Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №2**

**«Перевод чисел между различными системами счисления»**

**Вариант: 76**

Выполнил:

студент 1 курса

Батманов Даниил Евгеньевич

Группа: Р3107

Принял:

Белозубов Александр Владимирович

Отчёт принят «\_\_»\_\_\_\_\_2022 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2022

**Оглавление**

[Задание 3](#_Toc116942465)

[Основные этапы вычисления 4](#_Toc116942466)

[Заключение 16](#_Toc116942467)

[Список литературы 17](#_Toc116942469)

# 

# 

# 

# Задание

1. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
2. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
3. Показать, исходя из выбранных вариантов сообщений (по 4 у каждого – часть №1 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
4. Наоснованииномеравариантазаданиявыбрать1полученноесообщениев виде последовательности 11-символьного кода.
5. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
6. Показать, исходя из выбранного варианта сообщений (по 1 у каждого – часть №2 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
7. Сложить номера всех 5 вариантов заданий. **Умножить полученное число на 4**. Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.
8. Написать программу на любом языке программирования, которая на вход из командной̆ строки получает набор из 7 цифр «0» и «1», записанных подряд, анализирует это сообщение на основе классического кода Хэмминга (7,4), а затем выдает правильное сообщение (только информационные биты) и указывает бит с ошибкой̆ при его наличии.

# Основные этапы вычисления

**Задание №1.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* |
| **№ I** | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| **№ II** | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| **№ III** | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| **№ IV** | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

Таблица 1

**Задание №2.**

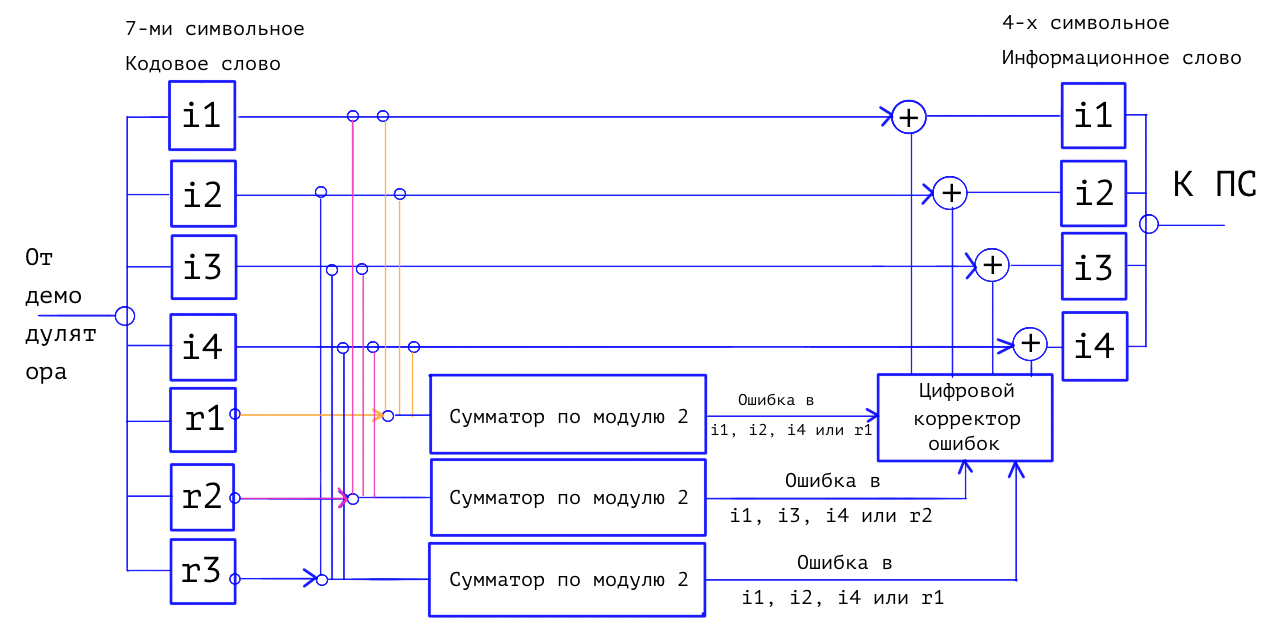


Рисунок 1

**Задание №3.**

Построим таблицу Хэмминга:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | S |
| **1** | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X | S2 |
|  |  |  |  | X | X | X | X | S3 |

