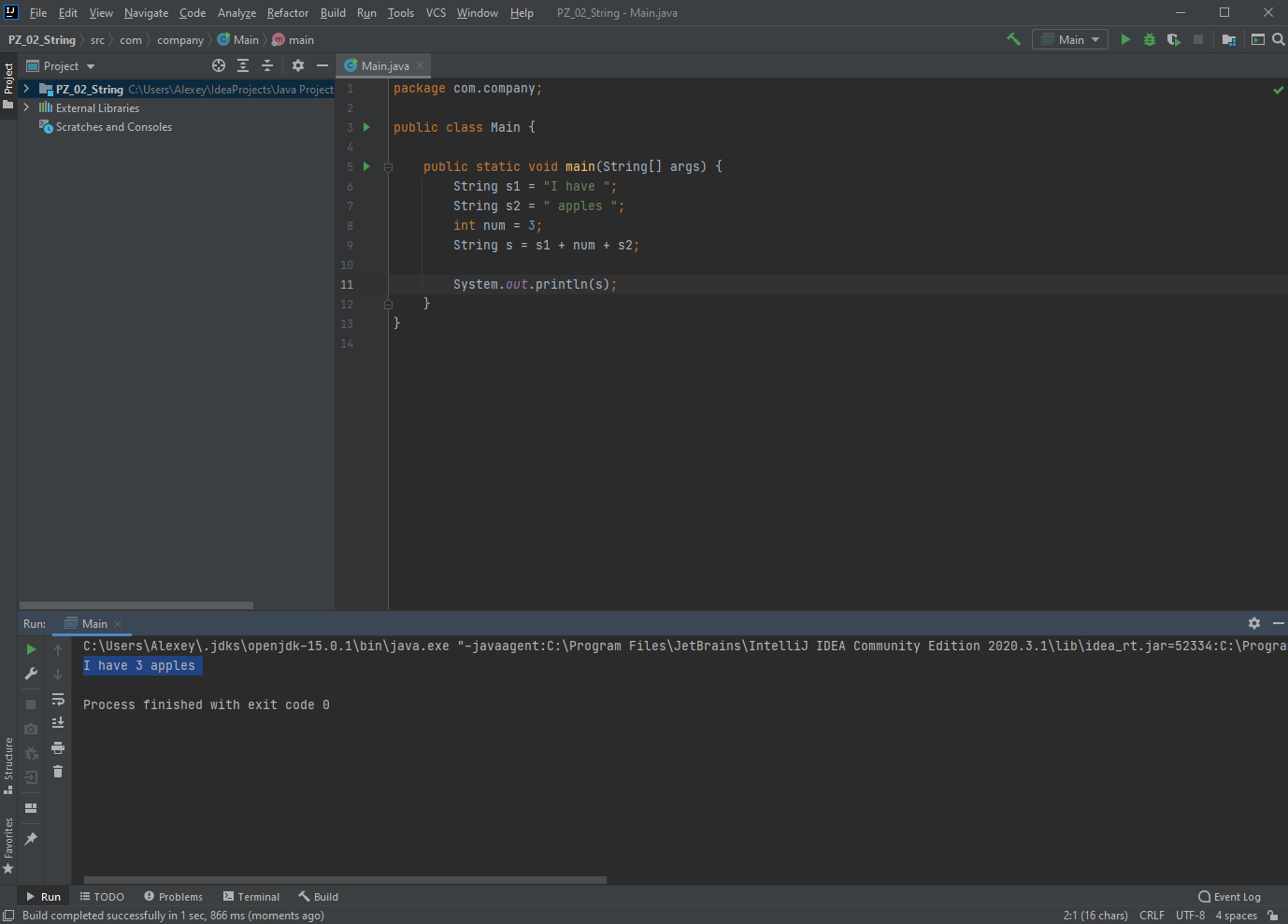
1. Скриншоты выполнения программ.



