



ЮГОЗАПАДЕН УНИВЕРСИТЕТ „НЕОФИТ РИЛСКИ“
ПРИРОДОМАТЕМАТИЧЕСКИ ФАКУЛТЕТ
КАТЕДРА „ИНФОРМАТИКА“

Росица Николаева Георгиева

**ИГРОВО-БАЗИРАНО ОБУЧЕНИЕ ЗА РАЗВИТИЕ НА
АЛГОРИТМИЧНО МИСЛЕНЕ В УЧИЛИЩНОТО
ОБРАЗОВАНИЕ**

АВТОРЕФЕРАТ

НА ДИСЕРТАЦИОНЕН ТРУД
ЗА ПРИСЪЖДАНЕ НА ОБРАЗОВАТЕЛНАТА И НАУЧНА СТЕПЕН
„ДОКТОР“
ОБЛАСТ НА ВИСШЕ ОБРАЗОВАНИЕ: 1. ПЕДАГОГИЧЕСКИ НАУКИ
ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ: 1.3. ПЕДАГОГИКА НА
ОБУЧЕНИЕТО ПО ...
ДОКТОРСКА ПРОГРАМА „МЕТОДИКА НА ОБУЧЕНИЕТО ПО
ИНФОРМАТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ“

Научен ръководител:
Проф. д-р Даниела Дурева-Тупарова

Благоевград, 2023 г.

Дисертационният труд “Игрово-базирано обучение за развитие на алгоритмично мислене в училищното образование“ съдържа 166 страници основен текст в отделно тяло и Приложения в 23 страници. Основният текст е илюстриран със 101 фигури и 16 таблици. Използваната литература включва 103 източници, от които 56 на кирилица, 47 – на латиница.

Дисертационният труд е структуриран в увод, четири глави, заключение, използвани библиографски източници. Приложенията включват използван инструментариум по време на обучението.

Дисертационният труд е обсъден и приет на заседание на Катедрен съвет на катедра „Информатика“, Природоматематически факултет към ЮЗУ, „Неофит Рилски“, Благоевград на 25.04.2023 г. и е насрочен за защита на 12.06.2023 г.

Материалите по защитата са на разположение на интересуващите се в катедра „Информатика“.

Съдържание

УВОД	3
ГЛАВА ПЪРВА. ТЕОРЕТИЧНИ ПОСТАНОВКИ	7
1.1. Същност и характеристики на алгоритмичното мислене.....	7
1.2. Дизайн мисленето като подход за обучение.....	9
1.3. Игровизация и игрово-базирано обучение.....	13
1.4. Подходящи игри и среди за програмиране в обучението.....	15
ГЛАВА ВТОРА. ОБУЧЕНИЕТО ПО ИНФОРМАТИКА В БЪЛГАРСКОТО УЧИЛИЩЕ. РОЛЯТА НА ЗАДАЧИТЕ В ОБУЧЕНИЕТО.	18
2.1. Началото	18
2.2. В настоящето	19
2.3. Извънкласните форми по информатика	20
2.4. Задачите в обучението по информатика	21
2.5. Етапи при решаването на задачи по информатика	22
ГЛАВА ТРЕТА. МОДЕЛ ЗА ИГРОВО-БАЗИРАНО ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ, СЪЧЕТАНО С ДИЗАЙН МИСЛЕНЕ ЗА РАЗВИВАНЕ НА АЛГОРИТМИЧНО МИСЛЕНЕ	23
3.1. Описание на модела	23
3.1.1. Обща схема	23
3.1.2. Реализация на модела.....	25
3.1.3. Типовете задачи в контекста на Таксономията на Блум	25
3.2. Система от задачи.....	26
ГЛАВА ЧЕТВЪРТА. ЕМПИРИЧНО ИЗСЛЕДВАНЕ НА МОДЕЛА	27
4.1. Описание на емпиричното изследване.	27
4.2. Статистически методи за анализ на резултатите.	28
4.3. Анализ на резултатите.....	28
4.3.1. Анализ на входните резултати.....	28
4.3.2. Анализ на изходните резултати	30
4.3.3. Анализ на анкетите за установяване на влиянието на системата от задачи при игрово-базираното обучение и мотивацията на учениците.....	33
ЗАКЛЮЧЕНИЕ	35
ПРИНОСИ	37
ПУБЛИКАЦИИ	37
БИБЛИОГРАФИЯ	37

УВОД

Светът в днешно време се нуждае от ново поколение умове с нови начини на мислене и действие. Бъдещите поколения ще бъдат призовани да решават глобални човешки, технологични и природни проблеми, които не са съществували преди. Това налага да се промени и мисленето за света и в частност за образованието. Един съвременен подход, подпомагащ обучението е дизайн мисленето (ДМ). (Тупарова & Георгиева, 2020)

От друга страна, бързото развитие на информационните технологии в последните десетилетия и масовото навлизане на дигитализацията в различни области на обществения живот поставят пред съвременното средно образование предизвикателства, на които трябва да се отговори адекватно и своевременно. (Тупарова & Георгиева, 2019)

Един от начините да се повиши мотивацията на учениците, е игрово-базираното обучение. Основна характеристика на използването на компютърните игрови среди в обучението е предизвикателството да се решават конкретни проблемни ситуации. Интересът, който предизвикват игрите, безспорно оказва влияние върху мотивацията за учене, критичното мислене и решаването на проблеми, практическите умения и самоконтролът.

Наред с това, уменията за планиране, точното, пълно и разбираемо описание на действията, ще помогнат на учениците да развият алгоритмичното си мислене. Използвайки подхода дизайн мислене при игрово-базирано обучение, се стимулират и развиват умения за креативно мислене, особено в ИКТ сферата, поощрява се процеса на разработване на иновации и съдейства за растежа и конкурентоспособността на бъдещите компютърни специалисти. Затова особено подходящо е прилагането на игрово-базирано обучение по програмиране, съчетано с дизайн мислене. (Тупарова & Георгиева, 2020)

В настоящия дисертационен труд е описан модел на игрово-базирано обучение, съчетан с подхода дизайн мислене за развитие на алгоритмичното мислене. Описана е система от задачи, прилагането на която цели усвояване на знания и умения по програмиране, развиване на алгоритмичното мислене и повишаване на дигиталната компетентност на учениците.

Дисертационният труд е разработен и оформен съгласно „Методология и методи на педагогическите изследвания“ (Бижков, 1995).

Тема: Игрово-базирано обучение за развитие на алгоритмично мислене в училищното образование

Обект на изследване са алгоритмичните умения на ученици от 8. до 10. клас в ПГСАГ „Васил Левски“- гр. Благоевград, които не са изучавали учебните предмети „Компютърно моделиране“ и „Информатика“ по учебна програма.

Предмет на изследването е процесът на формиране и развитие на алгоритмични умения на учениците от първи гимназиален етап чрез използване на игрово-базирано обучение по програмиране, съчетано с дизайн мислене като подход за развиване на алгоритмично мислене.

Цел на изследването е чрез игрово-базирано обучение по програмиране, съчетано с дизайн мислене като подход да се развие алгоритмичното мислене на учениците, да се повиши дигиталната им компетентност, както и мотивацията им за учене.

Задачи:

1. Да се направи анализ на тенденциите в обучението по информатика в национален мащаб:

- преглед на литературата по въпроса;
- проучване на подходящи платформи, среди и езици за игрово-базирано обучение по програмиране;
- игрово-базираното обучение и неговата роля при усвояване на знания, умения и компетенции;
- приложение на дизайн мислене като подход за обучение в България и света;

2. Да се направи анализ на състоянието на обучението по информатика в задължителна и извънкласна форми в национален мащаб:

- проучване и анализ на задължителното обучение по информатика в страната;
- проучване на обучението по програмиране в извънкласни форми;
- ролята и мястото на задачите в обучението по програмиране.

3. Да се проучи възможността за повишаване на дигиталните умения и развиване на алгоритмичното мислене на учениците с помощта на игрово базирано обучение и дизайн мисленето като иновативен подход;

4. Да се разработи и модел за игрово-базирано обучение по програмиране за развиване на алгоритмичното мислене, съчетан с подхода дизайн мислене;

5. Да се изследват резултатите от предложения модел по отношение на дигиталната компетентност и развитието на алгоритмично мислене на учениците, както и въздействието върху мотивацията за учене.

6. Чрез емпирично изследване да се апробира разработеният модел.

Изследователски въпроси

Игрово-базираното обучение, съчетано с дизайн мислене в обучение по програмиране, ще доведе ли до развиване на алгоритмичното мислене на учениците?

Възможно ли е да се повиши дигиталната компетентност на учениците при обучение по програмиране, базирано на игри?

Ще се повиши ли мотивацията за учене и интересът на учениците към програмирането?

Издигната е хипотеза, състояща се от три компоненти:

Използването на предложения модел за игрово-базирано обучение по програмиране съчетан с дизайн мислене в гимназиален етап, ще доведе до:

1. Развиване на алгоритмичното мислене на учениците;
2. Повишаване на общата дигитална компетентност на учениците;
3. Повишаване на интереса и мотивацията на учениците;

Изследователски методи:

- Обзор и анализ на литература, свързана с предмета на изследване;
- Педагогическо изследване: констатиращо, основно и заключително;
- Методи за събиране на емпиричен материал: педагогически експеримент, анкета, тестиране;
- Статистически методи за обработка и анализ на емпиричния материал.

Етапи на изследването:

- Етап (2018-2019 г.) – Проучване на актуалната към момента литература, свързана с предмета на изследване.

- Етап (2019-2020 г.) – Проучване на актуалната към момента литература, свързана с предмета на изследване. Разработване на модел за игрово-базирано обучение.

- Етап (2021-2022 г.) – Проучване на актуалната към момента литература, свързана с предмета на изследване. Апробиране на разработения модел за игрово-базирано обучение. Анкетирание на учениците, участващи в изследването, с цел установяване мнението им по отношение на използването на игрите в обучението по програмиране. Тестване на учениците за установяване на входно нивото и изходящото ниво на знания, Статистическа

обработка и анализ на резултатите от емпиричното педагогическо изследване.

Инструментарium:

- Анкета за определяне мнението и отношението на учениците
- Тест и практическа задача за проверка и оценка на знанията и уменията на учениците
- За статистическата обработка на емпиричните данни основно са използвани софтуерните пакети IBM SPSS Statistics и JASP.

Дисертационният труд е структуриран в съответствие с поставените задачи. Състои се от увод, четири глави, заключение, библиография, списък с публикациите, свързани с дисертацията, приложения, списъци на таблици, задачи и фигури.

ГЛАВА ПЪРВА. ТЕОРЕТИЧНИ ПОСТАНОВКИ

1.1. Същност и характеристики на алгоритмичното мислене

В литературата не съществува единно понятие за „алгоритмично мислене“. Трудности са породени от различни фактори като един от тях е, че често понятието „алгоритмично“ се заменя с понятия като „логическо“, „инженерно“, „математическо“, „абстрактно“, „изчислително“ и други.

От друга страна проблема с дефинирането на понятието е свързан с разбирането за същността му. Проведено проучване (Терзиева, 2010, стр. 479-486) за разбиране на понятието алгоритмично мислене показва липса на единно разбиране относно съдържанието на понятието алгоритмично мислене.

Според Терзиева, алгоритмичното мислене е „начин на мислене, осигуряващ решение на дадена задача чрез последователност от елементарни действия“ (Терзиева, 2012, стр. 31). Същият изследовател допълва, че „алгоритмично мислене се състои от широк диапазон от умения и се влияе от много други когнитивни фактори: абстрактно и логическо мислене, структурно мислене, способност за решаване на проблеми и креативност“ (Терзиева, 2012, стр. 31).

В научната литература се срещат и други определения на алгоритмичното мислене (Табакова-Комсалова, 2018), (Наков & Колектив, 2017). За целите на настоящето изследване се възприема формулировката за алгоритмично мислене на (Терзиева, 2012).

Основните компоненти на алгоритмичното мислене са систематизирани и представени на следващата фигура I-1.



Фигура I-1. Компоненти на алгоритмичното мислене

Днес ползите от алгоритмичното мислене са осъзнати не само от сектора на ИКТ, но и от образователната система в различни страни, включително и в България.

Развитието на алгоритмичното мислене в най-ранна възраст (ученическа) е важно, защото „позволява на учениците да решават проблеми, да вникват в системите за проектиране, както и да разберат силата и границите на човешкото мислене и интелигентността на машината. Това е умение, което създава предпоставка всички ученици да извършват реална самооценка и да развиват и усъвършенстват компетенциите си“ (Табакова-Комсалова, 2018, стр. 26). Формирането на алгоритмичното мислене е „нелека задача, изискваща прецизен дидактически подход“ (Табакова-Комсалова, 2018, стр. 21).

Една от основните дидактически цели на училищното обучение в съвременното българско училище е развитието на алгоритмично мислене. Още в 3. (МОН, 2018) и 4. (МОН, 2019) клас, развитието на алгоритмично мислене е заложено в програмата по компютърно моделиране, като усъвършенстването му продължава в 5.-7. клас, в заложената учебна програма по компютърно моделиране и информационни технологии (МОН, 2020). Развитието на алгоритмичното мислене в българските училища е застъпено както в общо образователните учебните програми (МОН, 2020) по компютърно моделиране и информационни технологии, така и в учебни програми за профилирана подготовка (информатика и информационни технологии) (МОН, 2018).

Алгоритмичното мислене позволява да се рационализира пътя към решение на една задача и се проявява в умението обучаемите да използват индуктивни и дедуктивни методи и мисловни операции при анализ на проблеми от различни области, използват методи за формализиране на решения на задача до запис на алгоритмичен език“ (Герзиева, 2012, стр. 34).

Един от познатите ефективни начини за развитие на алгоритмичното мислене при учениците от началното училище е чрез „допълнителните часове по информационни технологии в избираеми часове, клубове по интереси, школи по информатика и други. В хода на изучаване на компютърно програмиране и информационни и комуникационни технологии се придобиват знания и умения за съставянето на алгоритми и използването им при решаване на голям клас задачи за програмиране“ (Табакова-Комсалова, 2018, стр. 21).

Пред българската образователна система се очертава необходимостта от ускоряване на процеса за формиране на алгоритмичното мислене на учениците, поради съвременните условия на динамично развитие на

дигитализацията и автоматизация във всички сфери на живота, включително и образованието.

Развитието на алгоритмично мислене подпомага не само решаването на учебните задачи, свързани с информатика и ИТ, но и решаването на други предизвикателства с учебен, трудов и житейски характер. Споделя се мнението за „неправилност на становището, че алгоритмичният стил на мислене и алгоритмите се присъщи само за работа с изчислителна техника, защото чрез дигиталните устройства се решават задачи от абсолютно всички сфери на живота“ (Табакова-Комсалова, 2018, стр. 20).

1.2. Дизайн мисленето като подход за обучение

Произходът на дизайн мисленето (Design Thinking) (ДМ) от научна гледна точка може да се проследи от края на 60-те години на 20 век, но понятието „дизайн мислене“ се появява в края на 80-те и 90-те, когато Ролф Фасте започва да преподава дизайн мисленето като „метод за креативно действие“ в Станфордския университет. ДМ е адаптирано и приложено в бизнеса от Дейвид Кели от Станфорд, основател на консултантската компания за дизайн IDEO през 1991 г. В бизнес речника, методът навлиза масово след популяризирането му от Тим Браун от IDEO (Inovation Starter, 2015). (Табл. 1)

Таблица 1. Модел на Станфорд (Naiman, 2019), (Inovation Starter, 2015), (Standford, 2018)

Практики	Принципи	Етапи	
		Кратък процес	Разширен процес
Развитие на дълбоко съпричастно разбиране на потребностите на потребителя	Discovery– Срещам ново предизвикателство. Как да го преодоляя?	Наблюдение и изучаване на поведението и опита на потребителите, изследване и дефиниране на проблем и гледна точка на базата на емпатия	Наблюдение и изучаване на поведението и опита на потребителите, изследване и дефиниране на проблем и гледна точка на базата на емпатия
Формиране на разнородни екипи	Interpretation– Какво научих и как да го интерпретирам?	Генериране на много идеи в огромно количество за кратък период от време	Генериране на много идеи в огромно количество за кратък период от време
Разговори на базата на диалог	Ideation– Виждам възможност, как да я оформя в идея?	Подбор и класифициране на идеите	Подбор и класифициране на идеите

Генериране на решения чрез експериментирание	Experimentation – Имам идея, как да я построя?	Прототипиране – продукт, за който бързо да има обратна връзка от потребителите	Прототипиране – продукт, за който бързо да има обратна връзка от потребителите
Използването на структуриран и улеснен процес	Evolution – Опитам нещо ново, как да го развия?	Обратна връзка – как да се подобри прототипа, ако се налага	Обратна връзка – как да се подобри прототипа, ако се налага
			Тестване дали идеята работи
			Изучаване и подобрене – отново на база обратна връзка

Днес в литературата съществуват различни дефиниции за понятието. Според (IDEO, 2008). „дизайн мисленето е ориентиран към човека подход към иновациите, който черпи от инструментариума на дизайнера за интегриране на нуждите на хората, възможностите на технологиите и изискванията за успех на бизнеса“. Пак в (IDEO, 2008) се допълва, че „дизайн мисленето е начин за решаване на проблеми чрез креативност, процес за креативно решаване на проблеми, който стимулира най-важното качество в днешния свят - постоянно да променяш средата около себе си, като я подобряваш, воден от своето любопитство и творческа увереност“.

“Дизайн мисленето се основава на логиката, въображението и интуицията и не се фокусира върху проблема, а е насочено към решението и е ориентирано към действието. Голяма част от концепцията за дизайн мисленето включва съпричастност към тези, за които се създават новите продукти” (Naiman, 2019).

По своята същност дизайн мисленето е кръгов процес, целта на който е да се усъвършенстват и развият продуктите, услугите, процесите и бизнес моделите (Inovation Starter, 2015).

Въпреки, че ДМ е бизнес методология, днес намира широко приложение в публичния, неправителствен, здравен и образователен сектор. В последното десетилетие все повече образователни институции се обръщат към ДМ като метод подпомагащ ефективното обучение.

ДМ в образованието може да се приложи в няколко ключови направления представени на следващата фигура I-2.



