

WEB BASED LEARNING TOOL OF HAMMING CODES IN MATRIX FORM¹¹

Yuksel Aliev, PhD Student

Department of Computing,
University of Ruse "Angel Kanchev", Bulgaria
E-mail: yaliev@uni-ruse.bg

Assoc. Prof. Galina Ivanova, PhD

Department of Computing,
University of Ruse "Angel Kanchev", Bulgaria
Phone: 086-821 521
E-mail: giivanova@uni-ruse.bg

Abstract: This report presents a web-based application for studying Hamming code in matrix form. The proposed application can work with up to 26-bit information combination in encoding mode and up to 32-bit code combination in decoding mode. The software provides an opportunity to work with the so-called. Hamming's extended code, which makes the code capable of detecting all double and corrects all single errors at the expense of an additional control bit. An interactive block diagram has been developed and it is integrated in the web-based application. The sequence of all steps for working with the environment is presented and students are informed at each stage with information about all the actions for the current stage. The application was created for educational purposes for students studying "Computer Systems and Technologies" at the University of Ruse "Angel Kanchev". Survey results with students will be presented and discussed.

Keywords: Hamming code, matrix, coding, decoding, error detecting, error correcting, learning tool, interactive model, JavaScript, konvaJS, html5

JEL Codes: I23, D83, L86

ВЪВЕДЕНИЕ

Съвременните студенти, родени с цифровите технологии, предпочитат интерактивния подход на учене пред традиционния. При традиционното обучение студентите често изпитват трудности при усвояването на преподавания материал. Прилагането на интерактивните симулационни програмни модели в учебния процес дава възможност на студентите да възприемат материала в по-удобен и атрактивен за тях вид. Един от приоритетите на университетите в наши дни е развитието на модерно образование с въвеждането на интерактивни образователни софтуерни инструменти, които ще дадат на студентите гъвкавост да учат по всяко време, на всяко място и с всяко интерактивно устройство.

Кодовете за контрол на грешки (ECC) (Hamming, R. W., 1950; Blahut, R. E., 1983) са поредица от числа, генерирани по специални алгоритми за откриване и/или коригиране на грешки по време на предаване през комуникационен канал с шум или по време на съхраняване в памет. Без кодовете за контрол на грешки няма да има надеждно предаване и съхраняване на данните. Кодовете на Хеминг са важен представител на кодовете за контрол на грешките. Поради тази причина, изучаването им намира място в учебен процес е важно.

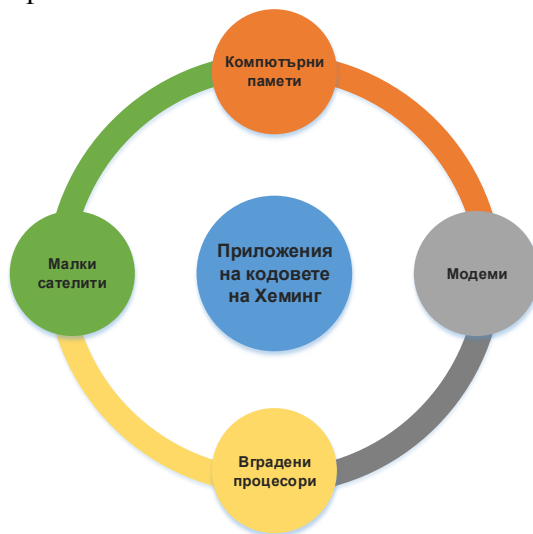
ИЗЛОЖЕНИЕ

Общи сведения за кодовете на Хеминг

¹¹ Докладът е представен на 13 ноември 2020 с оригинално заглавие на български език: УЕБ БАЗИРАН ОБУЧИТЕЛЕН ИНСТРУМЕНТ ЗА ИЗУЧАВАНЕ НА КОДОВЕ НА ХЕМИНГ В МАТРИЧНА ФОРМА

Кодовете на Хеминг спадат към групата на линейните, блокови, двоични и систематични кодове. Имат способността да откриват и коригират всички единични грешки (Moon, T. K., 2005).

Въведени са през 1950г. от Richard Hamming (Hamming, R. W., 1950). Областите на приложение са посочени на фиг. 1.