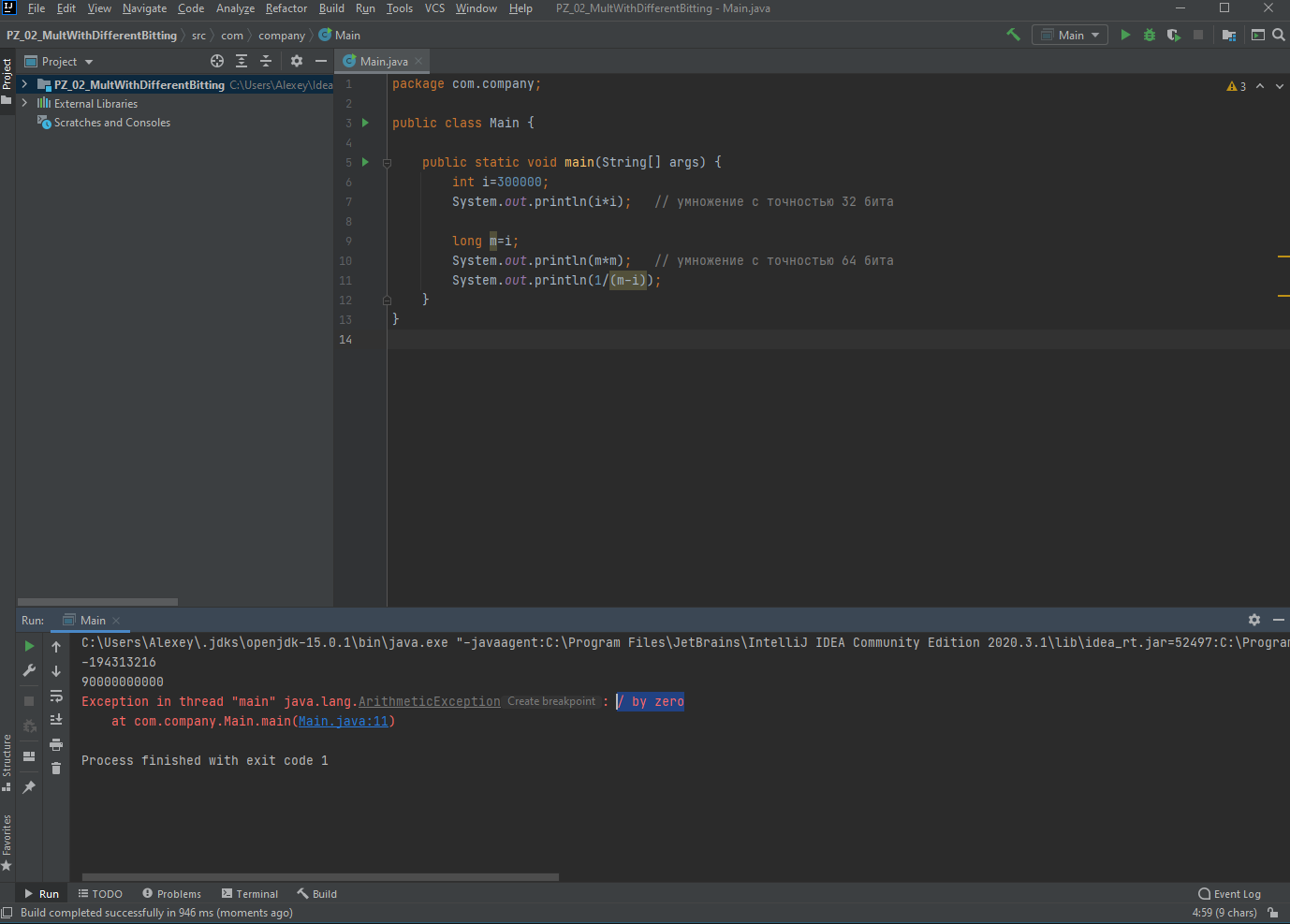
1. *Фрагменты программного кода, приведенного тексте «Практическое занятие № 2».*



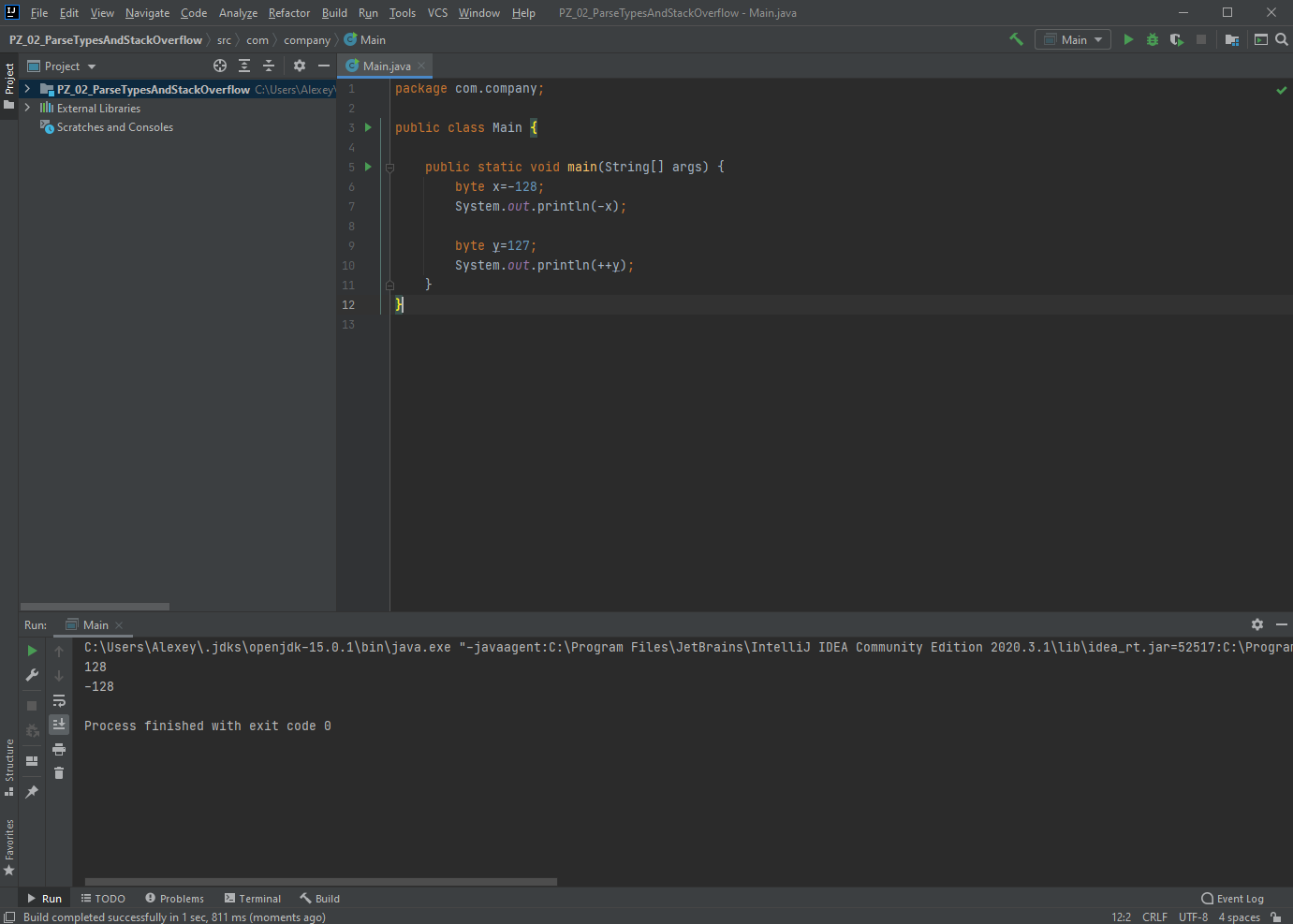
1. *Фрагменты программного кода, приведенного