



## Проектният подход в обучението по информатика – пример с реализация и използване на библиотека за аритметика с рационални числа

Ивайло Дончев, Габриела Чотова

### The Project-Based Approach in Informatics Education – A Real-World Example of Rational Number Arithmetic Library Implementation and Use

Ivaylo Donchev, Gabriela Chotova

**Abstract:** *Project-Based Learning (PBL) has long been a well-known and widely used constructivist teaching method in various fields, in which students learn by taking active part in real-world and personally meaningful projects. When it comes to teaching Computer Science, developing software projects is a natural implementation of this pedagogical method.*

*In this paper, the authors share their experience of adapting a project they have used for several years in undergraduate student training to the high school course in Computer Science. Emphasis has been placed on interactive teaching methods in the classroom and the interdisciplinary connection with mathematics. The achieved results have been analysed.*

**Keywords:** *didactic methods; programming; project-based approach; Computer Science; library; rational numbers; object-oriented.*

## ВЪВЕДЕНИЕ

Съвременната педагогическа наука отчита факта, че младите хора днес, така нареченото Z-поколение, са твърде различни от своите учители и родители. Това е поколението, родено с дигиталните технологии. Естествена част от живота на поколението са интернет, мобилните устройства, виртуалният свят, 3D реалността – неща, с които все още много от възрастните, включително учители, трудно боравят. Утвърдилите се като традиционни методи за обучение изискват ученикът да стои на едно място и пасивно да усвоява информация, до която вече има лесен достъп по всяко време. В резултат учителят губи увереност – той вече не е основният източник на информация. Ученикът иска да действа самостоятелно, да му се има доверие, но учителят никога не е бил учен как да управлява подобен процес и се страхува от него. Ситуацията предполага глобално преразглеждане на методите на обучение в училище, така че да се осигурят ефективна комуникация и взаимно обогатяване и на двете страни – ученици и учители [8]. В съзвучие с международната практика през последните години България също предприе стъпки към реформа в образованието.

Вдъхновени от факта, че работата по реални проекти силно мотивира студентите, решихме да пренесем и адаптираме част от заданията за гимназисти. Един от най-лесно адаптираните проекти, който представяме тук, беше аритметиката с рационални числа (обикновени дроби). За учениците дробите, които се изучават в пети клас, са дори по-близки, отколкото за студентите, и лесно може да се направи междупредметна връзка. Работата по проекта обаче изисква придобиването на много знания за важните концепции и механизми в програмирането и проектирането на софтуерни приложения, както и изграждането на умения за ефективно използване на съвременна среда за разработка и съвременен език за програмиране. Затова именно се разчита на спирало-

виден подход с постепенно надграждане на функционалностите и паралелна работа по няколко проекта.

## **ИНТЕРАКТИВНО И ПРОБЛЕМНО ОРИЕНТИРАНО ОБУЧЕНИЕ**

Педагогическата практика показва, че ефективността на обучението зависи силно от ангажираността на учениците в учебния процес и от това те да виждат пряко приложението на всичко изучавано. За тях е съществено в обозримо бъдеще да осъзнават потребността от информацията, която са принудени да усвояват. Това може да се постигне чрез разглеждане в часовете на реални казуси и решаване на реални (с практическа приложимост) задачи (проблеми) – проблемно ориентирано обучение. То е подходяща среда за сътрудничество при учене. Целта е да се изградят умения за прилагане на наученото в типични и специфични ситуации. Ключово умение е приспособяването на учениците към работа с информационни ресурси. Те трябва да превръщат в приложение получената информация. Важен елемент в процеса е и взаимодействието „ученик–ученик“, което води до развиването на умения за работа в екип. Този метод е най-ефективен при работа в екип, тъй като мисленето и обмяната на идеи и опит в сътрудничество гарантират, че и най-сложният проблем може да бъде решен. Това предполага не само съвместна работата по конкретен проект, но и помощ по време на лабораторна работа в час от ученик, когато негов колега партньор среща трудности, а учителят в това време помага на друг. Спецификата на подхода предполага, че учениците имат достатъчно знания и умения, за да подходят към проблема, а в процеса на разрешаването му те доразвиват уменията си за решаване на проблеми и се подготвят.

Добрата практика изисква ученик и учител съвместно да правят анализ на проблемите и обективно да оценяват работата си. Така диалогът като интерактивен метод е важен елемент в съвременното обучение. Участието на всички в процеса на обучение трябва да е активно. Много по-добри резултати се постигат, когато от всеки ученик се изисква активно да прави заключения и да прилага знанията си в практиката.