Фиг. 1. Приложение на кодовете на Хеминг

Условните обозначения на параметрите, които ще се използват по-долу за обяснение на кодовете са:

- m – брой на информационните битове;
- k – брой на контролните битове;
- n – брой на битовете в кодовата дума, където $n = m + k$;
- l_0 – брой на грешките, които кода може да открива;
- d – минимално кодово разстояние (Хемингово разстояние).

Минималното кодово разстояние при стандартните кодовете на Хеминг е $d=3$ (кода може да открива и коригира единични грешки ($l_0=1$)), и $d=4$ за разширените кодове на Хеминг (кода може да открива двойни и коригира единични грешки ($l_0=2$)) (Ranpara, S., 1999). При случай на $l_0=1$, броят на контролните битове се определя от следните две условия:

$$2^k - k \geq m + 1, \quad (1)$$

$$2^k \geq n + 1. \quad (2)$$

За построяването на кодовете на Хеминг съществува два подхода – обикновен и матричен метод. В настоящия доклад се разглежда матричния метод, който е заложен и в обучителния интерактивен модел.

За построяването на кодовете на Хеминг в матрична форма се използват две матрици, обозначени с H и H' , състоящи се от нули и единици. На фиг.2а е показана H матрица за код с $n=12$ битова кодова комбинация и $m=8$ бита информационна комбинация. Стойностите на матрицата се попълва по колони от горе на долу, които се определят по двоично-десетичния код 8421 на номера на колоната, където най-младшият разряд се намира на последния ред. На получената матрица се поставят етикети на колоните обозначени със S_i и C_i , където е $i > 0$ е пореден номер. Със S_i се обозначават информационните, а с C_i контролните битове. Поредицата от тези етикети оказва и последователността на битовете в закодираната кодова комбинация (Gherman, V., 2011).

Матрицата H' (фиг.2б.) се получава като резултат от H матрицата, но с разменени колони. Колони на матрицата за посочения по-горе код има следната последователност: $S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 C_4 C_3 C_2 C_1$, т.е първите 8 позиции са за информационните битове, а последните 4 – за контролните.

$$\begin{array}{c}
 \begin{array}{cccccccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
 C_1 & C_2 & S_1 & C_3 & S_2 & S_3 & S_4 & C_4 & S_5 & S_6 & S_7 & S_8
 \end{array} \\
 H = \begin{array}{c}
 8 \\
 4 \\
 2 \\
 1
 \end{array} \left| \begin{array}{cccccccccccc}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0
 \end{array} \right|
 \end{array}
 \qquad
 \begin{array}{c}
 \begin{array}{cccccccccccc}
 S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & C_4 & C_3 & C_2 & C_1
 \end{array} \\
 H' = \left| \begin{array}{cccccccc|cccc}
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1
 \end{array} \right|
 \end{array}$$

а) матрица H б) матрица H'

Фиг. 2. Матриците за построяване на кодовете на Хеминг

За всеки ред на матрицата H' се извеждат уравненията за кодиране/декодиране, колкото са и контролните битове. На фиг. 3. е показана системата от уравнения за кодиране и декодиране за код (12,8).

$$\begin{array}{l}
 C_4 = S_5 \oplus S_6 \oplus S_7 \oplus S_8 \\
 C_3 = S_2 \oplus S_3 \oplus S_4 \oplus S_8 \\
 C_2 = S_1 \oplus S_3 \oplus S_4 \oplus S_6 \oplus S_7 \\
 C_1 = S_1 \oplus S_2 \oplus S_4 \oplus S_5 \oplus S_7
 \end{array}$$

Фиг. 3. Система от уравнения за кодиране/декодиране

Интерактивен обучителен модел

Разработен е уеб базиран обучителен софтуер за кодове на Хеминг в матрична форма. Приложението е предвиден за работа с до 26-битова информационна комбинация в режим на кодиране и до 32-битова кодова комбинация в режим на декодиране.