тексте «Практическое занятие № 2».*



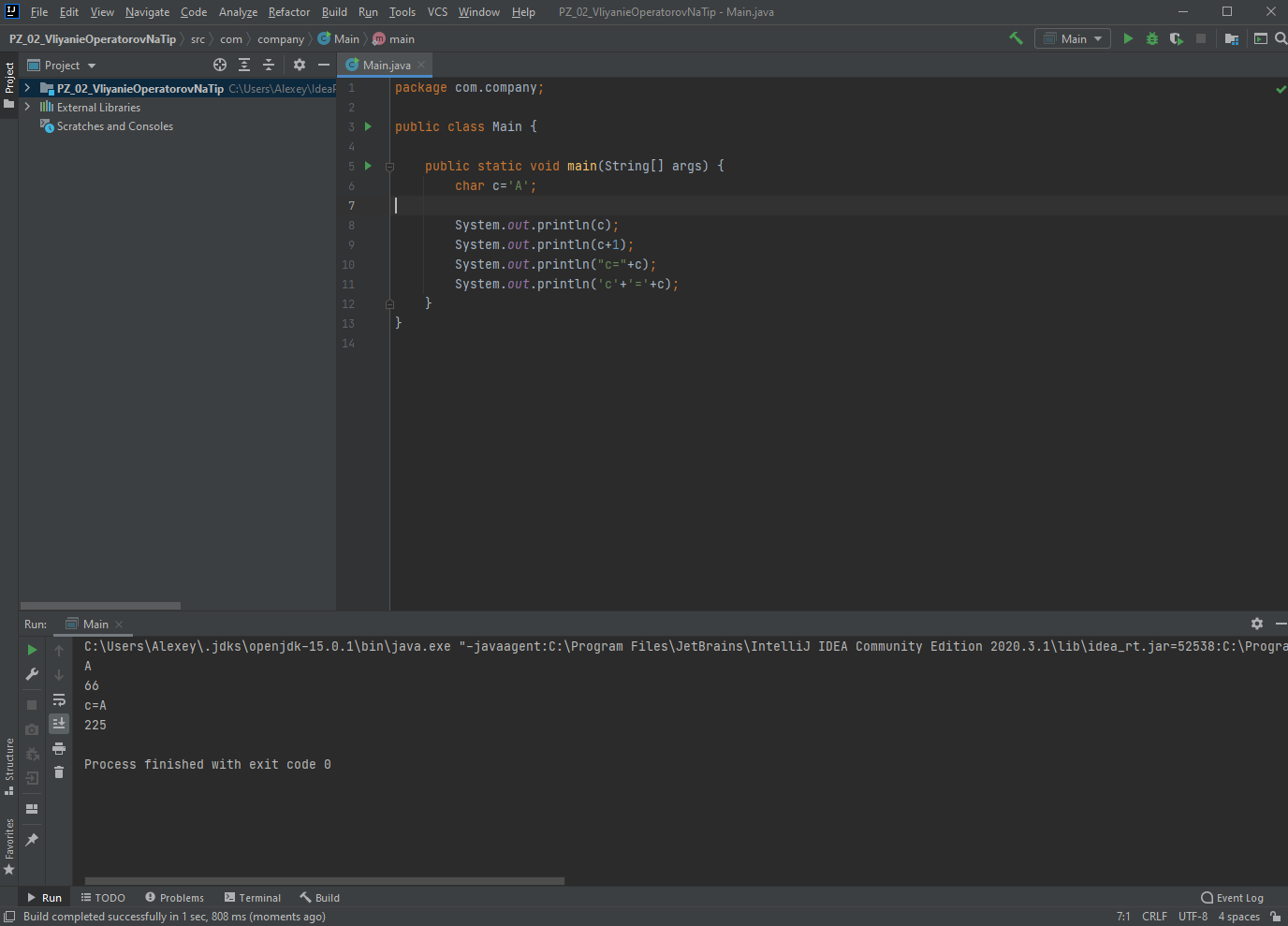
1. *Фрагменты программного кода, приведенного тексте «Практическое занятие № 2».*



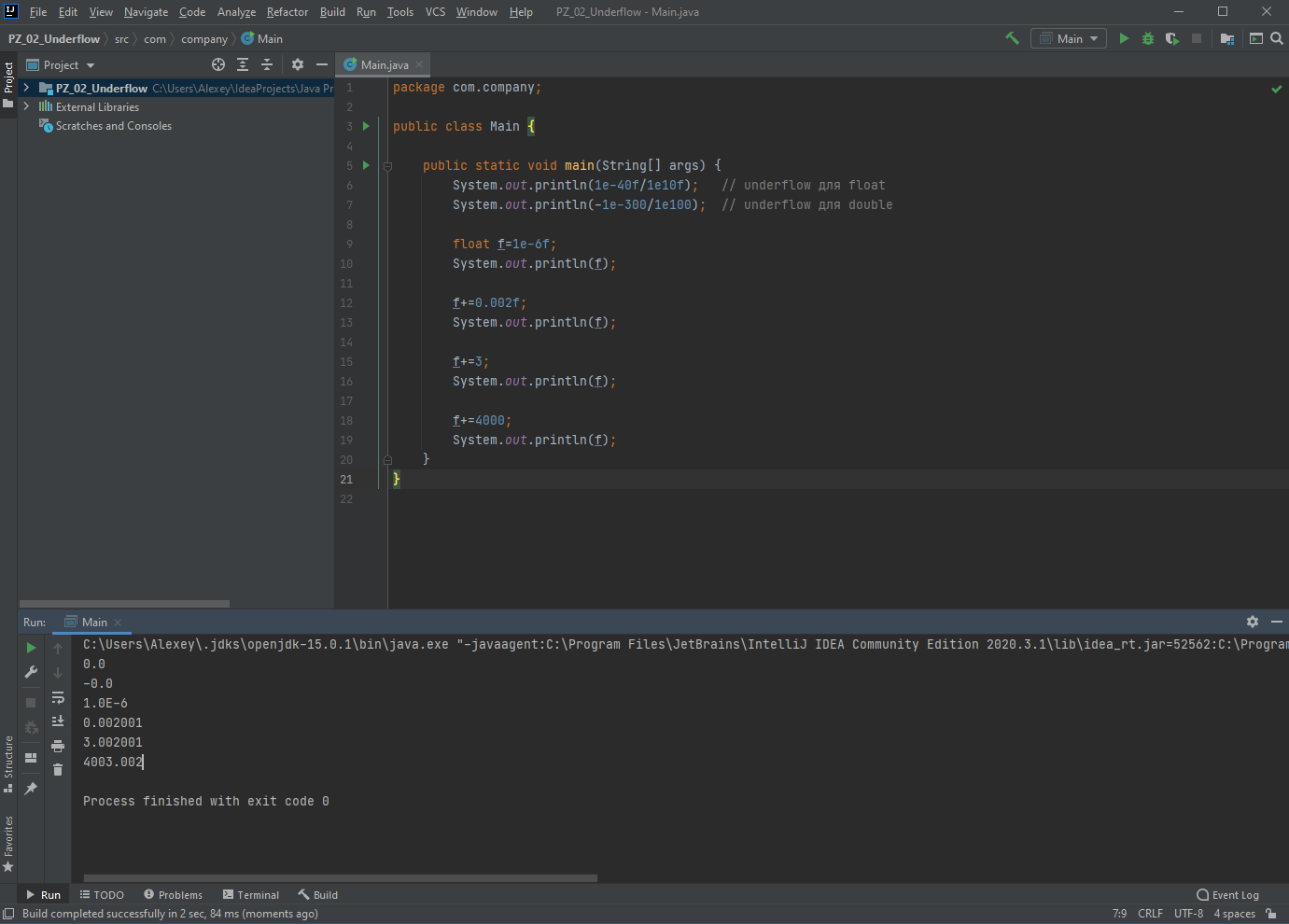