Учениците учат ефективно чрез изследване. „Обучението не е спортно състезание, в което учениците са зрители. Те не трябва само да стоят в клас, да слушат учителя, да запомнят неговите думи и да отговарят на въпросите му. Те трябва да дискутират за това, което изучават; да го свързват със своя личен опит, да го прилагат в ежедневието си. Правейки това, те изучават част от себе си“ [1].

## **ХАРАКТЕРИСТИКИ НА ПРОЕКТНИЯ ПОДХОД В ОБУЧЕНИЕТО ПО ИНФОРМАТИКА**

Проектно ориентираното обучение е въведено е през 20-те години на XX век в САЩ [9]. Известно е и като „учене чрез правене“. Залага на идеята за активно обучение чрез целенасочена дейност на ученика, съобразена с неговите интереси и познания. Този метод на обучение се прилага в редица области, но е много подходящ за компютърните специалности. Залага се на сътрудничеството за постигането на резултати. На учещите се дават задачи, базирани на факти, проблем за разрешаване или конкретна цел, до която те трябва да достигнат. За да се справят с проекта, обучаемите трябва да търсят допълнителна информация, знанията от други учебни предмети, уменията и опита си и така да откриват нови за тях области. Резултатът е, че те придобиват нови знания и опит в процеса на работа по проекта. В основата на метода са екипната работа и акцентът върху мотивацията и креативността.

Учителят играе ролята както на ръководител, така и на помощник и партньор. Това насърчава изграждането на умения за вземане на информирани решения и поемане на отговорност. Учителят трябва да осигури на учениците си достъп до подходящи ресурси и да им помага в избора на средства и методи за разработката. Той съдейства на учениците си да влязат в ролята на изследователи по съответния проблем. Негова задача е да диагностицира откритите проблеми и да помага в случай на необходимост. Той може да се включва активно в ученическите екипи не само по време на учебните занятия. При колективната работа трудност е оценяването на учениците въз основа на приноса на всеки от тях.

Като силни страни и основни характеристики на подхода в [8, стр. 50] са изброени атрактивност, самостоятелност, творчество, свобода, работа в сътрудничество, създаване на автентичен продукт. Авторката предлага собствена реализация на подхода в синергия с изследователския подход, който паралелно развива важни нетехнически умения и дава безспорно добри резултати в изграждане на критично и аналитично мислене, развиване на способности за абстрактно мислене, обобщение, оценяване, изводи и пренасянето им върху нови проблемни ситуации.

Съществен за успеха на обучението е изборът на конкретните проекти. Те трябва да са предизвикателни за учениците и да обхващат атрактивни за тях проблеми, но нивото трябва да е съобразено с техните възможности, така че целите да са постижими. Затова студентските проекти, които използваме, имат нужда от адаптация.

Резултат от проекта трябва да бъде автентичен, използваем продукт, следствие от изследователска и творческа дейност. Разработката трябва да включва проучване на литература (софтуерна документация) и експерименти (с новите възможности на езиците).

Подходящи за обучението по информатика са интердисциплинарните проекти, съчетаващи природни науки, математика, езици. Така ще се държат в готовност знания от различни учебни дисциплини, както и такива, които се придобиват в извънучилищна среда.

Практиката ни сочи, че най-добри резултати се постигат, когато проектите се възлагат на малки екипи – от 2 до 4 ученици. Групата сама разпределя ролите, избира подходящи помощни инструменти, но носи колективна отговорност за кода и спазването на графика.

Спецификата на обучението по информатика е такава, че при възлагането на проекта учениците не познават изцяло възможностите на езика и средата за програмиране, както и методите за анализ и проектиране на софтуер. Естествено е да се възприеме спираловиден подход, при който понятията се поднасят за първи път без пълно изчерпване на техните характеристики. На по-късен етап, след като знанията и практическият опит на учениците са достигнали определено ниво, се прави ново връщане към същото понятие и на базата на известните до момента свойства то се обогатява и разширява. Обемът на поднесения учебен материал на даден етап от обучението (при поредното завъртане на спиралата) се определя от възрастовите особености, знанията и жизнения опит на учениците. Много е важна и „хоризонталната“ организация при поднасяне на различните понятия, т.е. връзките, които съществуват между различните понятия, поднасяни спираловидно, да се определят така, че на даден етап от завъртане на спиралата учебното съдържание да има завършен вид [2].