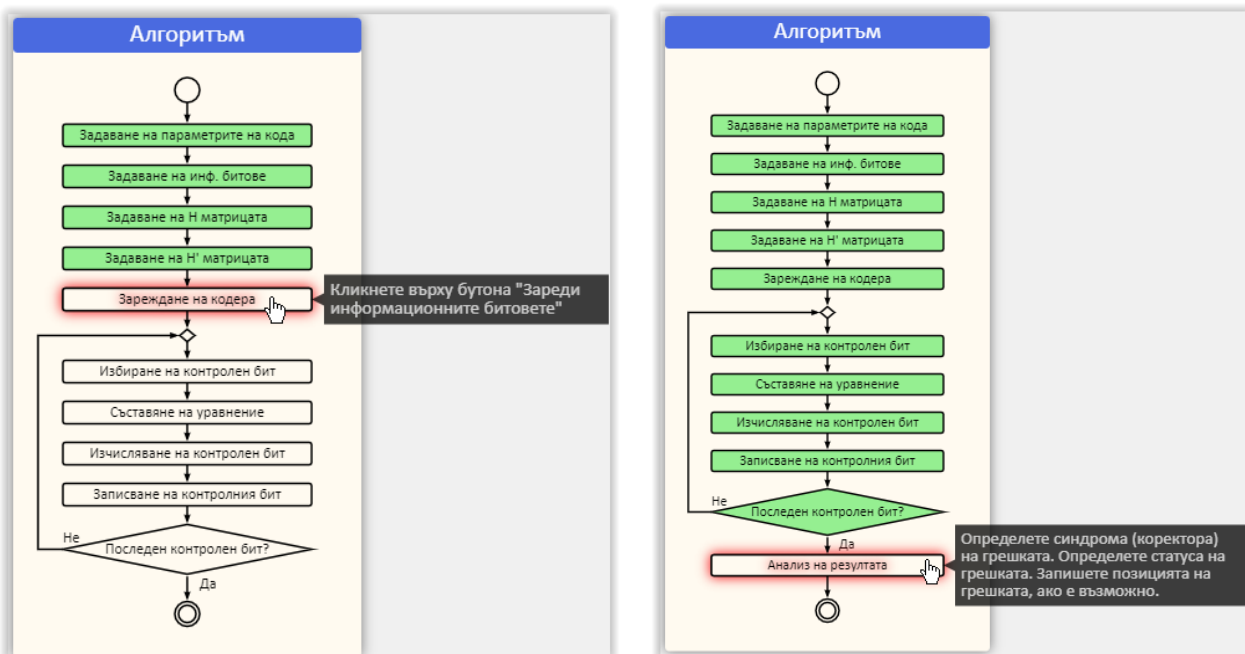
Приложението е разработено чрез използване на програмните езици *HTML5* и *JavaScript* (Mackie, S., 2018) (Mackie, S., 2018). За разработка на отделните компоненти на приложението са използвани библиотеките *KonvaJS* (KonvaJS, 2020,) и *JQuery* (JQuery, 2020).

Потребителския интерфейс на програмния модел е реализиран на два езика - български (BG) и английски (EN). Приложението може да работи в два процеса на работа – *кодиране* и *декодиране*. Състои се от три компонента: *HCmat*, *Algorithm* и *Statistics*.

HCmat е основният компонент на приложението, в който се реализира главната функционалност. По време на работа със софтуерната система след всяка изпълнена стъпка се прави валидация на действието, в случай на грешка се извежда съобщение за грешката. Докато не се изпълни правилната стъпка или не се въведе правилната стойност не се допуска преминаване към следващата стъпка.