Фигура I–2. Направления на приложение на ДМ в образованието

В образованието ДМ следва определени етапи и методи, представени на следващата фиг. I-3 (Тупарова & Георгиева, 2020):



Фигура I–3. Методи на обучение в етапите на дизайн мисленето

Към момента са натрупани множество добри практики, свързани с приложението на ДМ в образованието. Един от примерите на световно ниво е D-school или Институт по дизайн към университета Станфорд, основан през

2005 година. За да намерят нов набор от решения за образователната си система, към D-school се създава Prototype Design Camp с цел въвеждане и прилагане на умения за дизайн мислене. Изводите, които споделят организаторите на Prototype Design Camp, е убеждението, че дизайн мисленето като подход за обучение научава учениците и студентите на сътрудничество (Standford, 2018), (Standford, 2019).

Друг такъв лагер е Bootcamp Design Thinking, където чрез поредица от ролеви игри с различни герои, се дава възможност на обучаемите бързо генериране на идеи с възможност за визуализиране и адаптиране на резултатите в реално време (Standford, 2022).

Прилагането на идеите на ДМ не подминава и България. В рамките на международни и регионални проекти се разработват програми за развитие на ДМ в областта на ИКТ. Програма "Дизайн шампиони" - Red Paper Plane е програма за проектно базирано обучение, която използва метода дизайн мислене. Програмата вече се прилага в 120 учебни заведения в 65 населени места в страната от 400 учители. (Red Paper Plane, 2022), (Радославова, 2018).

Друг пример е Академия „Телерик“, която включва програмите „Софтуерна академия“ (софтуерно инженерство), „Училищна академия“ (софтуерно програмиране за ученици на възраст между 14 и 19 години), „Детска академия“ (обучение по програмиране на деца от 10 до 12 години).

Софтуерният университет (<https://softuni.bg/>) е базиран на идеята за иновативен образователен център, който използва модела „учене чрез правене“, като предоставя обучение по програмиране в комбинация с реален практически опит и знания в новите технологии. Към него е и SoftUni Kids – програма, подходяща за ученици от 1. до 6. клас, която се базира на иновативен учебен подход, основан на игри за придобиване на знания и умения в сферата на програмирането (Inno Spark, 2015)

Университетът по библиотекознание и информационни технологии (УниБИТ) също е добър пример за обучение, до голяма степен базирано на дизайн мисленето като методология, като основни приоритети са постигане на високо качество в обучението в сектор ИКТ и тяхното приложение в практиката (УНИБИТ, 2022).

Интерес представлява и проектът „Професионално развитие на учителите за насърчаване на уменията за мислене в областта на дизайна и академичния успех на учениците“ (DTS)¹. Проектът е финансиран от програма Еразъм+ на Европейския съюз и партньори в проекта са Турция, България

¹ <https://dts4teachers.eu/>

(ЮЗУ „Неофит Рилски - Благоевград), Португалия и Италия. Проектът е насочен към ученици, учители и институции в средното образование.

1.3. Игровизация и игрово-базирано обучение

Игровизация е понятието, произлизащо от английската дума за игра (game). Самият термин е измислен през 2002 г. от Ник Пелинг (Nick Pelling), дизайнер на игри. В (Pelling, 2011) авторът казва: „... измислих умишлено грозната дума „игровизация“, с което имах предвид прилагането на подобен на игра ускорен дизайн на потребителски интерфейс, за да направя електронните транзакции едновременно приятни и бързи“.

Авторите (Tuparova, Tuparov, Veleva, & Nikolova, 2018), цитиращи Капп, определят игровизацията като „използване на базирана на играта механика, естетика и игрово мислене, с цел да ангажира хората, да мотивира действията, да насърчава ученето и решаването на проблеми“. Понятието също така може да бъде описано като концепция за въвеждане на игрови техники в различни процеси и области като бизнес, маркетинг и образование. (Trybus, 2015).

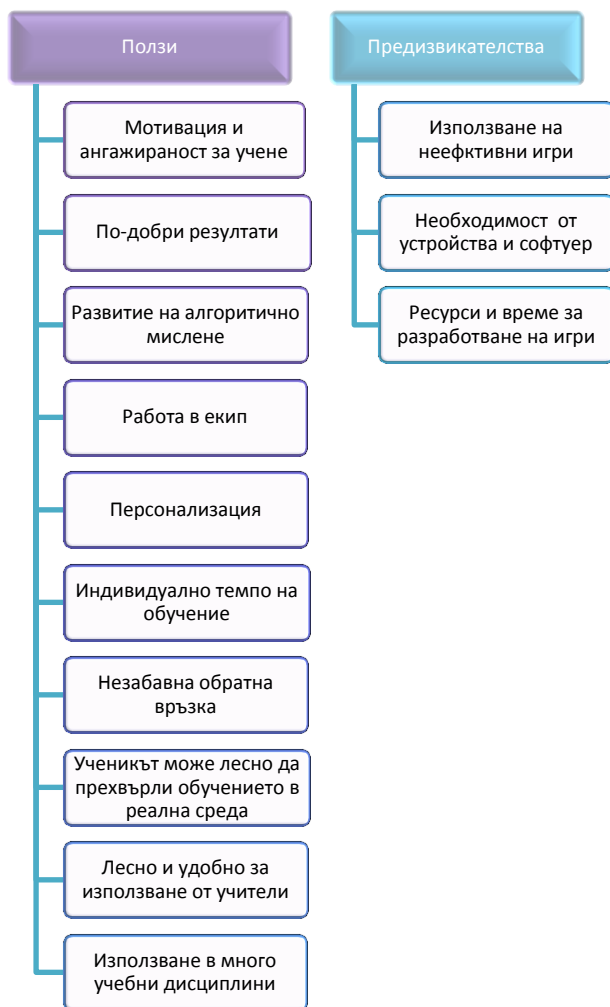
В сферата на образованието, игровизацията се проявява чрез концепцията за игрово-базирано обучение (ИБО). ИБО е средство за придобиване на знания и умения чрез игра, където игровите дейности изискват от играча да решава проблеми и предизвикателства, предоставени за постигане на постижения (Kirriemuir & McFarlane, 2004). Други автори дефинират ИБО като „форма на обучение, насочено към ученика, което използва дигитални игри за образователни цели. Този израз е семантична композиция от двете понятия, игра и учене, с фокус върху ученето“ (Aleksić, Ivanović, Popesco, & Budimac, 2016, p. 355).

Според (Спирова, 2018, стр. 45), ИБО „има два варианта:

- Инструктивистки модел: учениците се учат, играейки образователни игри, зададени и разработени от други (учител);
- Конструктивистки модел: учениците се учат, играейки образователни игри, зададени и разработени от тях самите“.

ИБО предоставя възможност на учителите да включат активно обучение в учебните часове, да насърчават интереса и ангажираността на учениците и да предоставят незабавна обратна връзка за индивидуалното и общо представяне.

Прилагането на ИБО в училищното образование е свързано с много предимства и някои недостатъци както за учениците, така и за учителите систематизирани и представени на следващата фиг. I-4:



Фигура I-4. *Ползи и предизвикателства от ИБО в училищното образование*

ИБО концепцията става все по-популярна във всички степени на образованието, включително и в училищното образование като ефективен начин за ангажиране на ученици в обучението по различни дисциплини. В по-голямата част от представените български научни публикации, интересът по темата за ИБО е насочен главно към използването на образователни компютърни игри и среди в началната степен на образование.

Редица автори правят проучвания по теми, свързани с игрово-базираното обучение. В (Спирова, 2018, стр. 41-46) се разглежда реализацията на някои от основните дидактически принципи чрез игрово-базираното обучение, в (Дурева & Касева, 2011) се очертават затрудненията, които изпитват учителите при обучението, базирано на игри, както и преимуществата на игрово-базираното обучение, в (Tuparova, Tuparov, Veleva, & Nikolova, 2018) е направено изследване на отношението на учителите по информатика и информационни технологии (ИТ) към използването на образователни компютърни игри и игровизацията. В статията на (Tuparov, Keremedchiev, Tuparova, & Stoyanova, 2018) е очертана рамката за идентифициране на характеристиките на игровизацията в отворените среди за електронно обучение и прилагането на игровизацията в дейностите по оценяване, оценка от съученици и самооценка в средата за електронно обучение Moodle. Разгледани са педагогически аспекти и технологичните характеристики на игровизацията в образованието, В статията на (Hakak, et al., 2019), фокусът на авторите е върху игровизацията и възможностите на облачните технологии като един от съвременните начини за модернизиране на учебните модули и изследва приложимостта на игровизацията в различни тематичните области, които могат да се преподават чрез облачната услуга.

1.4. Подходящи игри и среди за програмиране в обучението

В научната литература се срещат различни термини и формулировки на понятия, свързани с приложението на компютърни игри в образователния процес – образователни видео игри, дидактически компютърни игри, сериозни игри, образователни компютърни игри. Образователните игри са онези сериозни игри, чиято цел е предаването на специфични знания и умения в някаква област и предоставяне на нови знания чрез игрови подход“ (Терзиева, Голев, & Ставрев, 2017). В (Дурева & Касева, 2011), образователните компютърни игри са определени „като положителен фактор за развитието на мисленето, паметта и вниманието на детето и формиране на умения за обобщение и класификация, особено при развитието на координацията и моториката в началното образование“.

Образователните игри са мощно средство за подпомагане на традиционното обучение в училище по различните учебни предмети. Особено важна е ролята им при формиране и развитие на алгоритмично мислене.

Основна характеристика на използването на компютърните игрови среди в обучението е предизвикателството да се решават конкретни проблемни ситуации. Ученикът взаимодейства с виртуалната среда на играта,

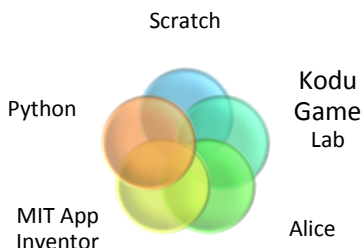
развивайки умения да експериментира, да взема различни решения, да контролира развитието на ситуацията. Интересът, който предизвикват игрите, безспорно оказва влияние върху мотивацията за учене, критичното мислене и решаването на проблеми, практическите умения и самоконтролът (Петров, Вълканова, & Димитров, 2016).

Класификация на средите за игрово-базирано обучение по програмиране

Средите се класифицират в две основни групи (Тупарова & Георгиева, 2019):

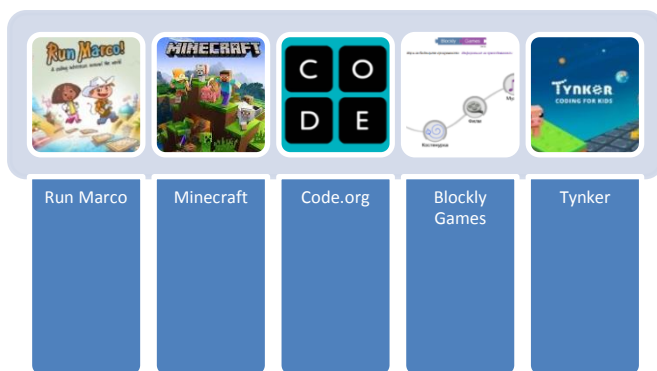
- Среди за програмиране – в които лесно се разработват образователни компютърни игри;
- Среди за програмиране в игрови ситуации.

Едни от най-популярните среди за програмиране за образователни компютърни игри са систематизирани и представени на следващата фиг. I-5:



Фигура I-5. Среди за програмиране за образователни компютърни игри

Най-популярните среди за програмиране в игрови ситуации, използвани често в практиката, са систематизирани и представени на следващата фиг. I-6:



Фигура I-6. Среди за програмиране в игрови ситуации

Ефективния избор на образователни игри/среди е свързан с придържане към определена рамка с няколко ключови критерии, представена на следващата фигура I-7. (Тупарова & Георгиева, 2019)



Фигура I-7. Рамка за сравняване на образователни игри/среди

1. Тип на средата – определя се от това дали е само за програмиране или осигурява програмиране в игрови ситуации.

2. Приложимост в конкретна образователна степен – предучилищна, начална, прогимназиална и гимназиална.

3. Интерфейс – локализация на различни езици, интуитивност при работа със средата.

4. Режим на работа – осигурява се възможност за работа онлайн или офлайн.

5. Разпространение – платено или безплатно.

6. Поддръжка на общностите на учители, родители, ученици.

7. Възможност за работа на различни устройства – като самостоятелно приложение или в приспособяващ се към устройството режим (responsive).

8. Възможност за преминаване към програмиране на друг език за програмиране.

9. Възможност за работа с индивидуално темпо – избор на теми, модули, упражнение и др.

10. Учебно съдържание по програмиране, което се покрива от средата.

Образователните компютърни игри и среди имат потенциала да създадат мотивираща и забавна учебна среда, отговаряща на образователните стандарти и целите на обучението по информатика и информационните технологии в съвременното българско училище.

Изводи: Традиционното обучение в училище все повече отстъпва на новите технологии. В последните години се наблюдава спад в общата мотивация за учене и това налага прилагане на нетрадиционни методи на обучение. От ефективността на прилаганите методи за обучение зависи и ефективността на обучението като цяло. Безспорно, игрово-базираното обучение и дизайн мисленето имат съществено място в обучението на съвременните деца за изграждането на знания, умения и компетенции, необходими за реализацията им в днешно време.

ГЛАВА ВТОРА. ОБУЧЕНИЕТО ПО ИНФОРМАТИКА В БЪЛГАРСКОТО УЧИЛИЩЕ. РОЛЯТА НА ЗАДАЧИТЕ В ОБУЧЕНИЕТО.

2.1. Началото

Изучаването на предмета Информатика в средните училища в България датира от 60-те години, когато в по-големите градове на страната се създават паралелки с разширено изучаване на математика. (Гъров, 2000)

Първите Математически гимназии се създават в началото на 70-те години, където обучението по информатика първоначално се провежда под формата на факултативни часове извън задължителното обучение.

През 1983 г. се осъществява реформа в средното образование, според която в първата степен на обучение до 10. клас се осъществява общото образование на учениците, във втората степен 11. - 12. клас се овладява професия в т. нар. Учебно-професионални комплекси (УПК), а за учениците от Математическите гимназии и някои техникуми е въведена подготовка за професията “Оператор-програмист на ЕИМ”. Същата година в МГ “Акад. К. Попов” - Пловдив се открива и първият в страната компютърен кабинет. Следва период на бързо компютъризиране на училищата. Така в средата на 80-те години информатиката е навлязла под различни форми в средното училище, но е неравномерно застъпена в различните класове. (Гъров, 2000)

През 1986 г. Министерство на народната просвета въвежда като отделен учебен предмет “Информатика” в 10. и 11. клас на средното общообразователно училище с използването на езика Бейсик.

В следващите няколко години се извършват множество промени в учебните планове по информатика. От драстично намаляване на хорариума на

часовете в общозадължителната подготовка, през изучаване на предмета само като СИП или ЗИП, до напълно премахване на часовете по информатика в някои училища. (Дурева, 2003)

Следва период (1994 г. – 1998 г.), в който след въвеждане на типови учебни програми по информатика, предмета се изучава като общозадължителен предмет, СИП или ЗИП в СОУ, СПТУ и Техникумите, а в Математическите гимназии и Икономическите техникуми, обучението по информатика е с профилиращ характер и по-голям годишен хорариум. През 1994 г. се въвежда като задължителен и учебния предмет „Информационни технологии“. (Николова, 2019)

През 1998 г., влиза в сила Националната стратегия по информационни и комуникационни технологии (ИКТ), с която се поставя началото на нова реформа в средното образование. Разработват се методики за изучаването на програмиране в средните училища.

През 2000 г. се приемат Държавните образователни изисквания и стандарти за учебни програми, с които се определят равнищата на общообразователна подготовка на учениците в края на етап и степен на образование, учебното съдържание на общообразователната подготовка, както и изискванията за учебно съдържание с включени общи и специфични ключови групи умения, интегрални и междупредметни области на учебно съдържание в зависимост от спецификата на учебните предмети и на културно-образователните области (в частност Математика, информатика и информационни технологии, Приложение № 3). (МОН, 2000)

С Националната стратегия за въвеждане на ИКТ в българските училища, се поставят основите на интегрирането и въвеждането на информационните и комуникационните технологии в образованието. (МОН, 2005).

Следва Стратегия за ефективно прилагане на ИКТ в образованието и науката на Република България (2014-2020), чиято основна цел „е да осигури равен и гъвкав достъп до образование и научна информация по всяко време и от всяко място - от стационарен компютър, лаптоп, таблет, мобилен телефон. За първи път се създава единна информационна среда, обслужваща училищното образование, висшето образование и науката (МОН, 2014)

2.2. В настоящето

През 2016 г. влиза в сила Закона за предучилищното и училищното образование (ЗПУО) и нови Държавни образователни стандарти. С Наредба № 4 за учебния план (МОН, 2015) и Наредба № 5 за общообразователната

подготовка (ЦИОО -МОН, 2015) се правят съществени промени в учебния план и учебното съдържание по информатика и информационни технологии. Предметът „Информатика“ в първи гимназиален етап се изучава само в 8. клас като общообразователен, и то само в профилирани паралелки с профили „Математически“, „Софтуерни и хардуерни науки“, „Икономическо развитие“, „Природни науки“. В 9. и 10. клас не е заложено изучаване на „Информатика“, а във втори гимназиален етап - в 11. и 12. клас е предвидена профилирана подготовка по предмета в четири модула. (МОН, 2016)

Друга новост в учебните програми след влизането в сила на ЗПУО през 2016 г., е въвеждането на предмета „Компютърно моделиране“ в начален етап на образование - 3. и 4. клас. В учебната програма по предмета са заложени базови компетентности, свързани с дигитални устройства, информация (типове данни), моделиране във визуална среда за блоково програмиране (основни конструкции и алгоритми). В прогимназиален етап, учебният предмет е „Компютърно моделиране и информационни технологии“ и областите на компетентности са компютърни системи, информация, електронна комуникация, създаване на дигитално съдържание с използване възможностите на блоково програмиране. Освен задължителните учебни часове, по Наредба №4 за учебния план (МОН, 2015), на училищата е предоставена възможност да се въведат избираеми учебни часове (ИУЧ) и факултативни учебни часове (ФУЧ).