Такъв подход е заложен още в учебните програми за III и IV клас по предмета „Компютърно моделиране“ [3, 4], в който се правят първите стъпки в програмирането. Положителна е промяната, заложена в учебните програми на новите учебни предмети „Компютърно моделиране и информационни технологии“ за V, VI и VII клас [5, 6, 7], където се прави преход от блоково програмиране към програмиране на скриптови езици и се избягва тригодишното прекъсване в обучението по програмиране. Очакваните резултати са по-ранно усвояване на много понятия, които след това се ползват по предмета „Информатика“ в VIII клас. Опитът ни в извънкласната работа с ученици показва бързо осъществяване на връзката след преминато обучение по блоково програмиране (Scratch).

## **ПРОЕКТ „АРИТМЕТИКА С РАЦИОНАЛНИ ЧИСЛА“**

Училищният курс по информатика в VIII клас е с твърде ограничен хорариум, за да има време за работа по голям проект. Вместо това развиваме паралелно няколко малки и свързани с ежедневието на учениците проекта.

Един такъв проект е свързан с решаването на квадратни уравнения, които също се изучават в VIII клас по математика. Подходящо е да се започне с конзолно приложение при изучаване на разклонени алгоритми по информатика. Уместно е използването на помощен инструмент за визуализация, например [10]. С придобиването на знания за графичния интерфейс се преминава към разработката на десктоп приложение (WPF или Windows Forms). Алгоритъмът от конзолното приложение се оформя като функция, добавят се проверка за коректност на данните и обработка на изключения.

Подобен план се прилага и при моделирането на йерархии от класове, представящи хора, ученици, учители, библиотеки с книги, геометрични фигури, животни, и интегрирането им в полезни потребителски програми с възможност за извършване на CRUD-операции.

Ще се спрем по-подробно на проекта за аритметика с рационални числа. Учениците се запознават с тях в V клас. Реализацията на проекта започва с опресняване на знанията им за понятието „рационално число“, представяно чрез отношението между две числа  $a$  и  $b$ . Рационалните числа най-често се записват като обикновени дроби във вида  $a/b$ , където  $a$  и  $b$  са цели числа и  $b$  е различно от нула, или като десетични дроби. Числото  $a$  в обикновената дроб се нарича числител, а числото  $b$  – знаменател. Когато числителят на дробта е по-малък от знаменателя, тя се нарича правилна дроб. Когато числителят е по-голям от знаменателя, дробта е неправилна. Естествено възниква въпросът как тези числа могат да се представят в програмите? Предоставя ли езикът вграден тип за тях, подобен на реалните типове `double`, `float`, `decimal`? Първата задача е да създадем такъв тип, който освен да представя данните, трябва и да предоставя операции за манипулирането им. Удачно е този тип да се опише в проект от вида `Class Library`, за да може да се ползва (като `DLL`) в други приложения и проекти. Дефинирайки клас с целочислени полета за числителя и знаменателя и нужните конструктори, учениците скоро забелязват, че едно и също число може да се представи по няколко начина. Стига се до идеята за единствено представяне като несъкратима дроб. Съкращаването изисква реализация на алгоритъма на Евклид за най-големия общ делител (НОД):

```
private static int GCD(int a, int b)    // Н. О. Д.
{
    while (a != b)
    {
        if (a > b) a -= b;
        else b -= a;
    }
    return a;
}
```

Освен чисто математическото понятие (НОД) трябва да се коментира какво е статичен метод; по какво се различават статичните от инстантните (обикновените) методи; как се предава информация към функция. Помага аналогията със системния клас `Math` и неговите статични методи, които често използват – `Math.Pow()`, `Math.Abs()`, `Math.Sqrt()`, които също се извикват, без да е създаден обект от съответния клас.

Първоначално операциите „събиране“, „изваждане“, „умножение“ и „деление“ реализираме с обикновени методи, макар и извикването им да изглежда странно:

```
Rat a = new Rat(2, 4);
Rat b = new Rat(3, 9);
Console.WriteLine($"{a} + {b} = {a.SumRat(b)}");
Console.WriteLine($"{a} - {b} = {a.SubRat(b)}");
```

На екрана се извежда текстът:

$1/2 + 1/3 = 5/6$

$1/2 - 1/3 = 1/6$

Конструкторът автоматично съкращава дробите, извиквайки помощната функция `Normalize()`. Имплементирането на методите изисква напомняне на формулите, по които се извършват аритметичните действия с дроби. Опитът ни от обучението на студенти показва, че трудности се срещат при разбирането на механизма на предаването на данни към функциите чрез параметри и аргументи (формални и фактически параметри). Затова при учениците започваме с най-простото предаване – по стойност.

```
public Rat SumRat(Rat r) =>
    new Rat(this.num * r.den + r.num * this.den, this.den * r.den);
```

```

    public Rat SubRat(Rat r) =>
    new Rat(this.num * r.den - r.num * this.den, this.den * r.den);
    public Rat MultRat(Rat r) =>
    new Rat(this.num * r.num, this.den * r.den);
    public Rat QuotRat(Rat r) =>
    new Rat(this.num * r.den, this.den * r.num);

```

На този етап трябва да е изчистено разбирането за понятията „клас“ и „обект“. Обяснено е какви са разликите и връзките между тях. Въвежда се и се обяснява механизмът на капсулиране на класа (чрез private компоненти).

За да се превърне библиотечният клас `Rat` в пълноправен тип данни, е необходимо да реализираме аритметичните операции с операторни функции (да се ползват знаците за операции `+`, `-`, `*`, `/`), да добавим релационни операции и операции за преобразуване до и от реален тип (`double`). Ето кодът само на някои от тях:

```

public static Rat operator +(Rat r1, Rat r2) =>
    new Rat(r1.num * r2.den + r2.num * r1.den, r1.den * r2.den);
public static Rat operator -(Rat r1, Rat r2) =>
    new Rat(r1.num * r2.den - r2.num * r1.den, r1.den * r2.den);
public static explicit operator double(Rat r) =>
    (double)r.num / r.den; // from Rat to double
public static explicit operator Rat(double d) =>
    new Rat(d); //from double to Rat

```

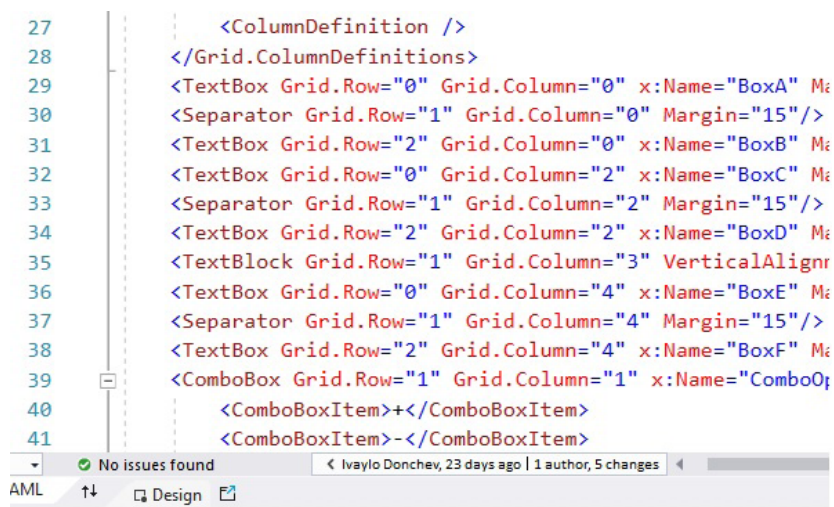
Учениците се убеждават в полезността на разработената от тях аритметична библиотека:

```

Console.WriteLine($"{a} * {b} = {a * b}");
Console.WriteLine($"{a} / {b} = {a / b}\n");
a = new Rat(-0.25); // Конструирание от десетична дроб
Console.WriteLine($"a = {a}"); // a = -1/4
var c = (double)a / 2; // c е от тип double
Console.WriteLine($"c = {c}"); // c = -0.125

```

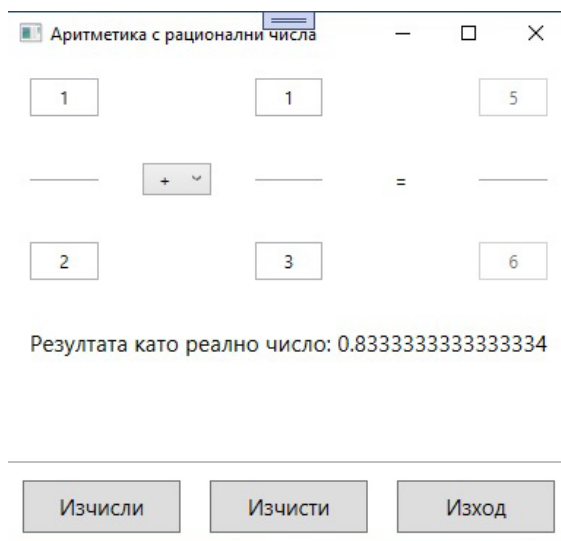
С помощта на тази библиотека може да се премине към разработката на WPF-приложение за .NET. Проектирането и изграждането на добър интерфейс се оказва предизвикателство, което затруднява учениците – изисква се не само подреждане на контроли във формата, а и повече от базови познания по езика XAML. Помощта на учителя при избора на точните контроли и настройката на техните характеристики е задължителна.