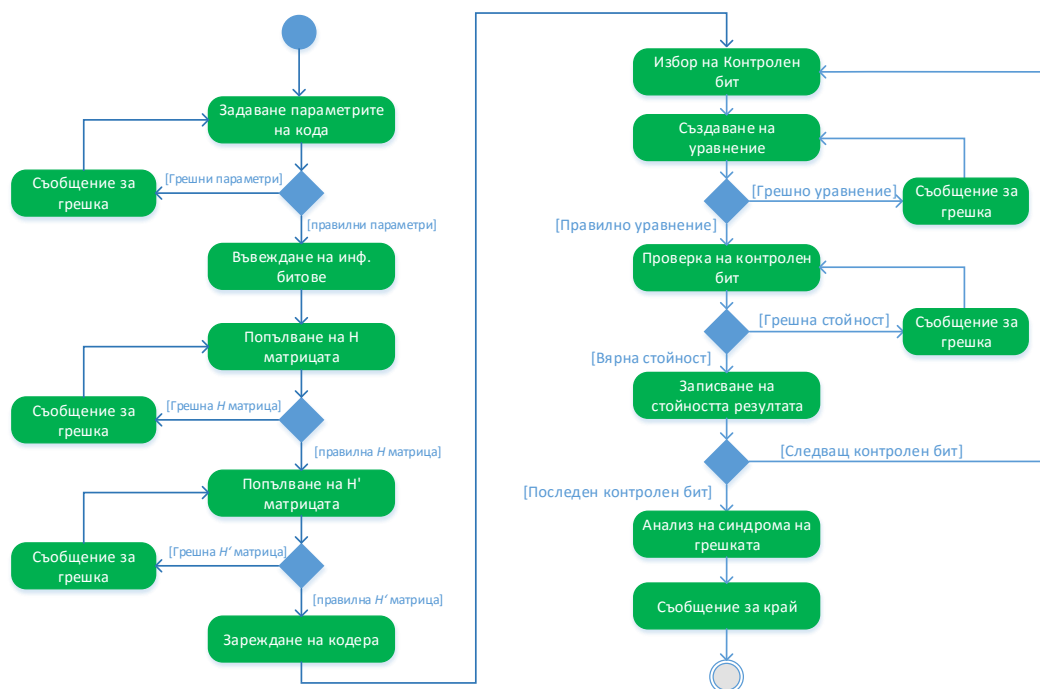
Компонентът *Algorithm* е спомагателен и служи за показване последователността от действия за решаване на задачата. На фиг. 4. е показана визуализацията на алгоритъма в режим на кодиране (фиг.4а) и декодиране (фиг.4б). Блоквата схема е интерактивна, като при посочване на дадена стъпка се извежда информация за нужното действие на решаване на стъпката. За по-добра нагледност, стъпките които са приключили се оцветяват в зелен цвят, а текущата – в жълт цвят.

Компонентът *Statistics* служи за събиране на информация като: време на работа със симулационната система (таймер), брой допуснати грешки както и историята на грешките.

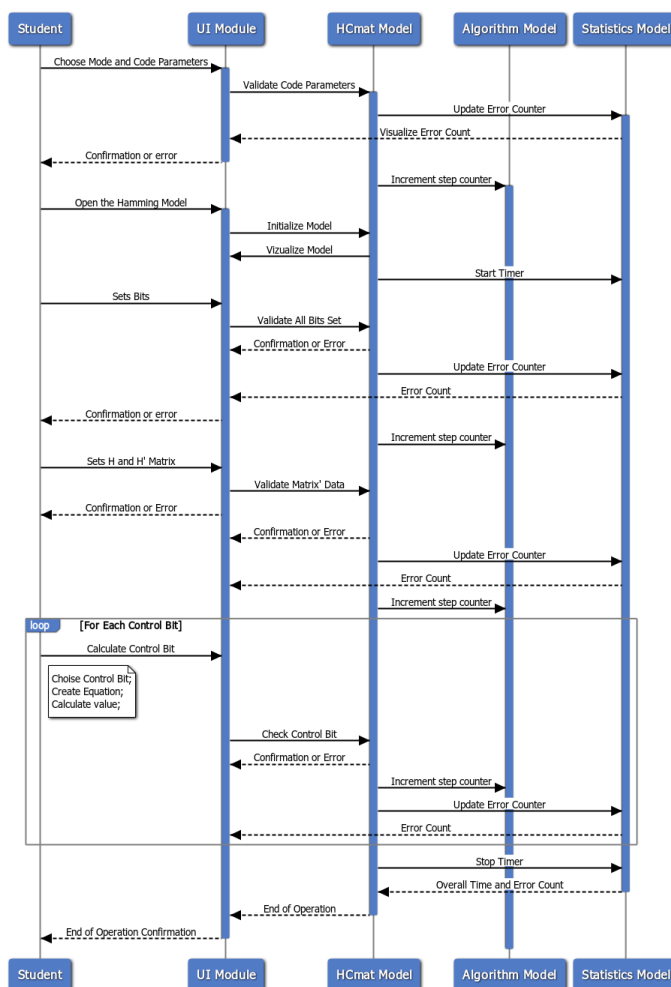


Фиг. 4. Интерактивна блок-схема на алгоритма на работа

На фиг.5. може да се види диаграмата на дейностите (Activity diagram) на приложение-то в процес на декодиране. От нея се вижда последователността от действия и местата, където се извършва валидация на стъпките. А на фиг. 6. може да се види диаграмата на последователностите (Sequence diagram), където се показва взаимодействието между потребителя (студента), потребителския интерфейс и компонентите на приложението (*HCmat Model, Algorithm model u Statistics model*).