Таблица 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Синдром S (S1, S2, S3 )* | **000** | **001** | **010** | **011** | **100** | **101** | **110** | **111** |
| *Конфигурация ошибок (позиция в сообщении)* | НЕТ | 0001000 | 0100000 | 0000010 | 1000000 | 0000100 | 0010000 | 0000001 |
| *Ошибка в символе* | НЕТ | r3 | r2 | i3 | r1 | i2 | i1 | i4 |

Таблица 3

Из представленных в таблице данных можно понять:

1. какие информационные биты (i) контролируют проверочные биты (r):

* r1 = i1 ⊕ i2 ⊕ i4;
* r2 = i1 ⊕ i3 ⊕ i4;
* r3 = i2 ⊕ i3 ⊕ i4;

1. синдромы (наборы контрольных сумм информационных и проверочных разрядов) для таблицы Хэмминга:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4;
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4;
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4.

Теперь, основываясь на таблицах 1 и 2, мы можем определить имеются ли в принятых 7-символьных сообщениях (кодах) из таблицы 1 ошибки. Также, если они есть, то мы их исправим.

**Сообщение №1.**

Само сообщение: 0001100

Составим таблицу для сообщения:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  |
| **Сообщение** | ***0*** | ***0*** | ***0*** | ***1*** | ***1*** | ***0*** | ***0*** |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | S |
| **1** | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X | S2 |
| **4** |  |  |  | X | X | X | X | S3 |

Таблица 4

Вычислим синдромы для сообщения:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

S1S2S3 = 100, соответственно (по таблице 3), ошибка заключается в проверочном бите r1. Чтобы исправить эту ошибку, необходимо инвертировать r1, значит, правильное сообщение: 1001100.

**Сообщение №2.**

Само сообщение: 1011110

Составим таблицу для сообщения:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  |
| **Сообщение** | ***1*** | ***0*** | ***1*** | ***1*** | ***1*** | ***1*** | ***0*** |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | S |
| **1** | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X | S2 |
| **4** |  |  |  | X | X | X | X | S3 |

Таблица 5

Вычислим синдромы для сообщения:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

S1S2S3 = 101, соответственно (по таблице 3), ошибка заключается в информационном бите i2. Чтобы исправить эту ошибку, необходимо инвертировать i2, значит, правильное сообщение: 1011010.

**Сообщение №3.**

Само сообщение: 0110001

Составим таблицу для сообщения:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  |
| **Сообщение** | ***0*** | ***1*** | ***1*** | ***0*** | ***0*** | ***0*** | ***1*** |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | S |
| **1** | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X | S2 |
| **4** |  |  |  | X | X | X | X | S3 |

Таблица 6

Вычислим синдромы для сообщения:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1

S1S2S3 = 011, соответственно (по таблице 3), ошибка заключается в информационном бите i3. Чтобы исправить эту ошибку, необходимо инвертировать i3, значит, правильное сообщение: 0110010.

**Сообщение №4.**

Само сообщение: 1010000

Составим таблицу для сообщения:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  |
| **Сообщение** | ***1*** | ***0*** | ***1*** | ***0*** | ***0*** | ***0*** | ***0*** |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | S |
| **1** | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X | S2 |
| **4** |  |  |  | X | X | X | X | S3 |

Таблица 7

Вычислим синдромы для сообщения:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0

S1S2S3 = 010, соответственно (по таблице 3), ошибка заключается в проверочном бите r2. Чтобы исправить эту ошибку, необходимо инвертировать r2, значит, правильное сообщение: 1000000.

**Задание №4.**

Сообщение в виде последовательности 11-символьного кода: 001110011000100.