Фиг. 1. Извадка от xaml файла във Visual Studio



В крайна сметка се получава съвременен и функционален интерфейс, който ще изглежда добре на всякакви платформи и устройства с всякакви размери (фиг. 2.).



Фиг. 2. Прозорец на приложението

Акцент при разработката на десктоп приложението са валидирането на въведените данни и обработката на изключения. Изчисленията се извършват от разработения преди това клас `Rat`. Ето извадка от кода при натискане на бутона „Изчисли“.

```
Rat arg1, arg2, result;
ValidateBoxes();
try
{
    arg1 = new Rat(int.Parse(BoxA.Text), int.Parse(BoxB.Text));
    arg2 = new Rat(int.Parse(BoxC.Text), int.Parse(BoxD.Text));
    result = new Rat();
    var (n, d) = arg1;    // deconstruction
    ...
    switch (ComboOperations.Text) {
        case "+": result = arg1 + arg2; break;
        case "-": result = arg1 - arg2; break;
```

Благодарение на функционалността на библиотеката изчисленията изглеждат просто и интуитивно, въпреки че зад тях стои доста работа.

Сериозна част от софтуерното производство заема тестването на продуктите. Нашият подход разчита, първо, всеки сам да тества частта от кода (клас, метод, обработчик на събитие), която е разработил, и след това отделен член на екипа (тестер програмист) да тества цялостния проект. Не се разработват специални юнит и интеграционни тестове, защото те са налични само в Enterprise версията на Visual Studio. В резултат на този процес се откриват множество неточности и необработени ситуации, което в нашия случай доведе до 30% увеличение на кода, предимно от обработката на изключения и подобряване на интерфейса.

## ЗАКЛЮЧЕНИЕ

Обучението по информатика винаги е било предизвикателство. Трудностите произлизат както от непрекъснатото развитие на технологиите, така и от променящия се психологически

профил на учениците. Експериментирането с нови подходи, средства и инструменти е част от работата на всеки педагог. Отчитайки съвременните изисквания, прилаганият от нас подход обединява идеи и добри практики от класическите подходи с използването на иновативни инструменти. Съществен елемент на подхода е мотивацията чрез сблъскване на учениците с реален проблем; спираловидното надграждане на разработваните приложения при усвояване на нови техники и интегриране в обучението на методи от процеса на софтуерното производство, какъвто е например програмирането по двойки.

Наблюденията ни показват, че подходът поражда трайни знания и дълбоко разбиране на основните концепции на обектно ориентираното програмиране. За следващата учебна година се планира реализацията на библиотеката и приложенията на езика Java.

## ЛИТЕРАТУРА

[1] **Иванов, И.** Интерактивни методи на обучение, Юбилейна научна конференция с международно участие 50 години ДИПКУ – Варна на тема: “Образование и квалификация на педагогическите кадри – развитие и проекции през XXI век”, 2005. // **Ivanov, I.** Interaktivni metodi na obuchenie, Yubileyna nauchna konferentsia s mezhdunarodno uchastie 50 godini DIPKU – Varna na tema: “Obrazovanie i kvalifikatsia na pedagogicheskite kadri – razvitie i proektsii prez XXI vek”, 2005.