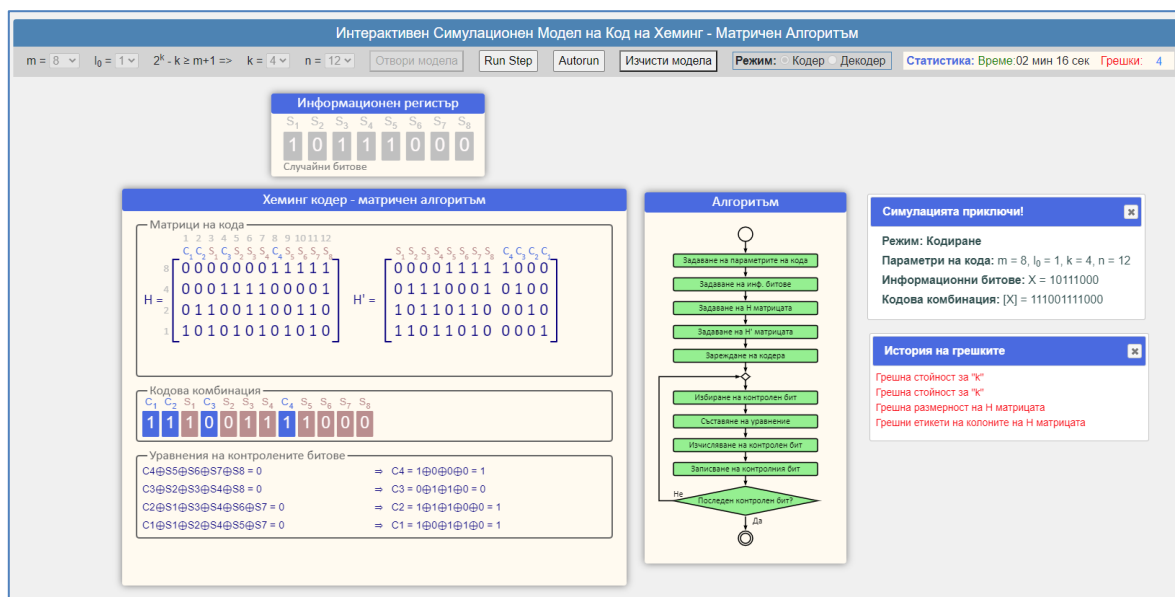


Фиг. 5. Диаграма на дейностите в процес на декодиране



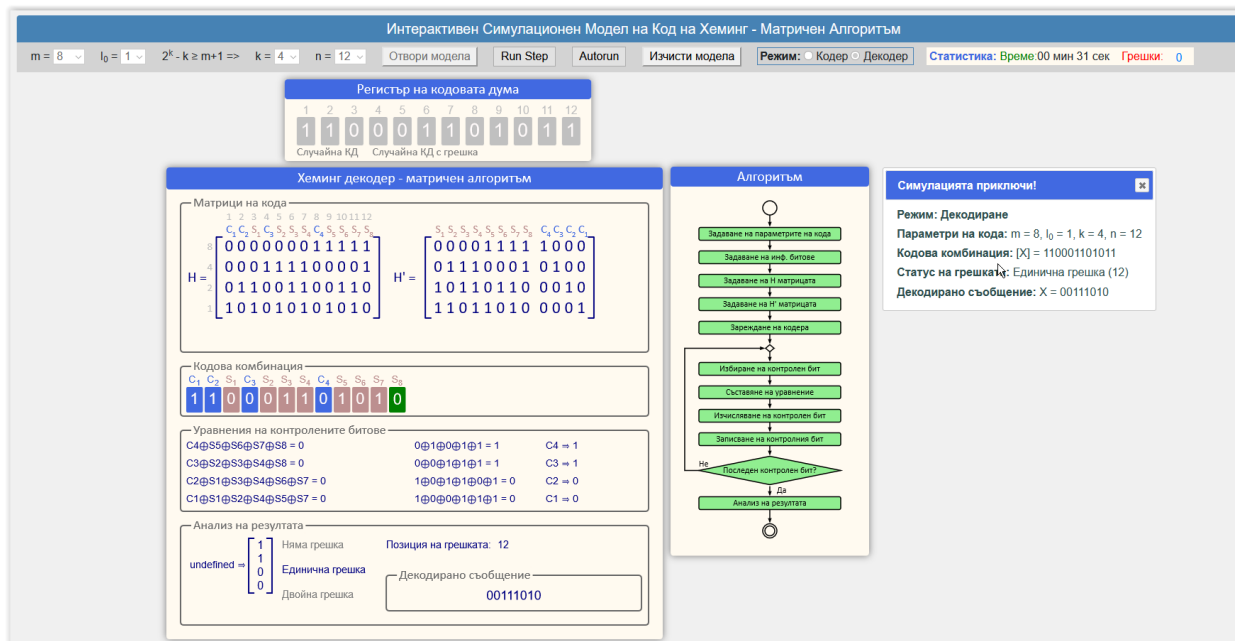
Фиг. 6. Диаграма на последователностите в процес на кодиране