**Задание №5.**

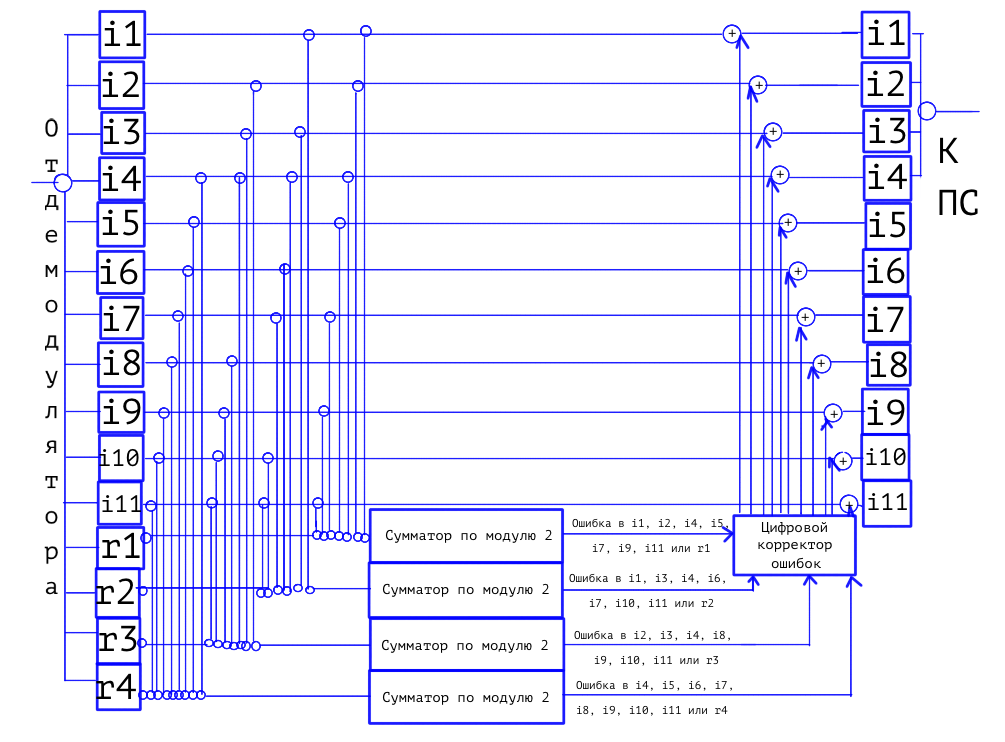
****

Рисунок 2

**Задание №6.**

Само сообщение: 001110011000100

Составим таблицу для сообщения:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
| **Сообщение** | ***0*** | ***0*** | ***1*** | ***1*** | ***1*** | ***0*** | ***0*** | ***1*** | ***1*** | ***0*** | ***0*** | ***0*** | ***1*** | ***0*** | ***0*** |
| 2k | *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* | *r4* | *i5* | *i6* | *i7* | *i8* | *i9* | *i10* | *i11* | S |
| **1** | X |  | X |  | X |  | X |  | X |  | X |  | X |  | X | S1 |
| **2** |  | X | X |  |  | X | X |  |  | X | X |  |  | X | X | S2 |
| **4** |  |  |  | X | X | X | X |  |  |  |  | X | X | X | X | S3 |
| **8** |  |  |  |  |  |  |  | X | X | X | X | X | X | X | X | S4 |

Таблица 8

Вычислим синдромы для сообщения:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11 = 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 = 1
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i­8 ⊕ i9 ⊕ i10 ⊕ i11 = 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1
* S4 = r4 ⊕ i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

S1S2S3S4 = 0111, соответственно (по таблице 3), ошибка заключается в информационном бите i10. Чтобы исправить эту ошибку, необходимо инвертировать i10, значит, правильное сообщение: 001110011000110.

**Задание №7.**

Сумма всех 5 вариантов заданий: 58 + 95 + 20 + 10 + 76 = 259

Эта сумма, умноженная на 4: 259 \* 4 = 1036

Соответственно, i (число информационных разрядов) = 1036

Тогда, высчитаем минимальное число проверочных разрядов для i по формуле 2r >= r + i + 1, где r – число проверочных разрядов:

2r  >= r + 1036 + 1

2r  >= r + 1037

Методом подбора, определим минимальное значение r:

|  |  |
| --- | --- |
| **Значение r** | **2r  >= r + 1037** |
| 1 | 2 >= 1038 (ложь) |
| … | … |
| 9 | 512 >= 1046 (ложь) |
| 10 | 1024 >= 1047 (ложь) |
| *11* | *2048 >= 1048 (истина)* |
| … | … |

Таблица 9

Соответственно, минимальное число проверочных разрядов (r) равно 11.

Вычислим коэффициент избыточности по формуле: КИ = r / (i + r)

Соответственно, КИ = 11 / (11 + 1036) = 11 / 1047 ≈ 0,0105062.