2.3. Извънкласните форми по информатика

Първите състезания по програмиране датират от началото на 70-те години. Първото Национално състезание е през 1979 г. и се провежда в гр. София. От 1982 г. стартират ежегодните Национални състезания Зимни математически състезания (Русе), Състезания по информатика в рамките на Пролетните конференции на СМБ и Състезания по програмиране на Централната станция на младите техници. От 1985 г. тези състезания прерастват в Национална олимпиада по информатика (НОИ) (Азълов, 2009). През м. Май 1987 г., в София се провежда „Отворено състезание по програмиране“, в което участват 28 ученици от 6 страни. (Кендеров, 2009)

На XXIV Генерална конференция на UNESCO в Париж, 1987 г., представителят на българската делегация акад. Благовест Сендов предлага в Петата главна програма (1988-1989) на UNESCO да се включи точка за основаване на Международна олимпиада по информатика (МОИ) (ММИБ). Така България става една от държавите, основали Международната олимпиада по информатика за ученици. През м. май 1989 г., в гр. Плевен, се провежда

Първата Международна олимпиада по информатика. (Манев, 2009) (Азълов, 1989) (Kenderov & Maneva, 1989) (СМБ). Интересът към тези състезания се увеличава и в страната се създава мрежа от кръжоци и школи за извънкласна дейност по програмиране. В следващите години и досега, ученици от България печелят многобройни призови места в изданията на МОИ. (Гъров, 2000)

С бързото развитие на компютърните науки и все по-широкото им приложение в различни икономически сфери през последните десетилетия, се повиши и интересът към създаването на софтуерни и мултимедийни продукти, уеб базирани приложения, бази от данни. През 2002 г. стартира Националната Олимпиада по Информационни технологии (НОИТ). Ежегодно се провеждат и редица други състезания – Национално състезание по информационни технологии, Национален пролетен турнир по информационни технологии, IT Знайко, Webloz – национално състезание за уеб дизайн, състезания тип „Хакатон“ и други.

След 2016 г., с въвеждането на новите учебни планове, учениците в гимназиален етап, които не се обучават в профилите „Математически“, „Софтуерни и хардуерни науки“, „Икономическо развитие“ и „Природни науки“ не изучават информатика. В задължителните учебни часове се изучава само предметът „Информационни технологии.“

В малка част от училищата, предметът „Информатика“ може да се изучава избираемо или факултативно, но основна алтернатива за изучаването му остават извънкласните форми.

2.4. Задачите в обучението по информатика

Редица автори разглеждат мястото и ролята на задачите по информатика (Асенова, 1990), методическите проблеми при решаването на задачи в училищните курсове по информатика и ИТ (Дурева, 2003), съставят се системи от опорни задачи за подготовката на ученици за участие в олимпиади и състезания по информатика и информационни технологии (Гроздев & Гъров, 2008) (Гъров & Тодорова, 2006) (Гъров, 2004) (Гъров, 2000).

За участие на учениците в горепосочените състезания е безспорно, че обучението по информатика и информационни технологии в учебните часове в училище е крайно недостатъчно. Извънкласните форми все повече се налагат като форма на обучение по програмиране не само за подготовка за състезания, а и като възможност за удовлетворяване интересите на учениците към компютърните науки като цяло. В училищата от 2018 г. стартира проект на МОН „Занимания по интереси. Друга възможност, макар и недостатъчно застъпена, са Центровете за подкрепа за личностно развитие, където ученици

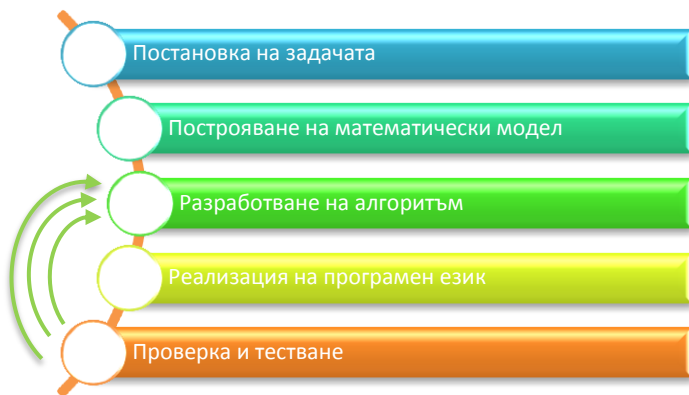
от 2. до 10. клас изучават програмиране (Пловдив, Варна) и информационни технологии (Ловеч, Пловдив, Бургас, Благоевград).

Основните акценти при извънкласните занимания по програмиране са насочени към бързото усвояване на базови знания и прилагането им при решаване на задачи от реалния живот, „създаване на условия за творческо и алгоритмично мислене, задълбочаване и разширяване на информатическата култура, формиране на умения и навици за самостоятелно овладяване на нови знания, създаване на практически умения за конструиране на компютърни програми“ (Гроздев & Гъров, 2007)

Функциите на задачите, които се прилагат при обучението по програмиране в извънкласните форми се обуславят от горепосочените основни акценти. Освен основните функции (методическа, дидактическа, организираща и управляваща (Дурева, 2003), (Гъров, 2010), цитиращи Сманцер, могат да се посочат и някои специфични: формиране на алгоритмичен стил на мислене, усвояване на основите на моделирането, усвояване формализацията при разкриване на взаимовръзката между различните форми на едно и също понятие, явление или процес, задачите са учебна форма на практиката.

2.5. Етапи при решаването на задачи по информатика

Редица автори се обединяват около една обща схема на етапи при решаването на задачи по програмиране (Гъров, 2010) (Гроздев & Гъров, 2007) (фиг. II-1):



Фигура II-1. Етапи при решаване на задачи по програмиране

I Етап – На този етап трябва да бъдат ясно определени учебните цели на задачата, необходимите знания и умения, както и възрастовите възможности на учениците, за да може всеки ученик сам да разбере формулировката ѝ.

II Етап – Определя се мястото на задачата в конкретен клас задачи, сходството ѝ (ако има такова) с решавани вече подобни задачи или се определят вече познати методи за решаване, актуализират се знания по математика.

III Етап – Конструира се общата схема на алгоритъма, определят се подалгоритмите и конструкциите, които ще се използват. Правят се опити решението на задачата (или част от него) да се сведе до прилагането на вече познати методи

IV Етап – Реализация на алгоритъма с конкретен език за програмиране. Въвеждат се входните данни и се изпълнява програмата на компютър, прилагат се различни алгоритми, прилага се творчески подход при решаване на проблеми с компютър.

V Етап - Прави се проверка на съответствието на алгоритъма с поставените в задачата условия, има ли синтактични или семантични грешки, програмата се тества с различни входни данни и се проследяват резултатите от тестването ѝ.

ГЛАВА ТРЕТА. МОДЕЛ ЗА ИГРОВО-БАЗИРАНО ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ, СЪЧЕТАНО С ДИЗАЙН МИСЛЕНЕ ЗА РАЗВИВАНЕ НА АЛГОРИТМИЧНО МИСЛЕНЕ

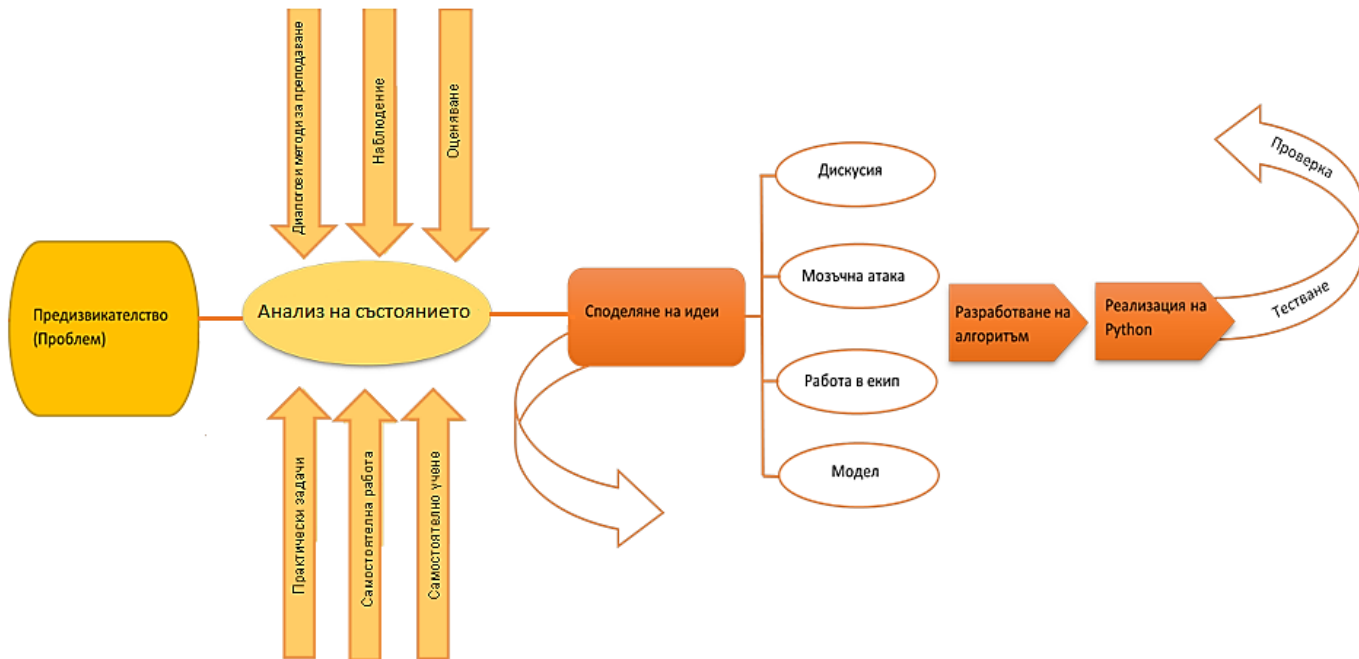
3.1. Описание на модела

3.1.1. Обща схема

Схемата е базирана на модела на работния процес на платформата за сътрудничество, базирана на дизайн мисленето Be-novative (<https://www.be-novative.com/>) и е адаптирана за игрово-базирано обучение по програмиране в гимназиална степен.

Условно, схемата може да се раздели на две основни, взаимно свързани части (фиг. III-1):

- Предварителна подготовка, включваща основни дейности - учител (Изложение, Диалог, Наблюдение, Оценяване), основни дейности – ученик (Практически задачи, Упражнение, Самостоятелна работа, Самостоятелно учене) и анализ на състоянието;
- Реализация - приложение на подхода дизайн мислене в основните етапи при решаване на задачи по програмиране (Предизвикателство (игра) → Споделяне на идеи → Разработване на алгоритъм → Реализация на играта (със средствата на редактор Ми на Python, режим Pугame→ Проверка и тестване).



Фигура III-1. Адаптиран модел на обучение по програмиране, базиран на дизайн мисленето

3.1.2. Реализация на модела

За реализация на задачите е избран езикът за програмиране Python с използване на библиотеки turtle и Pygame, в режим Pygame Zero в среда за програмиране Mu.

Python е съвременен, интерпретативен, интерактивен, обектно ориентиран език за програмиране от високо ниво, създаден от Гуидо ван Росум в началото на 90-те години². Има изчерпателна стандартна библиотека и хиляди допълнителни библиотеки в Python Package Index, библиотеките на много други програмни езици могат да се използват и в Python, системна интеграция, програмираните приложенията могат да работят на всички операционни системи, за които има интерпретатор на Python.

Предложените задачи за обучение по програмиране са използвани в извънкласна форма на обучение с ученици от 8. и 10. клас, които не изучават предмета информатика. Използван е игровият подход от гледна точка на обучение по програмиране чрез създаване на игри. Осъществява се със средствата на средата за разработка Mu на Python в режим Pygame Zero.

Първоначално Mu е създаден от Python Software Foundation за проекта micro:bit на BBC. Mu и е подходящ за ученици, които правят първите си стъпки в програмирането с Python, работи под Windows, OSX, Linux и Raspberry Pi. Лесен за използване, предлага се на всички основни платформи и е добре документиран дори и за начинаещи³. Mu е безплатен и може да бъде лесно свален и инсталиран от официалната страница на Mu⁴.

В част от примерните задачи се използва и библиотека turtle. Библиотеката turtle предоставя графични примитиви за костенурки, както по обектно-ориентиран, така и по процедурно-ориентиран начин. Чрез комбиниране на различни команди лесно могат да се нарисуват сложни форми и картини⁵.

3.1.3. Типовете задачи в контекста на Таксономията на Блум

Таксономия (от гръцки taxis – разположение, строй, порядък, и nomos –закон) представлява теория за класифициране и систематизиране на сложно организирани области на дадена дейност. В частност Педагогическата таксономия представлява система от педагогически цели, разпределени по категории и йерархични нива. Към когнитивната област на

² <https://brochure.getpython.info/media/releases/psf-python-brochure-vol.-i-final-download.pdf/view>

³ <https://mu.readthedocs.io/en/latest/>

⁴ <https://codewith.mu/en/download>

⁵ [Turtle. https://docs.python.org/3/library/turtle.html](https://docs.python.org/3/library/turtle.html)

педагогическите таксономии се отнася Таксономията на Блум, в която познавателните цели йерархично са разположени на шест нива: Знания → Разбиране → Приложение → Анализ → Синтез → Оценка. (Бижков, 1995)

В (Теодосиев, 2013) се разглежда обучението в стил на програмиране, с използване на Таксономията на Блум за определяне на целите на обучението по програмиране за студенти. Съпоставянето на етапите в усвояването на стила на програмиране и равнищата в пирамидата на Блум (Теодосиев, 2013, стр. 231) могат да бъдат адаптирани и приложени и в обучението по програмиране в гимназиална степен (фиг. III-2).



Фигура III-2. Процес на обучение по програмиране в пирамида на Блум.

3.2. Система от задачи

При експерименталното обучение по програмиране е използван конструктивисткия модел за игрово-базирано обучение, при който учениците се обучават, създавайки игри.

В Таблица 2. е представено разпределението на учебните часове по всяка една от темите в курса на обучение.

Таблица 2. Разпределение на часовете по теми

Тема	Урок	Брой часове по темата
Тема 1: Редактор Му на Python. Графика с библиотека turtle.	Запознаване с редактор Му на Python. Изчертаване на фигури.	4 часа
	Цветове и запълване на фигури. Рисувам с turtle.	4 часа
	Изчертаване на по-сложни фигури. Потребителски функции.	6 часа

Тема 2. Създаване на 2D игри с библиотека Pygame на Python.	Първи игри с Pygame.	8 часа
	Работа с изображения, герои, звуци и музика. Анимация.	4 часа
Тема 3. Софтуерен проект – създаване на 2D игра.	Създаване на 2D игра.	10 часа
Общ брой часове:		36 часа

Към всяка една от темите и съответните уроци е предложена методическа разработка на урока и система от задачи, включваща подробни описания на етапите при решаване на задачата и синтаксиса и семантиката на използваните програмни конструкции в нея. Посочено е и конкретното приложение на дизайн мисленето като подход при решаването на част от задачите. Задачите са използвани както за усвояване на нов учебен материал, така и за затвърждаване на вече придобити знания и умения.

Системата от задачи е описана подробно в параграф 3.2. и включва 25 задачи, като всяка от задачите е номерирана и включва условие на задачата, примерна реализация на Python и примерен изход от нея. Описани са и методически насоки за решаването им.

В параграф 3.2. са дадени и интернет адреси на платформи за самостоятелно учене, свързани с езика Python, библиотеките turtle и Pygame и редактор Ми на Python. Посочени са и адреси на галерии от изображения, които са използвани в част от задачите, както и адрес на приложение за създаване на собствени изображения, което се разпространява свободно.

ГЛАВА ЧЕТВЪРТА. ЕМПИРИЧНО ИЗСЛЕДВАНЕ НА МОДЕЛА

4.1. Описание на емпиричното изследване.

Експерименталното игрово-базирано обучение по програмиране с дизайн мисленето като подход е осъществено с ученици от 8. и 10. клас в ПГСАГ „Васил Левски“- Благоевград през учебната 2021/2022 година. Учениците не са изучавали учебните предмети „Компютърно моделиране“ и/или „Информатика“. В експерименталната група (ЕГ) участват 16 ученици. Контролната група (КГ) се състои от 17 ученици. Моделът за игрово-базирано обучение се приложи в обучението по програмиране в експерименталната група в извънкласна форма - Клуб по програмиране „Дигитален свят“.

За целите на изследването се създаде следния изследователски инструментариум:

- Тест за установяване на входното ниво на учениците;

- Анкета за установяване на влиянието на системата от задачи и мотивацията на учениците;
- Тест за установяване на изходящото ниво на учениците;
- Практическа задача за оценяване на постиженията на учениците;

Емпиричното изследване премина през следните етапи:

- I. Констатиращ експеримент – формиране на експериментална и контролни групи и установяване на входното равнище на алгоритмичното мислене на учениците и дигиталните им компетентности в двете групи.
- II. Същински експеримент - Емпирично тестване на модела
- III. Заключителен експеримент – установяване на изходното ниво на алгоритмичното мислене и дигитални умения в двете експериментални групи.

В параграф 4.1., в матрици на Тейлър, са представени тестовите спецификации на входното и изходното ниво по Таксономията на Блум, таблици с описание на целите за въпросите от входното и изходното ниво, както и критерии за оценка на практическата задача, използвана за измерване постиженията на учениците от експерименталната група (ЕГ).

4.2. Статистически методи за анализ на резултатите.

Анализът на данните е направен с IBM SPSS Statistics, JASP и MS Excel. Приложени са следните непараметрични статистически методи с равнище на значимост 0,05 - Ман-Уитни (Mann-Whitney), Колмогоров-Смирнов (Kolmogorov-Smirnov) и Уилкоксън (Wilcoxon).

Изследване надеждността на входно и изходно ниво

За проверка на надеждността на входното и изходното ниво е използван методът на Спийърмън-Браун (Spearman-Brown prophecy formula). Установи се, че при входното ниво ($r = 0.615$) надеждността е висока, а при изходното ниво ($r = 0.474$) съответно – средна.

4.3. Анализ на резултатите

4.3.1. Анализ на входните резултати

Тестът за определяне на входното ниво е съставен в две части – Дигитални компетентности и Алгоритмично мислене. Част от въпросите в него са от НВО Дигитални компетентности - 2015, 2016, 2018, 2020 година. Статистическите методи са приложени поотделно и общо за двете части на теста, като фокусът на изследването е върху частта „Алгоритмично мислене“.

Учениците, участвали в двете групи на изследването не са изучавали предметите „Компютърно моделиране“ и/или „Информатика“.