На фиг. 7. е показан общият вид на потребителския интерфейс на софтуерната система, след като е извършен процес на кодиране на примерна 8-битова информационна комбинация (10111000). В края на задачата се вижда, че за контролните битове се получава: $C_1=1$, $C_2=1$, $C_3=0$, $C_4=1$, които се поставят на зададените им местоположения в кодовата комбинация. В резултат за кодираната кодова комбинация се получава 111001111000.



Фиг. 7. Общ вид на потребителския интерфейс в края на процес кодиране

На фиг. 8. може да се види общият вид на потребителския интерфейс на софтуерната системата, след като е извършен процес на декодиране на примерна 12-битова кодова комбинация (110001101011). В резултат от направената проверка на контролните битове е получен синдром коректор 1100, който показва, че в кодовата комбинация е настъпила единична грешка, позицията на която се определя от матрицата H' . За конкретния пример вектора на синдром коректора съвпада със стълб S_8 на матрицата H' , т.е. грешката е в позиция 12 според матрицата H . В крайна сметка след корекцията за декодираното съобщение се получава комбинацията 00111010.



Фиг. 8. Общ вид на потребителския интерфейс в края на процес декодиране

ЗАКЛЮЧЕНИЕ

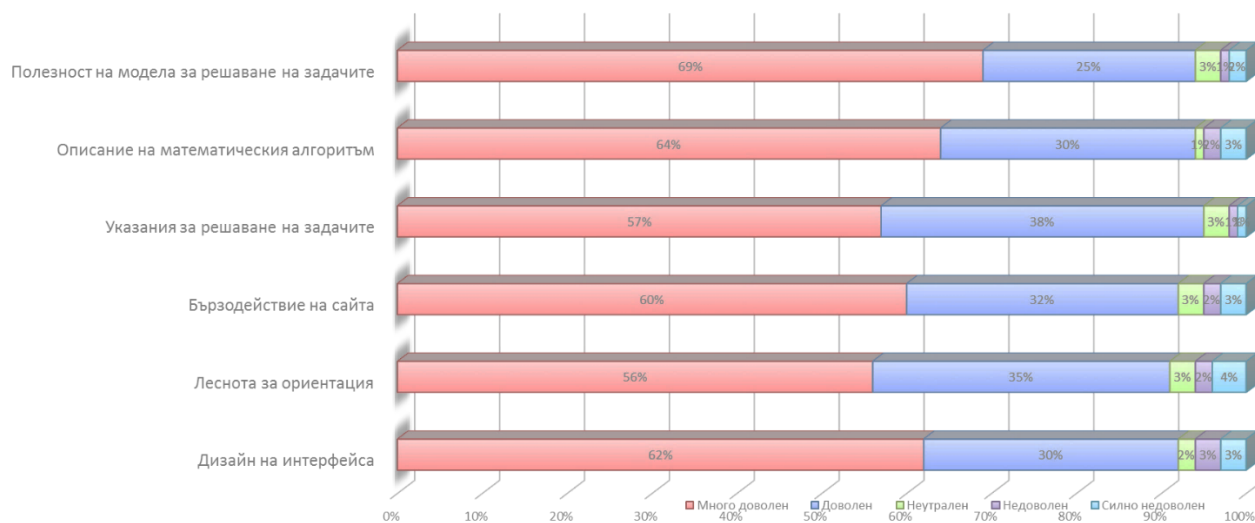
Представена е интерактивна система за изучаване кодовете на Хеминг, позволяващ симулирането на процесите на кодиране и декодиране. Уеб базираната система помага на студентите при изучаване на процесите на кодиране (до 28-битова информационна комбинация) и декодиране (до 32-битова кодова комбинация). Приложението е създадено с обучителна цел за студентите от специалност Компютърни системи и технологии към Русенски университет „Ангел Кънчев“ за практическите упражнения за изучаване на теорията на кодирането (Ivanova, G., 2014).