1. *Фрагменты программного кода умножения с разным количеством битов чтения и попытки деления приведенного в тексте лекционного занятия № 2.*



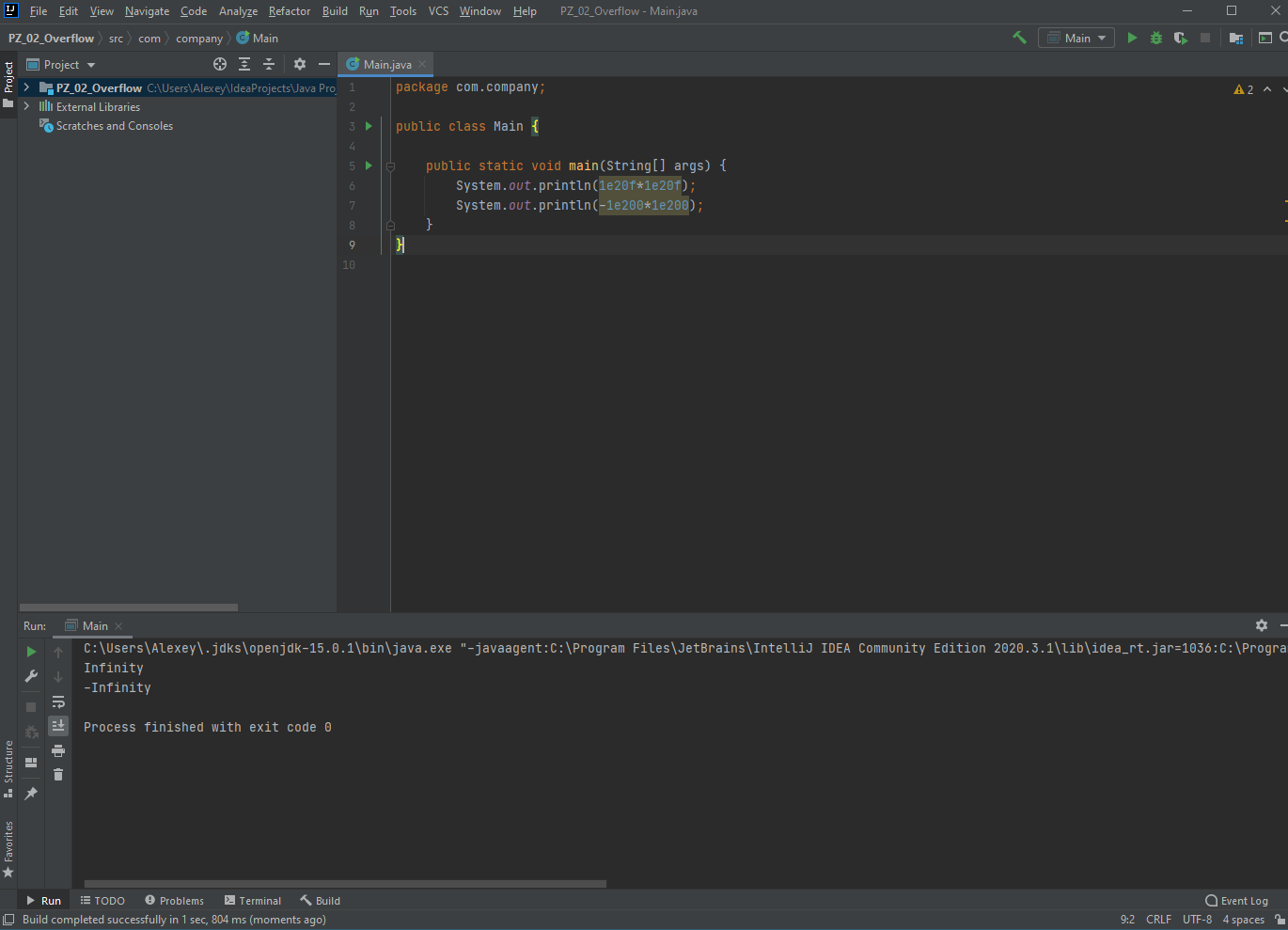