За целите на изследването основните хипотези H_0 са: Използването на предложения модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене в гимназиална степен, ще доведе до:

1. Развиване на алгоритмичното мислене на учениците;
2. Повишаване на общата дигитална компетентност на учениците;
3. Повишаване на интереса и мотивацията на учениците;

За проверка на нормалност при резултатите от входното ниво е приложен тест на Колмогоров-Смирнов (Kolmogorov-Smirnov) със статистически значимо ниво на Лилиефорс (Lilliefors). Тъй като броят на учениците е по-малък от 100, е приложен и тестът на Шапиро-Уилк (Shapiro-Wilk). (Ганева, 2016)

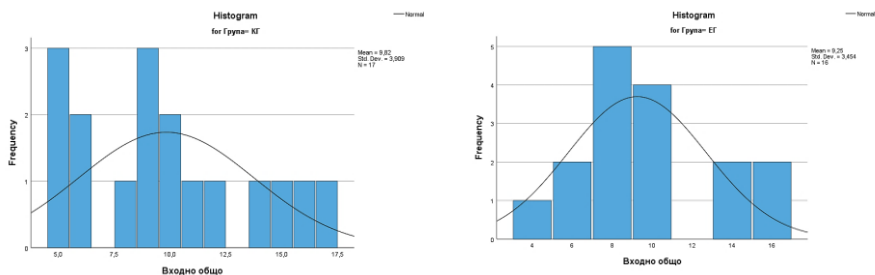
Проверяват се хипотезите:

H_0 – Данните имат нормално разпределение

H_1 – Данните нямат нормално разпределение

Установи се, че разпределенията и в двете групи не следват нормалното разпределение. Отхвърля се нулевата хипотеза H_0 – „Разпределението е нормално“ с равнище на значимост 0.05 и при двете групи.

Резултатите от проверката за нормалност са представени на фиг. IV-1.



Фигура IV-1. Резултати от тест за нормалност на разпределенията

Сравнение на резултатите от входния тест на контролната група (КГ) и експерименталната група (ЕГ).

Тъй като и двете групи не са с нормално разпределение и поради факта, че броят на изследваните лица във всяка група е по-малък от 30, е приложен теста на Mann-Whitney U. Този тест е един от най-популярните непараметрични тестове за проверка на две независими извадки, като при него се сравняват медианите на двете групи. (Ганева, 2016)

Резултатите от теста дават основание да се приеме с равнище на значимост 0.05 хипотезата H_0 – Двете групи статистически не се различават по отношение на постиженията, измерени с входното ниво (*Asimp. Sig. (P) = 0.677 > 0.05*)

4.3.2. Анализ на изходните резултати

В края на емпиричното изследване е направено финално оценяване на постиженията на учениците – изходен тест и практическа задача.

За анализ на изходните резултати са приложени статистическите тестове на Колмогоров-Смирнов и Шапиро-Уилкоксън за определяне нормалността на разпределенията, тест на Фридман (Friedman) за три и повече зависими извадки и теста на Ман-Уитни за сравнение на независими извадки.

Сравнение на резултатите от входен и изходен тест

Както входният, така и изходният тест е разделен на две части – „Дигитални компетентности“ и „Алгоритмично мислене“. Приложени са статистически тестове за резултатите общо за входен/изходен тест и по отделните критерии – Знание+Разбиране, Приложение и Анализ.

Сравнение на резултатите от изходен и входен тест в експерименталната група (ЕГ)

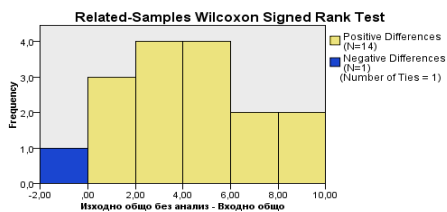
За сравнение на резултатите от изходния и входния тест на експерименталната група (ЕГ) е използван знаковият рангов тест на Уилкоксън (Wilcoxon signed ranks test). Този тест е непараметричната алтернатива на t -теста, който вместо да сравнява средноаритметичните стойности, преобразува стойностите на променливата в рангови и ги сравнява. Тестът изчислява броя на позитивните и негативните разлики и взема решение на базата на броя и размера на тяхното различие. (Ганева, 2016)

Хипотезите, които се проверяват със знаковия рангов тест на Уилкоксън:

H_0 - Вероятността на положителните разлики е равна на вероятността на негативните разлики, т.е. разликите не са статистически значими;

H_1 - Вероятността на положителните разлики е различна от вероятността на отрицателните разлики, т.е. разликите са статистически значими;

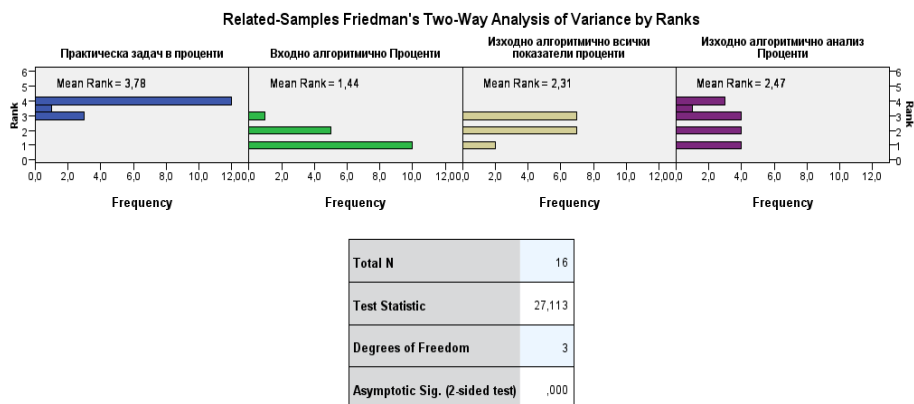
При сравнението на постиженията в частите от теста „Алгоритмично мислене“ - Знание+Приложение (*Asimp. Sig. = 0.016 < 0.05*) и Приложение (*Asimp. Sig. = 0.03 < 0.05*), както и при сравнението на общите резултатите от изходното и входното ниво в ЕГ, с равнище на значимост 0.05 се отхвърля нулевата хипотеза H_0 (*Asimp. Sig. = 0.001 < 0.05*). Това означава, че има повишаване резултатите по тези критерии в частта „Алгоритмично мислене“, както и на общите резултати на експерименталната група (ЕГ). (Фиг. IV-2).



Total N	16
Test Statistic	118,000
Standard Error	17,575
Standardized Test Statistic	3,300
Asymptotic Sig. (2-sided test)	.001

Фигура IV-2. Тест на Wilcoxon – изходно без анализ/входно общ резултат в ЕГ

За съпоставяне на резултатите в експерименталната група (ЕГ) от практическата задача, частите входно ниво – алгоритмично мислене, изходно ниво – алгоритмично мислене (Знание + Приложение) и по критерий – Анализ е използван тестът на Фридман (Friedman) за три и повече зависими извадки от рангови данни, който е непараметричната алтернатива на еднофакторния дисперсионен анализ за корелирани извадки (Ганева, 2016). Броят на точките за практическата задача, входно ниво – алгоритмично мислене и изходно ниво – алгоритмично мислене е различен. По тази причина, стойностите са нормирани в проценти (фиг. IV-3)



Фигура IV-3. Статистика на резултатите – тест на Фридман

Изследван е и коефициента на конкордация на Кендал W - непараметричен статистически тест, който се използва за измерване на статистическата връзка между множество обекти (резултати). Приема стойности от 0 до 1. Когато всички твърдения съвпадат, той е равен на 1, а когато различията са големи, то $W=0$. (Димитров & Карашранова, 1993) Получените резултати, както и стойността на коефициента $W=0.565$ дава основание да се приеме, че има статистически значима разлика в резултатите, което налага **извода**:

Постиженията на учениците от експерименталната група (ЕГ) в частта от теста „Алгоритмично мислене“ са значително по-високи. Това съществено се е отразило при прилагането на усвоените знания и умения на практика при решаването на практическата задача и се потвърждава основната хипотеза на изследването, а именно: Предложения модел развива алгоритмичното мислене.

Сравнение на резултатите от входно и изходно ниво на контролната група (КГ) и експерименталната група (ЕГ)

За сравнение на резултатите от входно ниво и изходно ниво на контролната (КГ) и експерименталната (ЕГ) групи е използван непараметричният тест на Ман-Уитни за две независими извадки. Резултатите от сравнението на резултатите от изходното ниво на контролната група (КГ) и експерименталната група (ЕГ) са представени в табл. 3:

Таблица 3. Сравнение на резултатите от изходното ниво на КГ и ЕГ

Критерии	Резултат
Дигитални компетентности – Знание+Разбиране	Няма статистически значима разлика
Дигитални компетентности – Приложение	Няма статистически значима разлика
Дигитални компетентности – Общо	Няма статистически значима разлика
Алгоритмично мислене – Знание+Приложение	Налице е статистически значима разлика. По-високи резултати в ЕГ.
Алгоритмично мислене – Приложение	Налице е статистически значима разлика. По-високи резултати в ЕГ.
Алгоритмично мислене – Анализ	Налице е статистически значима разлика. По-високи резултати в ЕГ.
Алгоритмично мислене – Общо с анализ	Налице е статистически значима разлика. По-високи резултати в ЕГ.
Изходно ниво – Общ резултат	Налице е статистически значима разлика. По-високи резултати в ЕГ.

Изводи: При резултатите на КГ и ЕГ от изходното ниво в част „Дигитални компетентности“ няма статистически значима разлика. Като логична причина за този резултат може да се посочи изучаването на предмета Информационни технологии – общообразователна подготовка и в двете групи от 8. до 10. клас. В частта от теста „Алгоритмично мислене“ – по отделните критерии и общия резултат, постиженията на учениците от експерименталната група са значително по-високи от тези в контролната група. Това налага извода, че игрово-базираното обучение, съчетано с дизайн мислене може да доведе до развиване на алгоритмичното мислене.

4.3.3. Анализ на анкетите за установяване на влиянието на системата от задачи при игрово-базираното обучение и мотивацията на учениците

Анализът се базира на резултатите от проведена стандартизирана анкета на 16 ученици от експерименталната група. Използвани са скалирани отговори с 5-степенна скала (рейтингова оценъчна скала). (Бижков, 1995) При проверка за надеждността на анкетата се използва коефициентът алфа α на Кронбах (Cronbach's coefficient alpha). „Ако стойността на коефициента е по-голяма от 0.70, се приема, че скалата е надеждна. „Ако скалата се състои от малък брой въпроси, може да се приемат за

нормални и стойности за α около 0.50“ (Ганева, 2016). Резултатът от теста за надеждност на анкетата $\alpha = 0.765$ показва, че анкетата е надеждна.

Изследвано е стандартното отклонение на всеки от въпросите в анкетата. Анализът показва, че използването на задачи с интересно условие, в които може да се прояви креативност, повишава интереса и активността на учениците. Предложените задачи са подходящи за усвояване и прилагане на практика на знания по програмиране с езика Python, модул Pygame с редактор Ми.

ЗАКЛЮЧЕНИЕ

Резултатите от проведените в съответствие с целта и задачите на дисертационния труд изследвания водят до следните изводи:

• **Приемат се работните хипотези: Използването на предложения модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене, ще доведе до:**

- Развиване на алгоритмичното мислене на учениците;

Стимулиране интереса и мотивацията на учениците;

• Прилагането на модела води до:

- по-високи общи тестови резултати;

- по-високи резултати по критерии – Знание+Приложение в част „Алгоритмично мислене“;

- по-високи резултати по критерий - Приложение в част „Алгоритмично мислене“

- по-високи резултати по критерий - Анализ в част „Алгоритмично мислене“;

- по-високи общи резултати в част „Алгоритмично мислене“;

- по-високи резултати при работа върху практическата задача.

• Използването на задачи с интересно условие, в които може да се прояви креативност, оказват положително влияние върху интереса и мотивацията на учениците по време на обучението по програмиране;

• Учениците предпочитат да им бъдат предоставени готови шаблони на функции за създаване алгоритъма на задачите;

• **Частично се отхвърля работната хипотеза: Използването на предложения модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене, ще доведе до повишаване на дигиталната компетентност на учениците:**

- по-високи общи резултати в част „Дигитални компетентности“ в експерименталната група;

- по-високи резултати по критерий – Приложение в част „Дигитални компетентности“ в експерименталната група;

- Няма промяна в резултатите по критерии – Знание и Знание+Разбиране в част „Дигитални компетентности“ в експерименталната група;
- Няма статистически значима разлика между контролната и експерименталната групи в част „Дигитални компетентности“
- Създаден е модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене.

Постиженията на учениците, в резултат от апробиране на модела, ни дават основание да продължим изследванията си в тази насока. Възможно е да се създаде модел за продължаващо обучение по програмиране с използване на езика Python и библиотека Pygame.

ПРИНОСИ

Научно-приложни:

1. Направено е проучване на среди за игрово-базирано обучение по информатика
2. Анализирани са възможностите за съчетаване на игрово-базирано обучение по програмиране с подхода дизайн мислене
3. Разработен е модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене;

Приложни:

1. Разработена е система от задачи, базирана на игри;
2. Разработеният модел за игрово-базирано обучение по програмиране, съчетан с дизайн мислене е апробиран в извънкласна форма на обучение – Клуб за занимания по интереси „Дигитален свят“ с ученици от 8. и 10 клас.

ПУБЛИКАЦИИ

1. Георгиева, Р., Тупарова, Д., Образователни компютърни игрови среди за обучение по програмиране и развитие на алгоритмично мислене – сравнителен анализ, 48. Пролетна конференция на Съюза на математиците в България, Боровец, 2019, ISSN 1313-3330
2. Тупарова, Д., Георгиева, Р., Дизайн мислене в помощ на обучението, Юбилейна международна конференция, Пампорово, 2020, ISBN 978-619-202-595-3
3. Георгиева, Р., Игрово-базирано обучение по програмиране за начинаещи в режим на Pygame Zero - примерни задачи с turtle графика на Python, Математика и информатика, София, 2023, 66(2), <https://doi.org/10.53656/math2023-2-6-gam>, ISSN 1314-8532 (online) 1310-2230 (print), стр. 170-186

Библиография

- Aleksić, V., Ivanovic, M., Popesco, E., & Budimac, Z. (2016). Commercial Off-the-Shelf Games as Learning Media. *17th International Conference on Computer Systems and Technologies 2016 - CompSysTech '16*, (стр. 355-360). Palermo, Italy.
- Hakak, S., Fazmidar, N., Noor, M., Nizam, M., Affal, H., Hussin, N., . . . Imranc, M. (2019). Cloud-assisted gamification for education and learning – Recent advances and challenges. *Computers & Electrical Engineering*, 22-34. Свалено от <https://www.sciencedirect.com/science/article/abs/pii/S0045790618304968>
- IDEO. (07 09 2008 г.). *Definitions of design thinking*. Изтеглено на 04 02 2022 г. от <https://designthinking.ideo.com/blog/definitions-of-design-thinking>
- Inno Spark. (2015). Сборник с добри практики и иновации. Изтеглено на 04 02 2022 г. от http://innospark-ict.eu/uploads/InnoSpark-IO4-Compendium_BG.pdf
- Inovation Starter. (04 06 2015 г.). *Дизайн мислене (Design Thinking)*. Изтеглено на 04 02 2022 г. от <http://innovationstarterbox.bg/resources/kakvo-e-dizain-mislene/>
- Kenderov, P. (2007). *Bulgaria – Birthplace of International*. Свалено от http://www.math.bas.bg/talents/en/inf/IOI_paper1.pdf: http://www.math.bas.bg/talents/en/inf/IOI_paper1.pdf
- Kenderov, P., & Maneva, N. (1989). International Olympiad in Informatics. *The International Olympiad in Informatics* (стр. 54). Sofia: Union of the Mathematicians in Bulgaria.
- Kirriemuir, J., & McFarlane, A. (2004). Literature Review in Games and Learning. *NESTA Futurelab series: report 8*, 1-38. Свалено от https://www.researchgate.net/publication/32231341_Literature_Review_in_Games_and_Learning
- Naiman, L. (20 05 2019 г.). *Design Thinking as a Strategy for Innovation*. Изтеглено на 04 02 2022 г. от <https://www.creativityatwork.com/design-thinking-strategy-for-innovation/>
- Pelling, N. (2011). *The (short) prehistory of “gamification”....* Свалено от Funding Startups (& other impossibilities): <https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>
- Red Paper Plane. (2022). Свалено от <https://bg.rpplane.com/schools>

- Stanford. (2018). Свалено от <https://web.stanford.edu/dept/SUSE/taking-design/presentations/Taking-design-to-school.pdf>
- Stanford. (2019). Свалено от https://web.stanford.edu/dept/SUSE/taking-design/proposals/Destination_Imagination_the_Fire_Within.pdf
- Stanford. (2022). *Design Thinking Bootcamp: Make Impact and Drive Growth in Your Organization*. Изтеглено на 04 02 2022 г. от <http://www.gsb.stanford.edu/exec-ed/programs/design-thinking-bootcamp>
- Trybus, J. (2015). *Game-Based Learning: What It Is, Why It Works, and Where It's Going*. Свалено от New Media Institute: <https://web.archive.org/web/20150911221236/http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html>
- Tuparov, G., Keremedchiev, D., Tuparova, D., & Stoyanova, M. (2018). Gamification and educational computer games in open source learning management systems as a part of assessment. *17th International Conference on Information Technology Based Higher Education and Training (ITHET)* (стр. 1-5). Olhao, Portugal : IEEE.
- Tuparova, D., Tuparov, G., Veleva, V., & Nikolova, E. (2018). Educational computer games and gamification in informatics and information technology education - Teachers' points of view. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (стр. 702-709). IEEE. Свалено от https://www.researchgate.net/publication/326699762_Educational_computer_games_and_gamification_in_informatics_and_information_technology_education_-_Teachers%27_points_of_view
- Азълъв, П. (1989). Първа международна олимпиада по информатика. *Математика*(7), 41-45.
- Азълъв, П. (2009). Олимпиадите по информатика - факти и събития, които все още се помнят. *Математика и информатика*(1), 3-18.
- Асенова, П. (1990). Построение и използване системи задач в курс алгоритмизации. Москва: Russian Academy of Science.
- Бижков, Г. (1995). *Методология и методи на педагогическите изследвания*. София: АСКОНИ-ИЗДАТ.
- Ганева, С. (2016). *Да претоткрием статистиката с IBM SPSS Statistics*. Свалено от <https://www.elbook.eu/images/book25.pdf>
- Гроздев, С., & Гърров, К. (2007). Основни учебни дейности при подготовка на ученици за участие в олимпиади по информатика. *Наука*,