[2] **Йовчева, Б.** Спираловидно обучение по програмиране на 10–11-годишни деца (на базата на езика ++C), Математика и математическо образование, 2007 // **Yovcheva, B.** Spiralovidno obuchenie po programirane na 10–11-godishni detsa (na bazata na ezika ++C), Matematika i matematicheskoto obrazovanie, 2007

[3] **МОН**, Учебна програма по компютърно моделиране за III клас, 2018, URL: [https://www.mon.bg/upload/12205/UP\\_KM\\_3kl.pdf](https://www.mon.bg/upload/12205/UP_KM_3kl.pdf) // **МОН**, Uchebna programa po kompyutarno modelirane za III klas, 2018, URL: [https://www.mon.bg/upload/12205/UP\\_KM\\_3kl.pdf](https://www.mon.bg/upload/12205/UP_KM_3kl.pdf)

[4] **МОН**, Учебна програма по компютърно моделиране за IV клас, 2019, URL: [https://www.mon.bg/upload/13767/UP9\\_KM\\_ZP\\_4kl.pdf](https://www.mon.bg/upload/13767/UP9_KM_ZP_4kl.pdf) // **МОН**, Uchebna programa po kompyutarno modelirane za IV klas, 2019, URL: [https://www.mon.bg/upload/13767/UP9\\_KM\\_ZP\\_4kl.pdf](https://www.mon.bg/upload/13767/UP9_KM_ZP_4kl.pdf)

[5] **МОН**, Учебна програма по компютърно моделиране и информационни технологии за V клас, 2020, URL: [https://www.mon.bg/upload/24186/UP\\_PC-modelirane-5kl.pdf](https://www.mon.bg/upload/24186/UP_PC-modelirane-5kl.pdf) // **МОН**, Uchebna programa po kompyutarno modelirane i informatsionni tehnologii za V klas, 2020, URL: [https://www.mon.bg/upload/24186/UP\\_PC-modelirane-5kl.pdf](https://www.mon.bg/upload/24186/UP_PC-modelirane-5kl.pdf)

[6] **МОН**, Учебна програма по компютърно моделиране и информационни технологии за VI клас, 2020, URL: [https://www.mon.bg/upload/24187/UP\\_PC-modelirane-6kl.pdf](https://www.mon.bg/upload/24187/UP_PC-modelirane-6kl.pdf) // **МОН**, Uchebna programa po kompyutarno modelirane i informatsionni tehnologii za VI klas, 2020, URL: [https://www.mon.bg/upload/24187/UP\\_PC-modelirane-6kl.pdf](https://www.mon.bg/upload/24187/UP_PC-modelirane-6kl.pdf)

[7] **МОН**, Учебна програма по компютърно моделиране и информационни технологии за VII клас, 2020, URL: [https://www.mon.bg/upload/24188/UP\\_PC-modelirane-7kl.pdf](https://www.mon.bg/upload/24188/UP_PC-modelirane-7kl.pdf) // **МОН**, Uchebna programa po kompyutarno modelirane i informatsionni tehnologii za VII klas, 2020, URL: [https://www.mon.bg/upload/24188/UP\\_PC-modelirane-7kl.pdf](https://www.mon.bg/upload/24188/UP_PC-modelirane-7kl.pdf)

[8] **Николова, Н.** Проектно-изследователски подход при преподаването на информатика и информационни технологии, дисертационен труд за придобиване на ОНС „доктор“, Софийски университет „Св. Кл. Охридски“, Факултет по математика и информатика, 2016 // **Nikolova, N.** Proektno-izsledovateliski podhod pri prepodavaneto na informatika i informatsionni tehnologii, disertatsionen trud za pridobivane na ONS „doktor“, Sofiyski universitet „Sv. Kl. Ohridski“, Fakultet po matematika i informatika, 2016

[9] **Kilpatrick, W. H.** Philosophy of education, Macmillan, New York, 1951 // **Kilpatrick, W. H.** Philosophy of education, Macmillan, New York, 1951

[10] **Rowe, J.** Super Mario Quadratics, URL: <https://teacher.desmos.com/activitybuilder/custom/5c7614041509d870d4838bfd> // **Rowe, J.** Super Mario Quadratics, URL: <https://teacher.desmos.com/activitybuilder/custom/5c7614041509d870d4838bfd>

## **ИНФОРМАЦИЯ ЗА АВТОРИТЕ**

---

Ивайло Дончев – доцент, доктор, факултет „Математика и информатика“, Великотърновски университет „Св. св. Кирил и Методий“, e-mail: i.donchev@abv.bg

Габриела Чотова – докторант, специалност „Методика на обучението по информатика и информационни технологии“, факултет „Математика и информатика“, Великотърновски университет „Св. св. Кирил и Методий“, e-mail: gabriela.chotova@gmail.com

## **ABOUT THE AUTHORS**

Ivaylo Donchev – Associate Professor, PhD, Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria, E-mail: i.donchev@abv.bg

Gabriela Chotova – PhD Student in Informatics and IT Teaching Methodology, Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria, E-mail: gabriela.chotova@gmail.com