**Задание №8.**

***Код на Java.***

import java.util.Scanner;

public class Main {

public static void main(String[] yolo) {

Scanner scan = new Scanner(System.in);

int stop = 0;

System.out.println("Введите сообщение в виде 7 цифр из бинарного набора:");

String code = scan.nextLine();

scan.close();

if (code.length() != 7) System.out.println("Введены некорректные данные.");

else {

String[] mass = code.split("");

int[] mas = new int[7];

for (int i = 0; i < 7; i++) {

if (mass[i].equals("1") || mass[i].equals("0")) mas[i] = Integer.valueOf(mass[i]);

else {

stop++;

if (stop == 1) System.out.println("Введены некорректные данные.");

}

}

if (stop == 0) {

int s1 = (mas[0] + mas[2] + mas[4] + mas[6]) % 2;

int s2 = (mas[1] + mas[2] + mas[5] + mas[6]) % 2;

int s3 = (mas[3] + mas[4] + mas[5] + mas[6]) % 2;

String str = Integer.toString(s1) + Integer.toString(s2) + Integer.toString(s3);

switch (str) {

case "000":

System.out.print("Правильное сообщение: " + code + "." + "\n");

System.out.print("Поздравляем! Бит с ошибкой отсутствует!" + "\n");

break;

case "001":

if (mas[3] == 0) mas[3] = 1;

else mas[3] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: побочный бит r3. Позиция: 0001000." + "\n");

break;

case "010":

if (mas[1] == 0) mas[1] = 1;

else mas[1] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: побочный бит r2. Позиция: 0100000." + "\n");

break;

case "011":

if (mas[5] == 0) mas[5] = 1;

else mas[5] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: информационный бит i3. Позиция: 0000010." + "\n");

break;

case "100":

if (mas[0] == 0) mas[0] = 1;

else mas[0] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: побочный бит r1. Позиция: 1000000." + "\n");

break;

case "101":

if (mas[4] == 0) mas[4] = 1;

else mas[4] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: информационный бит i2. Позиция: 0000100." + "\n");

break;

case "110":

if (mas[2] == 0) mas[2] = 1;

else mas[2] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: информационный бит i1. Позиция: 0010000." + "\n");

break;

case "111":

if (mas[6] == 0) mas[6] = 1;

else mas[6] = 0;

System.out.print("Правильное сообщение: ");

for (int i = 0; i < 7; i++) System.out.print(mas[i]);

System.out.print("." + "\n");

System.out.print("Бит с ошибкой: информационный бит i4. Позиция: 0000001." + "\n");

break;

}

}

}

}

}

***Примеры выполнения программы:***

*Тест 1*

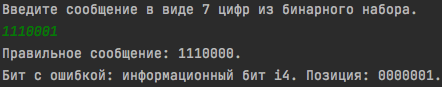


Рисунок 3

*Тест 2*

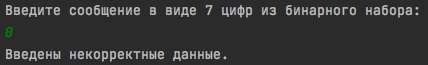


Рисунок 4

*Тест №3*

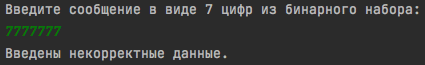


Рисунок 5

*Тест №4*

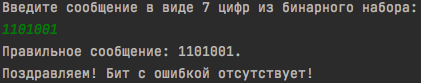


Рисунок 6

# Заключение

В ходе выполнения данной лабораторной работы я изучил код Хэмминга: научился строить таблицы кода Хэмминга, вычислять число проверочных разрядов из числа информационных разрядов, а затем и вычислять коэффициент избыточности; научился искать ошибки в сообщениях в виде кода Хемминга и исправлять их. Более того, мне удалось написать программу на языке программирования Java, которая проводит анализ для классического кода Хэмминга (7;4), а затем выдает правильное сообщение и указывает бит с ошибкой при его наличии.

# Список литературы

**Oracle** Java Documentation. - https://docs.oracle.com/en/java/

**Балакшин П.В. Соснин В.В., Машина Е.А.** – СПб: Университет ИТМО, 2020 // Информатика. Методическое пособие "Информатика". Раздел 2 "Системы счисления". https://vk.com/doc-31201840\_566998093

**Петр Михалевский** Легкий старт в Java. - Stepik - https://stepik.org/course/90684/syllabus