- образованието и времето като грижа. Юбилейна научна конференция с международно участие.* Смолян.
- Гроздев, С., & Гъров, К. (2008). За системите от опорни задачи при подготовката за участие в олимпиади по информатика. Комбинаторни обекти и алгоритми. *37 Пролетна конференция на СМБ*, (стр. 304-311). София.
- Гъров, К. (2000). Обучението по информатика и информационни технологии в средното училище - състояние и перспективи. *Юбилейна научна сесия - 30 години ФМИ, ПУ "Паусий Хилендарски", Пловдив.* Пловдив.
- Гъров, К. (2004). Система от опорни задачи при подготовката на талантиливи и изявени ученици за участие в олимпиади и състезания по информатика. *33 Пролетна конференция на СМБ*, (стр. 316-321). София.
- Гъров, К. (2010). Задачите в обучението по информатика и информационни технологии. *Национална конференция "Образованието в информационното общество.* Пловдив.
- Гъров, К., & Годорова, Е. (2006). Примерна система от опорни задачи по темата "Алгоритми и задачи от теория на числата" за подготовка на талантиливи ученици по информатика. *35 Пролетна конференция на СМБ.* Боровец.
- Димитров, Б., & Карашчанова, Е. (1993). *Статистика за нематематици.* Благоевград: ЮЗУ "Неофит Рилски", Благоевград.
- Дурева, Д. (2003). *Проблеми на методиката на обучение по информатика и информационни технологии.* Благоевград: Университетско издателство ЮЗУ "Неофит Рилски".
- Дурева, Д., & Касева, М. (2011). Компютърните образователни игри в обучението в началното училище. *Информационно-комуникационни технологии, медии и образование.* Благоевград.
- Кендеров, П. (2009). България - родното място на Международната олимпиада по информатика за ученици. *Математика и информатика, кн.3, 3-7.* (Е. Сендова, Интервюиращ) София.
- Манев, К. (2009). Международната олимпиада по информатика - "Българската" олимпиада. *Математика и информатика(2), 3-11.*
- ММИБ. (н.д.). *Международни олимпиади по информатика (1989-2008).* Свалено от http://mmib.math.bas.bg/?page_id=20788: http://mmib.math.bas.bg/?page_id=20788

- МОН. (2000). *Държавни образователни изисквания за учебното съдържание*. Свалено от <https://mon.bg/bg/100105>.
- МОН. (2005). *Национална стратегия за въвеждане на ИКТ в българските училища*. Свалено от <https://www.strategy.bg/StrategicDocuments/View.aspx?lang=bg-BG&Id=398>:
<https://www.strategy.bg/StrategicDocuments/View.aspx?lang=bg-BG&Id=398>
- МОН. (2014). *Стратегия за ефективно прилагане на информационни и комуникационни технологии в образованието и науката на Република България (2014-2020 г.)*. Свалено от <https://www.strategy.bg/StrategicDocuments/View.aspx?Id=904>.
- МОН. (2015). *Наредба № 4 за учебния план*. Свалено от <https://mon.bg/bg/59>.
- МОН. (2016). *Наредба № 7 за профилираната подготовка*. Свалено от <https://mon.bg/bg/59>.
- МОН. (2018). *Учебни програми за профилирана подготовка*. Свалено от Министерство на образованието и науката: <https://www.mon.bg/bg/100598>
- МОН. (2018). *Учебна програма по компютърно моделиране за трети клас (общо образователна подготовка)*. Свалено от Министерство на образованието и науката: <https://www.mon.bg/bg/1689>
- МОН. (2019). *Учебна програма по компютърно моделиране за четвърти клас (общо образователна подготовка)*. Свалено от Министерство на образованието и науката: <https://www.mon.bg/bg/2190>
- МОН. (2020). *Учебна програма по компютърно моделиране и информационни технологии за V клас*. Свалено от Министерство на образованието и науката: <https://www.mon.bg/bg/100884>
- Наков, С., & Колектив. (2017). *Основи на програмирането със C#*. Фабер. Свалено от <https://raw.githubusercontent.com/SoftUni/Programming-Basics-Book-CSharp-BG/master/resources/Programming-Basics-CSharp-v2017.pdf>
- Николова, Е. (2019). Интегративен подход в обучението по информатика в гимназиалната училищна степен. Благоевград.
- Петров, А., Вълканова, В., & Димитров, И. (2016). Игрово-базирано обучение във виртуално образователно пространство. *Юбилейна научна конференция с международно участие* (стр. 451-456). Пловдив: Пловдивски университет „Паисий Хилендарски“. Свалено от

- http://research.bfu.bg:8080/jspui/bitstream/123456789/909/1/451_PDFsam_Final_Tom2.pdf
- Радославова, С. (27 04 2018 г.). *Ученето като удоволствие, провокация на креативността и игра*. Изтеглено на 31 01 2022 г. от https://www.capital.bg/specialni_izdaniia/obrazovanie/2018/04/27/3375246_ucheneto_kato_udovolstvie_provokaciia_na_kreativnostta/
- СМБ. (н.д.). *Международни олимпиади по информатика*. Свалено от http://mmib.math.bas.bg/?page_id=20786.
- Спирова, М. (2018). Място на игрово-базираното обучение в училище. *XI Национална конференция „Образованието и изследванията в информационното общество“*, (стр. 41-46). Пловдив. Свалено от <http://webcache.googleusercontent.com/search?q=cache:MDvQKGmK58MJ:sci-gems.math.bas.bg/jspui/bitstream/10525/2945/1/ERIS2018-book-p04.pdf+&cd=1&hl=bg&ct=clnk&gl=bg&client=firefox-b-d>
- Табакова-Комсалова, В. (2018). *Формиране на алгоритмично мислене у учениците в началното училище и прогимназията чрез обучението по информатика и информационни технологии. Дисертационен труд*. Пловдив: Пловдивски Университет "Паисий Хилендарски".
- Теодосиев, Т. (2013). Таксономията на Блум и обучението в стил на програмиране. *VI Национална конференция "Образованието в информационното общество"*, (стр. 227-236). Свалено от <http://sci-gems.math.bas.bg/jspui/bitstream/10525/2333/1/EIS2013-book-p23.pdf>
- Терзиева, Т. (2010). Резултати от изследване върху понятието алгоритмично мислене. . *Юбилейна международна научна конференция „Синергетика и рефлексия в обучението по математика“*. 10-12 Септември (стр. 479-486). Бачиново: Факултет по математика и информатика ПУ "Паисий Хилендарски". Свалено от <http://fmi-plovdiv.org/GetResource>
- Терзиева, Т. (2012). Развитие на алгоритмичното мислене в обучението по информатика. *Дисертационен труд за присъждане на образователна и научна степен "Доктор"*. Пловдив: Пловдивски Университет "Паисий Хилендарски".
- Терзиева, Т., Голев, А., & Ставрев, С. (2017). Сериозните игри – иновативно средство за обучение. *"Иновативни софтуерни инструменти и технологии с приложения в научни изследвания по математика, информатика и педагогика на обучението"*, (стр. 107-114). Пампорово.

- Тупарова, Д., & Георгиева, Р. (2019). Образователните компютърни игрови среди за обучение по програмиране и развитие на алгоритмично мислене - сравнителен анализ. *48 Пролетна конференция на СМБ*, (стр. 150-155). Боровец.
- Тупарова, Д., & Георгиева, Р. (2020). Дизайн мислене в помощ на обучението. *Юбилейна международна научна конференция "Синергетика и рефлексия в обучението по математика"*. Пампорово: Университетско издателство "Паисий Хилендарски".
- УНИБИТ. (2022). *Мисия и цели*. Изтеглено на 03 02 2022 г. от <https://www.unibit.bg/about-unibit/overview/mission-goals>
- ЦИОО -МОН. (2015). *Наредба № 5 за общообразователната подготовка*. Свалено от <https://cioo.mon.bg/>: https://cioo.mon.bg/wp-content/uploads/2014/07/nrdb5_30.11.2015_obshtoobr_podgotovka.pdf



SOUTH-WEST UNIVERSITY "NEOFIT RILSKI"
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
DEPARTMENT OF INFORMATICS

Rositsa Nikolaeva Georgieva

**GAME-BASED LEARNING FOR THE DEVELOPMENT OF
ALGORITHMICAL THINKING IN SCHOOL EDUCATION**

ABSTRACT

ON DISSERTATION PAPER
FOR THE AWARD OF THE EDUCATIONAL AND SCIENTIFIC DEGREE
"DOCTOR"

AREA OF HIGHER EDUCATION: 1. PEDAGOGICAL SCIENCES
PROFESSIONAL FIELD: 1.3. PEDAGOGY OF TRAINING IN ...
DOCTORAL PROGRAM "METHODOLOGY OF TRAINING IN
INFORMATICS AND INFORMATION TECHNOLOGIES"

Supervisor:
Prof. Dr. Daniela Dureva-Tuparova

Blagoevgrad, 2023

The dissertation "Game-based learning for the development of algorithmic thinking in school education" contains 166 pages of main text in a separate body and Appendices in 23 pages. The main text is illustrated with 101 figures and 16 tables. The literature used includes 103 sources, of which 56 are in Cyrillic, 47 are in Latin.

The dissertation work is structured in an introduction, four chapters, a conclusion, used bibliographical sources. Appendices include tools used during training.

The dissertation work was discussed and accepted at a meeting of the Departmental Council of the Department of Informatics, Faculty of Science and Mathematics at SWU, "Neofit Rilski", Blagoevgrad on 25.04.2023 and is scheduled for defense on 12.06.2023.

Defense materials are available to those interested in the "Informatics" department.

CONTENT:

INTRODUCTION.....	3
CHAPTER ONE. THEORETICAL POSITIONS.....	7
1.1. Nature and characteristics of algorithmic thinking.....	7
1.2. Design thinking as an approach to learning.....	9
1.3. Gamification and game-based learning.....	13
1.4. Appropriate games and learning programming environments.....	16
CHAPTER TWO. INFORMATICS EDUCATION IN THE BULGARIAN SCHOOL. THE ROLE OF TASKS IN EDUCATION.....	19
2.1. The beginning.....	19
2.2. In the present.....	21
2.3. Extracurricular forms of informatics.....	21
2.4. Tasks in informatics education.....	22
2.5. Stages in solving tasks in informatics.....	23
CHAPTER THREE. A MODEL FOR GAME-BASED PROGRAMMING LEARNING COMBINED WITH DESIGN THINKING TO DEVELOP ALGORITHMICAL THINKING.....	25
3.1. Model description.....	25
3.1.1. General scheme.....	25
3.1.2. Implementation of the model.....	27
3.1.3. Task types in the context of Bloom's Taxonomy.....	27
3.2. Task system.....	28
CHAPTER FOUR. EMPIRICAL RESEARCH OF MODEL.....	30
4.1. Description of the empirical study.....	30
4.2. Statistical methods for the analysis of the results.....	30
4.3. Analysis of results.....	31
4.3.1. Analysis of input results.....	31
4.3.2. Analysis of output results.....	32
4.3.3. Analysis of surveys to establish the influence of the task system in game-based learning and student motivation.....	36
CONCLUSION.....	37
CONTRIBUTIONS.....	38
PUBLICATIONS.....	39
BIBLIOGRAPHY.....	39

INTRODUCTION

The world today needs a new generation of minds with new ways of thinking and acting. Future generations will be called upon to solve global human, technological and natural problems that did not exist before. This necessitates a change in thinking about the world and, in particular, about education. One modern approach supporting learning is design thinking (DM). (Tuparova & Georgieva, 2020)

On the other hand, the rapid development of information technologies in recent decades and the massive penetration of digitization in various areas of public life pose challenges to modern secondary education that must be answered adequately and promptly. (Tuparova & Georgieva, 2019)

One of the ways to increase student motivation is game-based learning. A key feature of using computer game environments in learning is the challenge of solving specific problem situations. The interest generated by games undoubtedly influences learning motivation, critical thinking and problem solving, practical skills and self-control.

In addition, planning skills, accurate, complete and understandable description of actions will help students develop their algorithmic thinking. Using the design thinking approach in game-based learning, creative thinking skills are stimulated and developed, especially in the ICT field, the process of developing innovations is encouraged and it contributes to the growth and competitiveness of future computer specialists. Therefore, the application of game-based programming learning combined with design thinking is particularly appropriate. (Tuparova & Georgieva, 2020)

In the present dissertation, a model of game-based learning combined with the design thinking approach for the development of algorithmic thinking is described. A system of tasks is described, the implementation of which aims to acquire knowledge and skills in programming, develop algorithmic thinking and increase the digital competence of students.

The dissertation work was developed and shaped according to "Methodology and methods of pedagogical research" (Bizkov, 1995).

Topic: Game-based learning for the development of algorithmic thinking in school education

The object of research is the algorithmic skills of students from the 8th to the 10th grade in the PGSAG "Vasil Levski" - Blagoevgrad, who did not study the subjects "Computer Modeling" and "Informatics" according to the curriculum.

The subject of the study is the process of formation and development of algorithmic skills of first high school students by using game-based programming learning combined with design thinking as an approach to developing algorithmic thinking.

The aim of the study is to develop students' algorithmic thinking, increase their digital competence, as well as their motivation to learn through game-based programming learning combined with design thinking as an approach.

Tasks:

1. To analyze trends in informatics education on a national scale:

- literature review on the issue;
- exploring suitable platforms, environments and languages for game-based programming learning;
- game-based learning and its role in acquiring knowledge, skills and competencies;
- application of design thinking as an approach to education in Bulgaria and the world;

2. To analyze the state of informatics education in compulsory and extracurricular forms on a national scale:

- research and analysis of compulsory training in informatics in the country;
- study of programming education in extracurricular forms;
- the role and place of tasks in programming education.

3. To explore the possibility of increasing digital skills and developing students' algorithmic thinking using game-based learning and design thinking as an innovative approach;

4. To develop a model for game-based training in programming to develop algorithmic thinking, combined with the design thinking approach;

5. To investigate the results of the proposed model regarding the digital competence and the development of algorithmic thinking of the students, as well as the impact on the motivation to learn.

6. To validate the developed model through empirical research.

Research questions

Will game-based learning combined with design thinking in programming education lead to developing students' algorithmic thinking?

Is it possible to increase students' digital competence in game-based programming education?

Will the students' motivation to learn and interest in programming increase?

A hypothesis consisting of three components was raised:

Using the proposed model for game-based programming learning combined with design thinking in high school will lead to:

1. Developing students' algorithmic thinking;
2. Increasing the general digital competence of students;
3. Increasing students' interest and motivation;

Research methods:

- Review and analysis of literature related to the subject of research;
- Pedagogical research: ascertaining, basic and concluding;
- Methods of collecting empirical material: pedagogical experiment, survey, testing;
- Statistical methods for processing and analyzing the empirical material.

Stages of the study:

- Stage (2018-2019) – Study of the current literature related to the subject of research.
- Stage (2019-2020) – Study of the current literature related to the subject of research. Development of a game-based learning model.

- Stage (2021-2022) – Study of the current literature related to the subject of research. Testing the developed model for game-based learning. A survey of the students participating in the study in order to establish their opinion regarding the use of games in programming education. Testing of students to establish the entry level and the exit level of knowledge, Statistical processing and analysis of the results of the empirical pedagogical research.

Toolkit:

- Survey to determine the opinion and attitude of the students
- Test and practical task to check and evaluate students' knowledge and skills
- The IBM SPSS Statistics and JASP software packages were mainly used for the statistical processing of the empirical data.

The dissertation work is structured in accordance with the set tasks. It consists of an introduction, four chapters, a conclusion, a bibliography, a list of publications related to the thesis, appendices, lists of tables, tasks and figures.

CHAPTER ONE. THEORETICAL POSITIONS

1.1. Nature and characteristics of algorithmic thinking

There is no single concept of "algorithmic thinking" in the literature. Difficulties are caused by various factors, one of which is that often the concept of "algorithmic" is replaced by concepts such as "logical", "engineering", "mathematical", "abstract", "computational" and others.

On the other hand, the problem of defining the concept is related to the understanding of its essence. A study conducted (Terzieva, 2010, pp. 479-486) to understand the concept of algorithmic thinking shows a lack of unified understanding regarding the content of the concept of algorithmic thinking.

According to Terzieva, algorithmic thinking is "a way of thinking that provides a solution to a given task through a sequence of elementary actions" (Terzieva, 2012, p. 31). The same researcher adds that "algorithmic thinking consists of a wide range of skills and is influenced by many other cognitive factors: abstract and logical thinking, structural thinking, problem-solving ability and creativity" (Terzieva, 2012, p. 31).

Other definitions of algorithmic thinking can be found in the scientific literature (Tabakova-Komsalova, 2018), (Nakov & Kolectiv, 2017). For the purposes of this study, the algorithmic thinking formulation of (Terzieva, 2012) is adopted. The main components of algorithmic thinking are systematized and presented in the following Figure I-1.

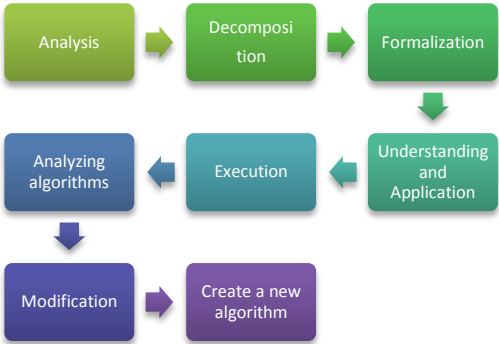


Figure I-1. Components of algorithmic thinking

Today, the benefits of algorithmic thinking are realized not only by the ICT sector, but also by the educational system in various countries, including Bulgaria.

Developing algorithmic thinking at an early age (school age) is important because it “allows students to solve problems, delve into designing systems, and understand the power and limits of human thinking and machine intelligence. This is a skill that creates a prerequisite for all students to carry out a real self-assessment and to develop and improve their competences” (Tabakova-Komsalova, 2018, p. 26). The formation of algorithmic thinking is "not an easy task, requiring a precise didactic approach" (Tabakova-Komsalova, 2018, p. 21).