Чрез различните режими на работа (*ръчен, стъпка по стъпка и автоматичен*) се постига по-ефективно изучаването на кодовете на Хеминг. Посредством интерактивния модул *Algorithm*, се дава бърз и ясен изглед към цялостните действия при работа със средата. Чрез информацията от модула *Statistics* (време за решаване, брой допуснати грешки, включително историята на грешките) на интерактивната система, преподавателите могат по-прецизно да следят степента на усвояване на преподавания материал и да обърнат повече внимание на по-честите допуснати грешки при решаването на задачите.

Интерактивните модели подпомагат не само на студентите, а и преподавателите, в случаите когато се налага да се обясняват сложни материали, каквито са кодовете за контрол на грешки в компютърните и комуникационните системи.

За оценка на уеб-базираното приложение е проведено анкетно проучване за мнението на студентите чрез пет степенна ликертова скала. В проучването участваха 43 студента. Скалата на оценките, които можеха да дадат студентите за всеки един критерий са: *много доволен, доволен, неутрален, недоволен и силно недоволен*. Критериите за оценка бяха: *Полезност на модела за решаване на задачите; Описание на математическия алгоритъм; Указания за решаване на задачите; Бързодействие на сайта; Леснота за ориентация; Дизайн на интерфейса*. Получените резултати са показани на фиг. 9. Обобщените резултати

за всички критерии показват, че най-голям е дялът на оценка *много доволен* (между 56% и 69%), следван от *оценка доволен* (между 25% и 38%), а останалите три оценки (*неутрален*, *недоволен* и *силно недоволен*) заемат най-малък дял (между 1% и 3%).



Фиг. 9. Резултати от анкетното проучване

От получените резултати от анкетното проучване на фиг. 9 може да се направи заключението, че много голяма част от анкетираните студенти намират разработеното приложение за необходимо и полезно.

REFERENCES

- Hamming, R. W. (1950). "Error detecting and error correcting codes," in *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147-160, April 1950.
- Blahut, R. E. (1983). *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley. ISBN: 978-0-470-84356-7
- Sklar, B. (2001). *Digital Communications: Fundamentals and Applications*. Second Edition, Upper Saddle River, NJ: Prentice-Hall.
- KonvaJS, (2020). *HTML5 2d canvas js library for desktop and mobile applications*. URL: <https://konvajs.org/> (Accessed on 20.02.2020).
- jQuery, (2020). *JavaScript library*. URL: <https://jquery.com/> (Accessed on 20.02.2020).
- Mackie, S., (2018). *JavaScript: Best Practice*. SitePoint Pty. Ltd.
- Fiedler, J. (2004) - preprint, 2004 - orion.math.iastate.edu
- Gherman, V., Evain, S., Seymour, N., & Bonhomme, Y. (2011). *Generalized parity-check matrices for SEC-DED codes with fixed parity*. In 2011 IEEE 17th International On-Line Testing Symposium (pp. 198-201). IEEE.
- Moon, T. K. (2005). *Error correction coding: mathematical methods and algorithms*. John Wiley & Sons.
- Ranpara, S., Dong Sam Ha (1999). *A low-power Viterbi decoder design for wireless communications applications*. IEEE Proceedings of the Twelfth Annual IEEE International Int. ASIC Conference 1999, Washington, DC, 15-18 Sept. 1999, pp. 377-381.
- Ivanova, G., & Smrikarova, S. (2014) Virtual Laboratory for Coding Theory Training. In International Conference on e-Learning (Vol. 14, p. 147).