1. *Фрагменты программного кода переполнения стека приведенного в тексте лекционного занятия № 2.*



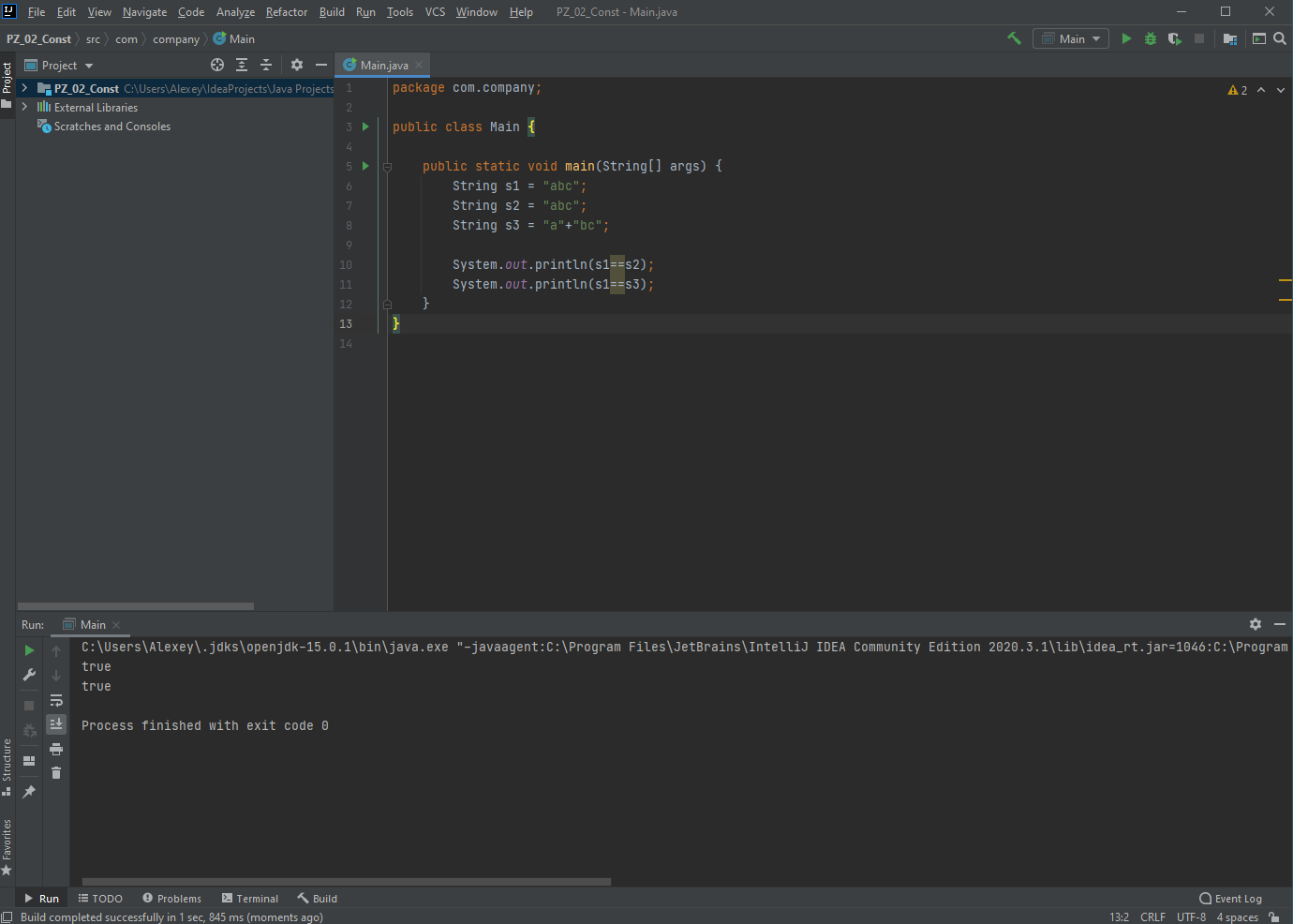
1. *Фрагменты программного кода о зависимости типа данных от операнда приведенного в тексте лекционного занятия № 2.*



1. *Фрагменты программного кода Underflow приведенного в тексте лекционного занятия № 2.*



1. *Фрагменты программного кода Overflow приведенного в тексте лекционного занятия № 2.*



1. *Фрагменты программного кода о константах приведенного в тексте лекционного занятия № 2.*
2. Ссылка на исходный код с помощью онлайн инструментария GitHub: (<https://github.com/AlexeyLepov/SPbCK_Lepov_PZ_02.git>).