One of the main didactic goals of school education in the modern Bulgarian school is the development of algorithmic thinking. Already in the 3rd (Ministry of Education, 2018) and 4th (Ministry of Education, 2019) grades, the development of algorithmic thinking is embedded in the computer modeling program, and its improvement continues in 5th-7th grades. class, in the set curriculum of computer modeling and information technologies (Ministry of Education, 2020). The development of algorithmic thinking in Bulgarian schools is advocated both in general educational curricula (Ministry of Education, 2020) on computer modeling and information technologies, and in curricula for profiled training (informatics and information technologies) (Ministry of Education, 2018).

Algorithmic thinking allows to rationalize the path to a solution of a task and is manifested in the ability of learners to use inductive and deductive methods and mental operations in the analysis of problems from various fields, use methods to formalize solutions to a task to record in an algorithmic language" (Terzieva, 2012, p. 34).

One of the known effective ways to develop algorithmic thinking in elementary school students is through “additional information technology classes in elective classes, interest clubs, informatics schools, and others. In the course of studying computer programming and information and communication technologies, knowledge and skills are acquired for the compilation of algorithms and their use in solving a large class of programming tasks" (Tabakova-Komsalova, 2018, p. 21).

The Bulgarian education system faces the need to speed up the process of forming students' algorithmic thinking, due to the modern conditions of dynamic

development of digitization and automation in all spheres of life, including education.

The development of algorithmic thinking supports not only the solution of educational tasks related to informatics and IT, but also the solution of other challenges of an educational, work and life nature. The opinion is shared about "the incorrectness of the opinion that the algorithmic style of thinking and algorithms are inherent only to work with computing equipment, because tasks from absolutely all spheres of life are solved through digital devices" (Tabakova-Komsalova, 2018, p. 20) .

1.2. Design thinking as an approach to learning

The origins of Design Thinking (DM) from a scientific point of view can be traced back to the late 1960s, but the term "design thinking" appeared in the late 1980s and 1990s, when Rolf Faste began teaching design thinking as a "method of creative action" at Stanford University. DM was adapted and applied in business by David Kelly of Stanford, founder of the design consultancy IDEO in 1991. In the business vocabulary, the method entered the mainstream after its popularization by Tim Brown of IDEO (Innovation Starter, 2015). (Table 1)

Table 1. *Stanford model (Naiman, 2019), (Innovation Starter, 2015), (Stanford, 2018)*

Practices	Principles	Stages	
		Short process	Extended process
Developing a deep empathic understanding of the user's needs	Discovery – I meet a new challenge. How do I overcome it?	Observing and studying user behavior and experience, exploring and defining a problem and perspective based on empathy	Observing and studying user behavior and experience, exploring and defining a problem and perspective based on empathy
Formation of heterogeneous teams	Interpretation – What did I learn and how do I interpret it?	Generating a lot of ideas in a huge amount in a short period of time	Generating a lot of ideas in a huge amount in a short period of time
Conversations based on dialogue	Ideation – I see an opportunity, how to shape it into an idea?	Selection and classification of ideas	Selection and classification of ideas

Generating solutions through experimentation	Experimentation - I have an idea how to build it?	Prototyping- a product for which there is quick feedback from users	Prototyping - a product for which there is quick feedback from users
The use of a structured and facilitated process	Evolution- Tried something new, how to evolve it?	Feedback - how to improve the prototype if needed	Feedback - how to improve the prototype if needed
			Testing if the idea works
			Learning and improving – again based on feedback

Today in the literature there are different definitions of the concept. According to (IDEO, 2008). "design thinking is a human-centered approach to innovation that draws on the designer's toolkit to integrate the needs of people, the capabilities of technology, and the requirements of business success." Again in (IDEO, 2008) it is added that "design thinking is a way of solving problems through creativity, a process for creative problem solving that stimulates the most important quality in today's world - to constantly change the environment around you by you improve, driven by your curiosity and creative confidence."

"Design thinking is based on logic, imagination and intuition and is not problem-focused, but solution-oriented and action-oriented. Much of the concept of design thinking involves empathy for those for whom the new products are being created" (Naiman, 2019).

In its essence, design thinking is a circular process, the purpose of which is to improve and develop products, services, processes and business models (Innovation Starter, 2015).

Although DM is a business methodology, today it is widely used in the public, non-governmental, health and education sectors. In the last decade, more and more educational institutions are turning to DM as a method to support effective learning.

DM in education can be applied in several key areas presented in the following figure I-2.

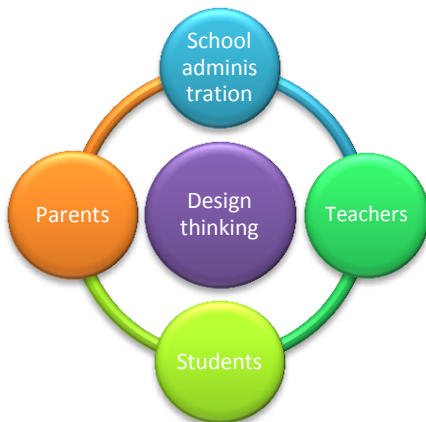


Figure I-2. Directions of application of DM in education

In education, DM follows certain stages and methods, presented in the following fig.I-3 (Tuparova & Georgieva, 2020):

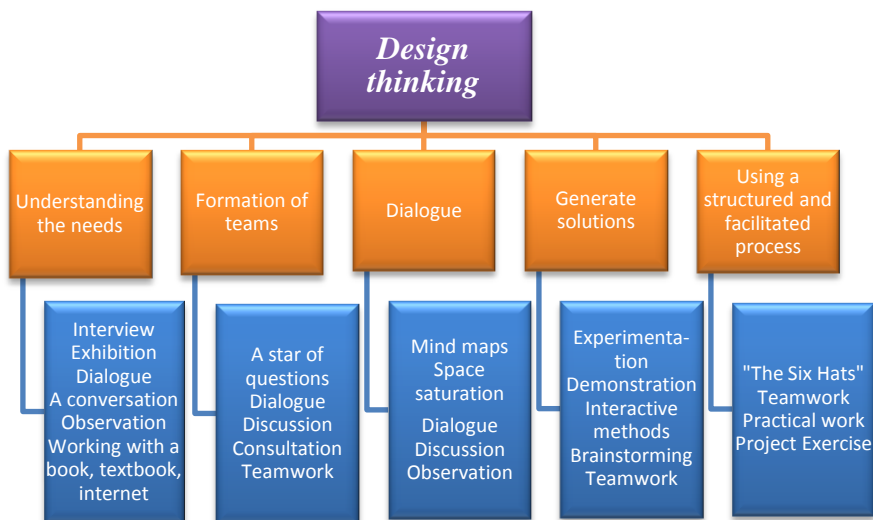


Figure I-3. Learning methods in the stages of design thinking

To date, many good practices related to the application of DM in education have been accumulated. One of the examples at the world level is the D-school or Institute of Design at Stanford University, founded in 2005. In order to find a new set of solutions for their education system, a Prototype Design Camp is created at D-school to introduce and apply design thinking skills. The conclusions shared by the organizers of Prototype Design Camp is the belief that design thinking as a teaching approach teaches students to collaborate (Standford, 2018), (Standford, 2019).

Another such camp is Bootcamp Design Thinking, where through a series of role-playing games with different characters, learners are given the opportunity to rapidly generate ideas with the ability to visualize and adapt the results in real time (Standford, 2022).

The implementation of the DM's ideas does not escape Bulgaria either. Within the framework of international and regional projects, programs for the development of DM in the field of ICT are being developed. Design Champions Program - Red Paper Plane is a project-based learning program that uses the design thinking method. The program is already implemented in 120 educational institutions in 65 settlements in the country by 400 teachers. (Red Paper Plane, 2022), (Radoslavova, 2018).

Another example is the Telerik Academy, which includes the programs "Software Academy" (software engineering), "School Academy" (software programming for students between the ages of 14 and 19), "Children's Academy" (programming training for children from 10 up to 12 years).

The Software University (<https://softuni.bg/>) is based on the idea of an innovative educational center that uses the "learning by doing" model, providing programming training combined with real practical experience and knowledge in new technologies. It also includes SoftUni Kids - a program suitable for students from 1st to 6th grade, which is based on an innovative learning approach based on games for acquiring knowledge and skills in the field of programming (Inno Spark, 2015)

The University of Library Science and Information Technology (UniBIT) is also a good example of training largely based on design thinking as a methodology, with

the main priorities being the achievement of high quality in ICT sector training and their application in practice (UNIBIT, 2022).

The project "Professional development of teachers to promote design thinking skills and academic success of students" (DTS)¹ is also of interest. The project is financed by the Erasmus+ program of the European Union and partners in the project are Turkey, Bulgaria (YUZU "Neofit Rilski - Blagoevgrad"), Portugal and Italy. The project is aimed at students, teachers and institutions in secondary education.

1.3. Gamification and game-based learning

Gamification is the concept derived from the English word for game. The term itself was coined in 2002 by Nick Pelling, a game designer. In (Pelling, 2011), the author says: "... I coined the deliberately ugly word 'gamification', by which I meant the application of game-like accelerated user interface design to make electronic transactions both enjoyable and fast."

The authors (Tuparova, Tuparov, Veleva, & Nikolova, 2018), citing Kapp, define gamification as "the use of game-based mechanics, aesthetics, and game thinking to engage people, motivate action, promote learning, and problem solving ". The term can also be described as the concept of introducing gaming techniques into various processes and fields such as business, marketing and education. (Trybus, 2015).

In the field of education, gamification is manifested through the concept of game-based learning (GBL). GBL is a means of acquiring knowledge and skills through play, where game activities require the player to solve problems and challenges provided to achieve achievements (Kirriemuir & McFarlane, 2004). Other authors define GBL as "a form of student-centered learning that uses digital games for educational purposes. This expression is a semantic composition of the two concepts, play and learning, with a focus on learning" (Aleksić, Ivanovic, Popesco, & Budimac, 2016, p. 355).

According to (Spirova, 2018, p. 45), GBL "has two options:

¹ <https://dts4teachers.eu/>

- Instructional model: students learn by playing educational games set and developed by others (teacher);
- Constructivist model: students learn by playing educational games set and developed by themselves”.

GBL provides an opportunity for teachers to incorporate active learning into lessons, encourage student interest and engagement, and provide immediate feedback on individual and overall performance.

The implementation of GBL in school education is associated with many advantages and some disadvantages for both students and teachers, systematized and presented in the following figure. I-4(p.15)

The GBL concept is becoming increasingly popular at all levels of education, including school education, as an effective way to engage students in learning across disciplines. In the majority of the presented Bulgarian scientific publications, the interest in the topic of GBL is mainly focused on the use of educational computer games and environments in the primary level of education.

A number of authors are doing research on topics related to game-based learning. In (Spirova, 2018, pp. 41-46) the implementation of some of the main didactic principles through game-based learning is considered, in (Dureva & Kaseva, 2011) the difficulties experienced by teachers in game-based learning are outlined, as and the advantages of game-based learning, in (Tuparova, Tuparov, Veleva, & Nikolova, 2018) a study was made of the attitude of informatics and information technology (IT) teachers to the use of educational computer games and gamification.

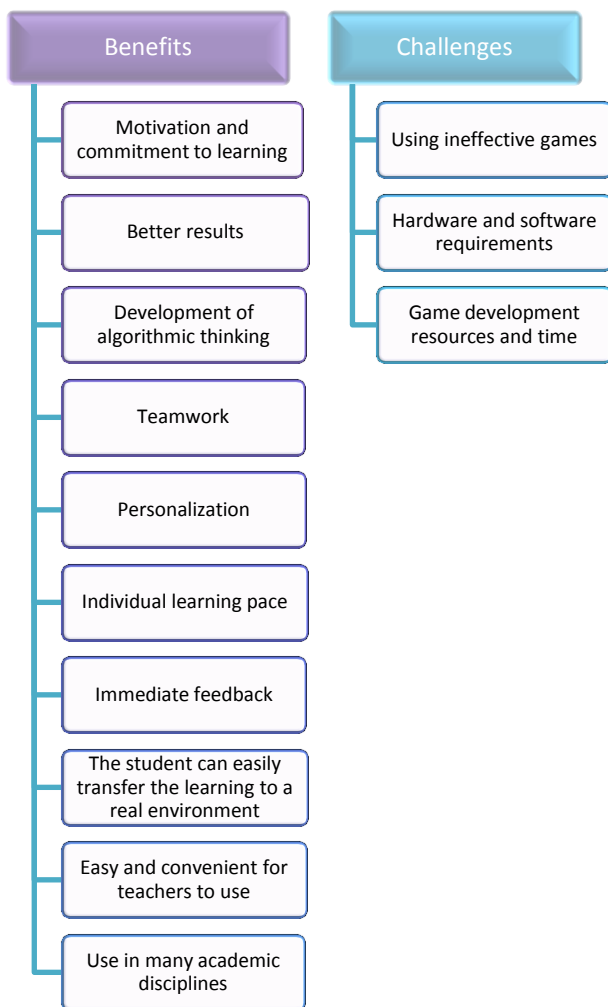


Figure I-4. *Benefits and challenges of GBL in school education*

In the article of (Tuparov, Keremedchiev, Tuparova, & Stoyanova, 2018) outlined the framework for identifying the characteristics of gamification in open e-learning environments and the application of gamification in assessment, peer assessment and self-assessment activities in the Moodle e-learning environment. Pedagogical aspects and technological characteristics of gamification in education are examined. In the article by (Hakak, et al., 2019), the authors' focus is on

gamification and the possibilities of cloud technologies as one of the modern ways to modernize learning modules and explores the applicability of gamification in the various subject areas that can be taught through the cloud service.

1.4. Appropriate games and learning programming environments

In the scientific literature, there are various terms and formulations of concepts related to the application of computer games in the educational process - educational video games, didactic computer games, serious games, educational computer games. Educational games are those serious games whose purpose is the transmission of specific knowledge and skills in some field and the provision of new knowledge through a game approach" (Terzieva, Golev, & Stavrev, 2017). In (Dureva & Kaseva, 2011), educational computer games are defined "as a positive factor for the development of the child's thinking, memory and attention and the formation of generalization and classification skills, especially in the development of coordination and motor skills in primary education".

Educational games are a powerful tool to support traditional school learning in various subjects. Their role in the formation and development of algorithmic thinking is particularly important.

A key feature of using computer game environments in learning is the challenge of solving specific problem situations. The student interacts with the virtual environment of the game, developing skills to experiment, make different decisions, control the development of the situation. The interest generated by games undoubtedly influences learning motivation, critical thinking and problem solving, practical skills, and self-control (Petrov, Valkanova, & Dimitrov, 2016).

Classification of game-based programming learning environments

Environments are classified into two main groups (Tuparova & Georgieva, 2019):

- Programming environments - in which educational computer games are easily developed;
- Environments for programming in game situations.

Some of the most popular programming environments for educational computer games are systematized and presented in the following figure. I-5:

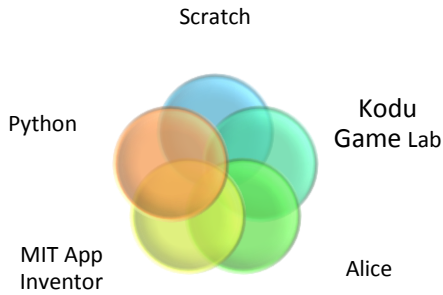


Figure I-5. *Programming environments for educational computer games*

The most popular programming environments in game situations, often used in practice, are systematized and presented in the following figure. I-6:

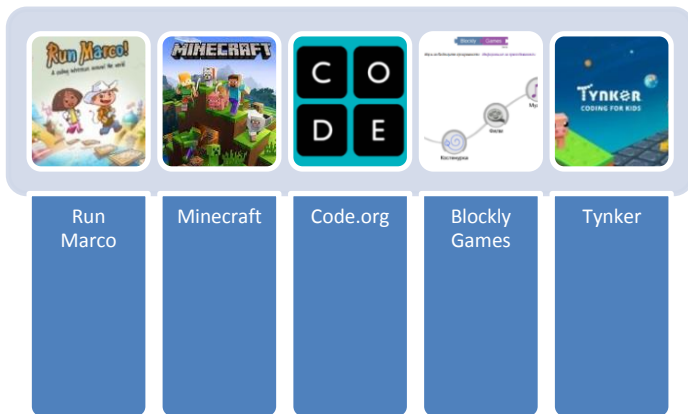


Figure I-6. *Programming environments in game situations*

The effective selection of educational games/environments is related to adherence to a certain framework with several key criteria presented in the following figure I-7. (Tuparova & Georgieva, 2019)



Figure I-7. *A framework for comparing educational games/environments*

1. Type of environment - determined by whether it is programming only or provides programming in game situations.
2. Applicability in a specific educational level - preschool, elementary, junior high and high school.
3. Interface - localization in different languages, intuitiveness when working with the environment.
4. Mode of operation - the possibility of working online or offline is provided.
5. Distribution – paid or free.
6. Support of communities of teachers, parents, students.
7. Ability to work on different devices - as a stand-alone application or in a device-adaptive mode (responsive).
8. Ability to switch to programming in another programming language.
9. Possibility to work at an individual pace - choice of topics, modules, exercises, etc.
10. Programming learning content covered by the environment.

Educational computer games and environments have the potential to create a motivating and fun learning environment that meets the educational standards and goals of computer science and information technology education in the modern Bulgarian school.

Conclusions: Traditional school learning is increasingly giving way to new technologies. In recent years, there has been a decline in the general motivation to learn, and this necessitates the application of non-traditional teaching methods. The effectiveness of the training as a whole depends on the effectiveness of the applied training methods. Undoubtedly, game-based learning and design thinking have an essential place in the education of modern children for the construction of knowledge, skills and competencies necessary for their realization today.

CHAPTER TWO. IT TRAINING IN THE BULGARIAN SCHOOL. THE ROLE OF TASKS IN TRAINING.

2.1. The beginning

The study of the Informatics subject in secondary schools in Bulgaria dates back to the 1960s, when classes with extended study of mathematics were created in the larger cities of the country. (Garov, 2000)

The first Mathematical Gymnasiums were established in the early 1970s, where informatics education was initially conducted in the form of optional classes outside of compulsory education.

In 1983, a reform was carried out in secondary education, according to which in the first level of education up to the 10th grade the general education of students is carried out, in the second level 11th - 12th grade a profession is mastered in the so-called Educational and professional complexes (UPK), and for students from Mathematical high schools and some technical schools, training for the profession "EIM operator-programmer" has been introduced. The same year in MG "Acad. K. Popov" - Plovdiv opens the first computer office in the country. A period of rapid computerization of schools followed. Thus, in the mid-1980s, informatics entered the secondary school in various forms, but was unevenly represented in the different classes. (Garov, 2000)

In 1986, the Ministry of National Education introduced "Informatics" as a separate subject in the 10th and 11th grades of secondary general education schools using the Basic language.

In the next few years, many changes are taking place in computer science curricula. From a drastic reduction in the hours of classes in the general compulsory preparation, through studying the subject only as a optional courses or mandatory elective training, to the complete elimination of informatics classes in some schools. (Dureva, 2003)

This was followed by a period (1994-1998) in which, after the introduction of standard curricula in informatics, the subject was studied as a compulsory subject, optional courses or mandatory elective training in SOU, SPTU and Technical Colleges, and in Mathematical High Schools and Economic Technical Colleges, the training in informatics has a profiling nature and a larger annual horary. In 1994, the study subject "Information Technologies" was introduced as compulsory. (Nikolova, 2019)

In 1998, the National Strategy for Information and Communication Technologies (ICT) came into force, which marked the beginning of a new reform in secondary education. Methods are being developed for the study of programming in secondary schools.

In 2000, the State educational requirements and standards for study programs were adopted, which determine the levels of general educational preparation of students at the end of the stage and level of education, the educational content of general educational preparation, as well as the requirements for educational content including general and specific key groups of skills, integral and cross-curricular areas of study content depending on the specifics of the study subjects and the cultural-educational areas (in particular Mathematics, informatics and information technologies, Appendix No. 3). (Ministry of Education, 2000)

The National Strategy for the introduction of ICT in Bulgarian schools lays the foundations for the integration and introduction of information and communication technologies in education. (Ministry of Education, 2005).

The following is the Strategy for the effective implementation of ICT in education and science of the Republic of Bulgaria (2014-2020), whose main goal "is to

ensure equal and flexible access to education and scientific information at any time and from any place - from a stationary computer, laptop, tablet, mobile phone. For the first time, a unified information environment serving school education, higher education and science is being created (Ministry of Education, 2014)

2.2. In the present

In 2016, the Law on Preschool and School Education (ZPUO) and new State Education Standards came into force. With Ordinance No. 4 on the curriculum (Ministry of Education, 2015) and Ordinance No. 5 on general educational preparation (CEO-Ministry of Education, 2015), significant changes are made to the curriculum and curriculum in informatics and information technologies. The subject "Informatics" in the first high school stage is studied only in the 8th grade as general education, and only in profiled classes with profiles "Mathematics", "Software and hardware sciences", "Economic development", "Natural sciences". In the 9th and 10th grades, the study of "Informatics" is not planned, and in the second high school stage - in the 11th and 12th grades, profiled training on the subject in four modules is provided. (Ministry of Education, 2016)

Another innovation in the curricula after the entry into force of the National Education and Training Program in 2016 is the introduction of the subject "Computer Modeling" in the initial stage of education - 3rd and 4th grade. The course curriculum includes basic competencies related to digital devices, information (data types), modeling in a visual environment for block programming (basic constructions and algorithms). At the lower secondary level, the subject is "Computer Modeling and Information Technology" and the areas of competence are computer systems, information, electronic communication, creation of digital content using the possibilities of block programming. In addition to the compulsory study hours, according to Ordinance No. 4 on the curriculum (Ministry of Education, Science and Technology, 2015), schools are given the opportunity to introduce optional study hours (ESH) and optional study hours (FAH).

2.3. Extracurricular forms of informatics

The first programming competitions date back to the early 1970s. The first National Competition was in 1979 and was held in the city of Sofia. Since 1982, the annual National Winter Mathematical Competitions (Ruse), Informatics

Competitions within the Spring Conferences of the SMB and Programming Competitions of the Central Station of Young Technicians have been started. Since 1985, these competitions have grown into the National Informatics Olympiad (NOI) (Azalov, 2009). In May 1987, an "Open Programming Competition" was held in Sofia, in which 28 students from 6 countries participated. (Kenderov, 2009)

At the XXIV General Conference of UNESCO in Paris, 1987, the representative of the Bulgarian delegation, Academician Blagovest Sendov, proposed to include in the Fifth Main Program (1988-1989) of UNESCO a point to establish an International Informatics Olympiad (IOM) (MMIB) . Thus, Bulgaria became one of the countries that founded the International Informatics Olympiad for students. In May 1989, in Pravets, the First International Informatics Olympiad was held. (Manev, 2009) (Azalov, 1989) (Kenderov & Maneva, 1989) (SMB). Interest in these competitions is increasing, and a network of clubs and schools for extracurricular activities in programming is being created in the country. In the following years and until now, students from Bulgaria have won numerous prizes in the editions of MOI. (Garov, 2000)

With the rapid development of computer sciences and their increasingly wide application in various economic spheres in recent decades, interest in the creation of software and multimedia products, web-based applications, databases has also increased. In 2002, the National Information Technology Olympiad (NOIT) was launched. A number of other competitions are also held annually - National Information Technology Competition, National Information Technology Spring Tournament, IT Expert, Webloz - National Web Design Competition, Hackathon-type competitions and others.

After 2016, with the introduction of the new curricula, high school students who do not study in the profiles "Mathematics", "Software and hardware sciences", "Economic development" and "Natural sciences" do not study informatics. In the compulsory study hours, only the subject "Information Technologies" is studied.

In a small part of schools, the subject "Informatics" can be studied as an optional or optional course, but the main alternative for studying it remains extracurricular forms.

2.4. Tasks in computer science education

A number of authors examine the place and role of tasks in informatics (Asenova, 1990), methodological problems in solving tasks in school courses in informatics and IT (Dureva, 2003), systems of supporting tasks are compiled for the preparation of students for participation in Olympiads and competitions in informatics and information technologies (Grozdev & Garov, 2008) (Garov & Todorova, 2006) (Garov, 2004) (Garov, 2000).

For the participation of students in the above-mentioned competitions, it is indisputable that the training in informatics and information technologies in school hours is extremely insufficient. Extracurricular forms are increasingly gaining ground as a form of programming education not only to prepare for competitions, but also as an opportunity to satisfy students' interests in computer science in general. Since 2018, the Ministry of Education and Science project "Activities by Interests" has been launched in schools. Another possibility, although insufficiently represented, are the Support Centers for personal development, where students from 2nd to 10th grade study programming (Plovdiv, Varna) and information technologies (Lovech, Plovdiv, Burgas, Blagoevgrad).

The main emphases of extracurricular activities in programming are aimed at the rapid acquisition of basic knowledge and its application in solving real-life tasks, "creating conditions for creative and algorithmic thinking, deepening and expanding informatics culture, forming skills and habits for independent mastering new knowledge, creating practical skills for constructing computer programs". (Grozdev & Garov, 2007)

The functions of the tasks that are applied in programming training in extracurricular forms are determined by the above-mentioned main emphases. In addition to the main functions (methodical, didactic, organizing and managing (Dureva, 2003), (Garov, 2010), quoting Smantzer, some specific ones can also be mentioned: forming an algorithmic style of thinking, mastering the basics of modeling, mastering the formalization of revealing the interrelationship between different forms of the same concept, phenomenon or process, tasks are a learning form of practice.

2.5. Stages in solving tasks in informatics

A number of authors unite around a common scheme of stages in solving programming tasks (Garov, 2010) (Grozdev & Garov, 2007) (fig. II-1):

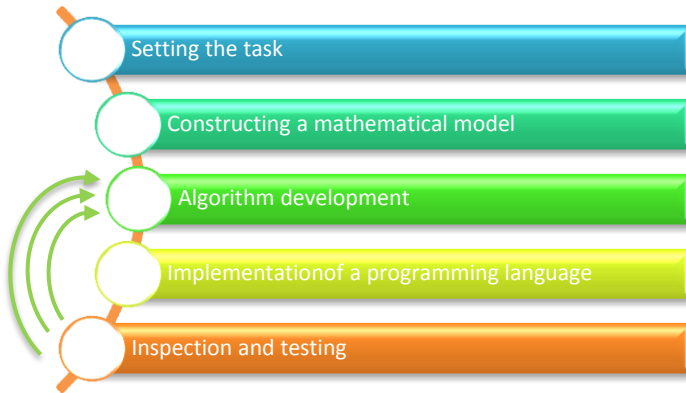


Figure II-1. *Stages in solving programming tasks*

Stage I - At this stage, the learning objectives of the task, the necessary knowledge and skills, as well as the age capabilities of the students must be clearly defined, so that each student can understand its formulation on his own.

II Stage – The task's place in a specific class of tasks is determined, its similarity (if any) with already solved similar tasks or already known solving methods are determined, knowledge in mathematics is updated.

Stage III – The general scheme of the algorithm is constructed, the sub-algorithms and constructions to be used are determined. Attempts are made to reduce the solution of the task (or part of it) to the application of already known methods

Stage IV – Implementation of the algorithm with a specific programming language. The input data is entered and the program is executed on a computer, various algorithms are applied, a creative approach is applied to computer problem solving.

V Stage - A check is made of the compliance of the algorithm with the conditions set in the task, whether there are syntactic or semantic errors, the program is tested with different input data and the results of its testing are tracked.

CHAPTER THREE. A MODEL FOR GAME-BASED PROGRAMMING LEARNING COMBINED WITH DESIGN THINKING TO DEVELOP ALGORITHMICAL THINKING

3.1. Model description

3.1.1. General scheme

The scheme is based on the workflow model of the collaborative design thinking platform Be-novative (<https://www.be-novative.com/>) and is adapted for game-based programming learning at high school level.

Conventionally, the scheme can be divided into two main, interconnected parts (Fig. III-1):

- Preliminary preparation, including main activities - teacher (Exposition, Dialogue, Observation, Evaluation), main activities - student (Practical tasks, Exercise, Independent work, Independent learning) and analysis of the situation;
- Implementation - application of the design thinking approach in the main stages of solving programming tasks (Challenge (game) → Sharing of ideas → Development of an algorithm → Implementation of the game (with the means of Python's Mu editor, Pygame mode → Verification and testing).

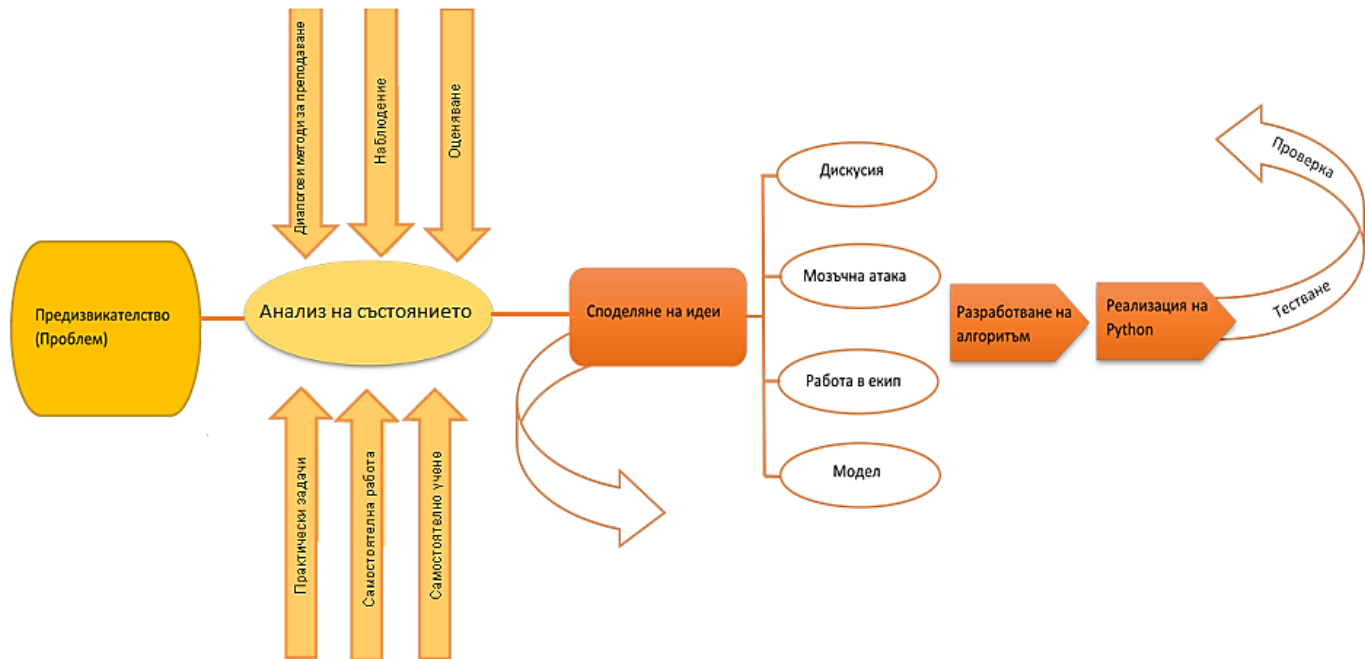


Figure III-1. An adapted model of programming learning based on design thinking

3.1.2. Implementation of the model

To implement the tasks, the Python programming language was chosen using turtle and Pygame libraries, in Pygame Zero mode in the Mu programming environment.

Python is a modern, interpretive, interactive, object-oriented, high-level programming language created by Guido van Rossum in the early 1990s². There is a comprehensive standard library and thousands of additional libraries in the Python Package Index, libraries of many other programming languages can also be used in Python, system integration, programmed applications can run on all operating systems for which there is a Python interpreter.

The proposed programming learning tasks were used in an extracurricular form of learning with 8th and 10th grade students who do not study informatics. The game approach was used from the point of view of learning programming by creating games. It is implemented using Python's Mu development environment in Pygame Zero mode.

Mu was originally created by the Python Software Foundation for the BBC's micro:bit project. Mu and is suitable for students taking their first steps in Python programming, runs on Windows, OSX, Linux and Raspberry Pi. Easy to use, available on all major platforms and well documented even for beginners³. Mu is free and can be easily downloaded and installed from the official Mu page⁴.

The turtle library is also used in some of the sample tasks. The turtle library provides graphical primitives for turtles in both object-oriented and procedural-oriented ways. By combining different commands, complex shapes and pictures can easily be drawn⁵.

3.1.3. Task types in the context of Bloom's Taxonomy

Taxonomy (from Greek taxis - placement, order, order, and nomos - law) is a theory for classifying and systematizing complexly organized areas of a given

² <https://brochure.getpython.info/media/releases/psf-python-brochure-vol.-i-final-download.pdf/view>

³ <https://mu.readthedocs.io/en/latest/>

⁴ <https://codewith.mu/en/download>

⁵ [Turtle. https://docs.python.org/3/library/turtle.html](https://docs.python.org/3/library/turtle.html)

activity. In particular, the Pedagogical Taxonomy represents a system of pedagogical goals, distributed by categories and hierarchical levels. Bloom's Taxonomy refers to the cognitive field of pedagogical taxonomies, in which cognitive goals are hierarchically located at six levels: Knowledge → Understanding → Application → Analysis → Synthesis → Evaluation. (Bizkov, 1995)

In (Teodosiev, 2013) programming-style learning is considered, using Bloom's Taxonomy to define programming learning goals for students. The comparison of the stages in learning the programming style and the levels in Bloom's pyramid (Teodosiev, 2013, p. 231) can be adapted and applied in programming education at the high school level (Fig. III-2).

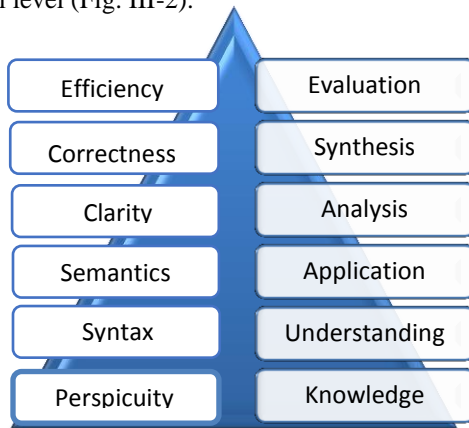


Figure III-2. *Programming learning process in Bloom's pyramid.*

3.2. Task system

The experimental programming learning used the constructivist game-based learning model, where students learn by creating games.

Table 2 presents the distribution of study hours for each of the topics in the training course.

Table 2. *Distribution of hours by topic*

Topic	Lesson	Number of hours on the topic
Topic 1: Python's Mu Editor. Graphics with turtle library.	Getting to know Python's Mu editor. Drawing figures.	4 hours
	Colors and filling shapes. I draw with a turtle.	4 hours
	Drawing more complex shapes. User functions.	6 hours
Topic 2. Creating 2D games with Python's Pygame library.	First games with Pygame.	8 hours
	Work with images, characters, sounds and music. Animation.	4 hours
Topic 3. Software project - creating a 2D game.	Create a 2D game.	10 hours
Total hours:		36 hours

A methodological development of the lesson and a system of tasks, including detailed descriptions of the stages in solving the task and the syntax and semantics of the program constructions used in it, are proposed for each of the topics and the corresponding lessons. The specific application of design thinking as an approach to solving some of the tasks is also indicated. The tasks are used both for learning new learning material and for consolidating already acquired knowledge and skills.

The system of tasks is described in detail in paragraph 3.2. and includes 25 tasks, with each task numbered and including a task condition, an example Python implementation, and example output from it. Methodological guidelines for solving them are also described.

In paragraph 3.2. Internet addresses of self-learning platforms related to the Python language, the turtle and Pygame libraries, and the Python Mu editor are also given. Also listed are addresses of image galleries that were used in some of the assignments, as well as an address of an application for creating your own images that is freely distributed.

CHAPTER FOUR. EMPIRICAL STUDY OF THE MODEL

4.1. Description of the empirical study.

The experimental game-based training in programming with design thinking as an approach was carried out with students from the 8th and 10th grades in the PGSAG "Vasil Levski" - Blagoevgrad during the academic year 2021/2022. Students have not studied the subjects "Computer Modeling" and/or "Informatics". 16 students participate in the experimental group (EG). The control group (CG) consists of 17 students. The game-based learning model was applied to programming education in the experimental group in extracurricular form - Digital World Programming Club.

For the purposes of the study, the following research toolkit was created:

- Test to establish the entry level of students;
- Survey to establish the influence of the task system and student motivation;
- Test to establish the students' exit level;
- Practical task for evaluating the achievements of students;

The empirical study went through the following stages:

I. Declarative experiment – formation of experimental and control groups and establishment of the entry level of students' algorithmic thinking and their digital competences in both groups.

II. Actual experiment - Empirical testing of the model

III. Final experiment – establishing the baseline level of algorithmic thinking and digital skills in the two experimental groups.

In paragraph 4.1., in Taylor matrices, the entry and exit level test specifications of Bloom's Taxonomy are presented, tables describing the objectives for the entry and exit level questions, as well as criteria for evaluating the practical task used to measure achievement of the students of the experimental group (EG).

4.2. Statistical methods for the analysis of the results.

Data analysis was done with IBM SPSS Statistics, JASP and MS Excel. The following non-parametric statistical methods were applied with a significance level of 0.05 - Mann-Whitney, Kolmogorov-Smirnov and Wilcoxon.

Study the reliability of the input and output level

The Spearman-Brown prophecy formula was used to check the reliability of the input and output levels. It was found that at the entry level ($r = 0.615$) the reliability is high, and at the exit level ($r = 0.474$) it is medium.

4.3. Results analysis

4.3.1. Analysis of input results

The entrance level test is composed of two parts - Digital Competences and Algorithmic Thinking. Some of the questions in it are from the Higher Education Institution Digital Competences - 2015, 2016, 2018, 2020. Statistical methods were applied separately and jointly to both parts of the test, with the focus of the study being on the "Algorithmic Thinking" part.

The students who participated in the two groups of the study did not study the subjects "Computer Modeling" and/or "Informatics".

For the purposes of the study, the main hypotheses H_0 are: Using the proposed game-based programming learning model combined with design thinking at the high school level will lead to:

1. Developing students' algorithmic thinking;
2. Increasing the general digital competence of students;
3. Increasing students' interest and motivation;

A Kolmogorov-Smirnov test with a statistically significant Lilliefors level was applied to check the normality of the input level results. Since the number of students was less than 100, the Shapiro-Wilk test was also applied. (Ganeva, 2016)

The hypotheses are tested:

H_0 – The data have a normal distribution

H_1 – The data do not have a normal distribution

The distributions in both groups were found not to follow a normal distribution. The null hypothesis H_0 - "The distribution is normal" is rejected with a significance level of 0.05 in both groups.

The results of the normality check are presented in Fig. IV-1.

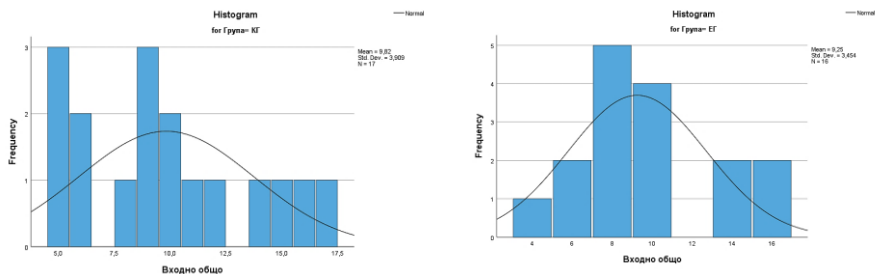


Figure IV-1. Results of a test for normality of distributions

Comparison of the entrance test results of the control group (CG) and the experimental group (EG).

Since both groups are not normally distributed and due to the fact that the number of subjects in each group is less than 30, the Mann-Whitney U test was applied. This test is one of the most popular non-parametric testing tests, of two independent samples, comparing the medians of the two groups. (Ganeva, 2016)

The results of the test give reason to accept with a significance level of 0.05 the hypothesis H_0 - The two groups statistically do not differ in terms of the achievements measured with the input level (**Asimp. Sig. (P) = 0.677 > 0.05**)

4.3.2. Analysis of baseline results

At the end of the empirical study, a final assessment of the students' achievements was made - an exit test and a practical task.

The Kolmogorov-Smirnov and Shapiro-Wilcoxon statistical tests for determining the normality of the distributions, the Friedman test for three or more dependent samples and the Mann-Whitney test for the comparison of independent samples were applied to analyze the output results.

Comparison of input and output test results

Both the entrance and exit tests are divided into two parts - "Digital Competencies" and "Algorithmic Thinking". Statistical tests were applied for the overall results for the entrance/exit test and for the individual criteria – Knowledge+Comprehension, Application and Analysis.

Comparison of exit and entry test results in the experimental group (EG)

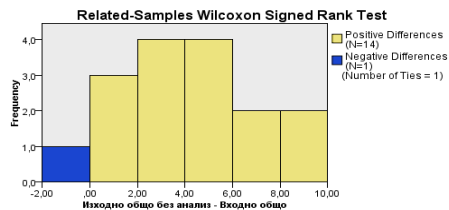
The Wilcoxon signed ranks test was used to compare the results of the initial and initial test of the experimental group (EG). This test is the non-parametric alternative to the t-test, which instead of comparing arithmetic means, converts the values of the variable to ranks and compares them. The test calculates the number of positive and negative differences and makes a decision based on the number and size of their difference. (Ganeva, 2016)

The hypotheses tested with the Wilcoxon signed rank test:

H_0 - The probability of positive differences is equal to the probability of negative differences, i.e. differences are not statistically significant;

H_1 - The probability of positive differences is different from the probability of negative differences, i.e. differences are statistically significant;

In the comparison of achievements in the parts of the test "Algorithmic thinking" - Knowledge + Application (**Asimp. Sig. = 0.016<0.05**) and Application (**Asimp. Sig. = 0.03<0.05**), as well as in the comparison of the overall results of the output and the input level in EG, with a significance level of 0.05 the null hypothesis H_0 is rejected (**Asimp. Sig. = 0.001<0.05**). This means that there is an increase in the results of these criteria in the "Algorithmic Thinking" part, as well as in the overall results of the experimental group (EG). (Fig. IV-2).



Total N	16
Test Statistic	118,000
Standard Error	17,575
Standardized Test Statistic	3,300
Asymptotic Sig. (2-sided test)	,001

Figure IV-2. Wilcoxon test - output no analysis/input total score in EG

To compare the results in the experimental group (EG) of the practical task, the parts entry level – algorithmic thinking, output level – algorithmic thinking (Knowledge + Application) and by criterion – Analysis, the Friedman test was used for three or more dependent samples from rank data, which is the non-parametric alternative to univariate analysis of variance for correlated samples (Ganeva, 2016). The number of points for the practical task, entry level - algorithmic thinking and exit level - algorithmic thinking is different. For this reason, the values are normalized in percentages (Fig. IV-3)

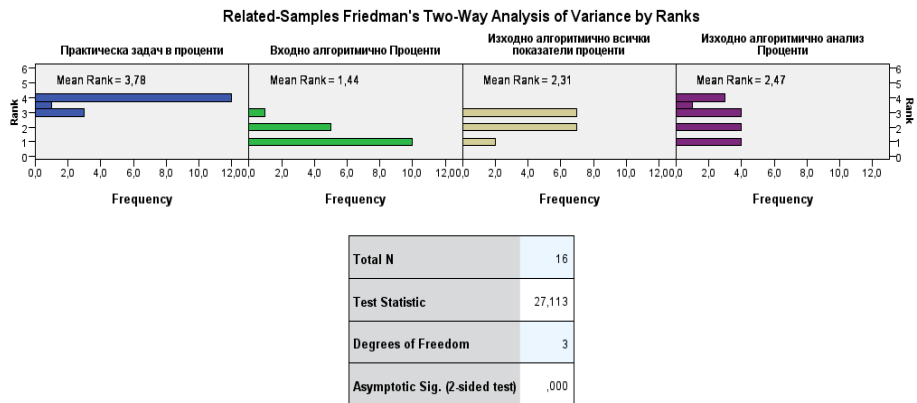


Figure IV-3. *Statistics of the results - Friedman's test*

Kendall's coefficient of concordance W - a non-parametric statistical test that is used to measure the statistical relationship between multiple objects (outcomes) - was also investigated. It takes values from 0 to 1. When all statements match, it is equal to 1, and when the differences are large, then $W=0$. (Dimitrov & Karashtranova, 1993) The obtained results, as well as the value of the coefficient $W=0.565$, gives reason to assume that there is a statistically significant difference in the results, which necessitates the conclusion:

The achievements of the students of the experimental group (EG) in the part of the test "Algorithmic thinking" are significantly higher. This was significantly reflected in the practical application of the acquired knowledge and skills in solving the practical task, and the main hypothesis of the study is confirmed, namely: The proposed model develops algorithmic thinking.

Comparison of baseline and baseline results of the control group (CG) and the experimental group (EG)

The non-parametric Mann-Whitney test for two independent samples was used to compare baseline and baseline results of the control (CG) and experimental (EG) groups. The results of the comparison of the baseline results of the control group (CG) and the experimental group (EG) are presented in the table. 3:

Table 3. Comparison of CG and EG baseline results

Criteria	Result
Digital competences – Knowledge+Understanding	There is no statistically significant difference
Digital competences – Application	There is no statistically significant difference
Digital competences - General	There is no statistically significant difference
Algorithmic thinking – Knowledge + Application	There is a statistically significant difference. Higher scores in EG.
Algorithmic Thinking – Application	There is a statistically significant difference. Higher scores in EG.
Algorithmic Thinking – Analysis	There is a statistically significant difference. Higher scores in EG.
Algorithmic thinking – Common with analysis	There is a statistically significant difference. Higher scores in EG.
Baseline - Overall score	There is a statistically significant difference. Higher scores in EG.

Conclusions: There is no statistically significant difference in the results of CG and EG from the initial level in the part "Digital Competences". As a logical reason for this result, the study of the subject Information Technologies - general educational preparation in both groups from 8th to 10th grade can be indicated. In the part of the test "Algorithmic thinking" - according to the individual criteria and the overall result, the achievements of the students of the experimental group were significantly higher than those of the control group. This leads to the conclusion that game-based learning combined with design thinking can lead to the development of algorithmic thinking.

4.3.3. Analysis of surveys to establish the influence of the task system in game-based learning and student motivation

The analysis is based on the results of a standardized survey of 16 students from the experimental group. Scaled responses with a 5-point scale (rated rating scale) were used. (Bizkov, 1995) Cronbach's coefficient alpha is used to check the

reliability of the survey. "If the value of the coefficient is greater than 0.70, it is assumed that the scale is reliable. "If the scale consists of a small number of questions, α values around 0.50 can be considered normal" (Ganeva, 2016). The survey reliability test result $\alpha = 0.765$ indicates that the survey is reliable.

The standard deviation of each of the survey questions was examined. The analysis shows that the use of tasks with an interesting condition, in which creativity can be shown, increases the interest and activity of students. The proposed tasks are suitable for learning and applying in practice knowledge of programming with the Python language, Pygame module with Mu editor.

CONCLUSION

The results of the research conducted in accordance with the purpose and tasks of the dissertation lead to the following conclusions:

• **Working hypotheses are accepted: Using the proposed game-based programming learning model combined with design thinking will lead to:**

- Developing students' algorithmic thinking;
- Stimulating students' interest and motivation;
- Applying the model leads to:
 - higher overall test scores;
 - higher results according to criteria - Knowledge + Application in the part "Algorithmic thinking";
 - higher scores by criterion - Application in part "Algorithmic thinking"
 - higher results by criterion - Analysis in the "Algorithmic Thinking" part;
 - higher overall results in the "Algorithmic Thinking" section;
 - higher results when working on the practical task.
- The use of tasks with an interesting condition, in which creativity can be manifested, have a positive impact on the interest and motivation of students during programming education;
- Students prefer to be provided with ready-made function templates for creating the task algorithm;

The working hypothesis is partially rejected: Using the proposed game-based learning model combined with design thinking will lead to an increase in students' digital competence:

- higher overall results in the "Digital competences" part in the experimental group;
- higher results by criterion - Application in the "Digital competences" part in the experimental group;
- There is no change in the results according to criteria - Knowledge and Knowledge+Understanding in the part "Digital competences" in the experimental group;
- There is no statistically significant difference between the control and experimental groups in the "Digital Competences" section
- A model for game-based programming learning combined with design thinking was created.

The achievements of the students, as a result of testing the model, give us reason to continue our research in this direction. It is possible to create a model for continuing education in programming using the Python language and Pygame library.

CONTRIBUTIONS

Scientific and applied:

1. A study of environments for game-based learning in informatics was made
2. The possibilities for combining game-based programming training with the design thinking approach have been analyzed
3. A model for game-based programming learning combined with design thinking was developed;

Applied:

1. A game-based task system has been developed;
2. The developed model for game-based training in programming, combined with design thinking, has been tested in an extracurricular form of training - the "Digital World" hobby club with 8th and 10th grade students.

PUBLICATIONS

1. Georgieva, R., Tuparova, D., Educational computer game environments for teaching programming and developing algorithmic thinking - a comparative analysis, 48th Spring Conference of the Union of Mathematicians in Bulgaria, Borovets, 2019, ISSN 1313-3330
2. Tuparova, D., Georgieva, R., Design thinking in support of learning, Jubilee International Conference, Pamporovo, 2020, ISBN 978-619-202-595-3
3. Georgieva, R., Game-based training in programming for beginners in Pygame Zero mode - example tasks with Python turtle graphics, Mathematics and Informatics, Sofia, 2023, 66(2), <https://doi.org/10.53656/math2023-2-6-gam>, ISSN 1314-8532 (online) 1310-2230 (print), pp. 170-186

Bibliography

- Aleksić, V., Ivanovic, M., Popesco, E., & Budimac, Z. (2016). Commercial Off-the-Shelf Games as Learning Media. *17th International Conference on Computer Systems and Technologies 2016 - CompSysTech '16*, 355-360. Palermo, Italy.
- Hakak, S., Fazmidar, N., Noor, M., Nizam, M., Affal, H., Hussin, N., Imranc, M. (2019). Cloud-assisted gamification for education and learning – Recent advances and challenges. *Computers & Electrical Engineering*, 22-34. Downloaded on <https://www.sciencedirect.com/science/article/abs/pii/S0045790618304968>
- IDEO. (07 09 2008 r.). Definitions of design thinking. Downloaded on 04 02 2022. from <https://designthinking.ideo.com/blog/definitions-of-design-thinking>
- Inno Spark. (2015). Compendium of good practices and innovations. Downloaded on 04. 02.2022 from http://innospark-ict.eu/uploads/InnoSpark-IO4-Compendium_BG.pdf
- Innovation Starter. (04 06 2015 r.). Design Thinking (Design Thinking). Downloaded on 04 02 2022. from <http://innovationstarterbox.bg/resources/kakvo-e-dizain-mislene/>
- Kenderov, P. (2007). *Bulgaria – Birthplace of International*. Downloaded on http://www.math.bas.bg/talents/en/inf/IOI_paper1.pdf:
http://www.math.bas.bg/talents/en/inf/IOI_paper1.pdf
- Kenderov, P., & Maneva, N. (1989). International Olympiad in Informatics. *The International Olympiad in Informatics* (срп. 54). Sofia: Union of the Mathematicians in Bulgaria.
- Kirriemuir, J., & McFarlane, A. (2004). Literature Review in Games and Learning. *NESTA Futurelab series: report 8*, 1-38. Downloaded on

- https://www.researchgate.net/publication/32231341_Literature_Review_in_Games_and_Learning
- Naiman, L. (20 05 2019 r.). *Design Thinking as a Strategy for Innovation*. Downloaded on 04 02 2022. <https://www.creativityatwork.com/design-thinking-strategy-for-innovation/>
- Pelling, N. (2011). *The (short) prehistory of “gamification”*, Downloaded on <https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>
- Red Paper Plane. (2022 Downloaded on <https://bg.rpplane.com/schools>
- Stanford. (2018). Downloaded on <https://web.stanford.edu/dept/SUSE/taking-design/presentations/Taking-design-to-school.pdf>
- Stanford. (2019). Downloaded on https://web.stanford.edu/dept/SUSE/taking-design/proposals/Destination_Imagination_the_Fire_Within.pdf
- Stanford. (2022). *Design Thinking Bootcamp: Make Impact and Drive Growth in Your Organization*. Downloaded on 04 02 2022, <http://www.gsb.stanford.edu/exec-ed/programs/design-thinking-bootcamp>
- Trybus, J. (2015). *Game-Based Learning: What It Is, Why It Works, and Where It’s Going*. Downloaded on New Media Institute: <https://web.archive.org/web/20150911221236/http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html>
- Tuparov, G., Keremedchiev, D., Tuparova, D., & Stoyanova, M. (2018). Gamification and educational computer games in open source learning management systems as a part of assessment. *17th International Conference on Information Technology Based Higher Education and Training (ITHET)* (срр. 1-5). Olhao, Portugal : IEEE.
- Tuparova, D., Tuparov, G., Veleva, V., & Nikolova, E. (2018). Educational computer games and gamification in informatics and information technology education - Teachers' points of view. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (срр. 702-709). IEEE. Downloaded on https://www.researchgate.net/publication/326699762_Educational_computer_games_and_gamification_in_informatics_and_information_technology_education_-_Teachers%27_points_of_view
- Azalov, P. (1989). First International Informatics Olympiad. *Mathematics(7)*, 41-45.(in Bulgarian)
- Azalov, P. (2009). The Olympiads in Informatics - facts and events that are still remembered. *Mathematics and Informatics(1)*, 3-18. (in Bulgarian)
- Asenova, P. (1990). Construction and use of a task system in an algorithmization course. Moscow: Russian Academy of Science. (in Bulgarian)

- Bizkov, G. (1995). Methodology and methods of pedagogical research. Sofia: ASKONI-IZDAT. (in Bulgarian)
- Ganeva, S. (2016). Let's reinvent statistics with IBM SPSS Statistics. Downloaded from <https://www.elbook.eu/images/book25.pdf>(in Bulgarian)
- Grozdev, S., & Garov, K. (2007). Main learning activities in preparing students for participation in informatics olympiads. Science, education and time as care. *Jubilee scientific conference with international participation*. Smolyan. (in Bulgarian)
- Grozdev, S., & Garov, K. (2008). On the systems of support tasks in the preparation for participation in olympiads in informatics. *Combinatorial objects and algorithms. 37 SMB Spring Conference*, (pp. 304-311). Sofia. (in Bulgarian)
- Garov, K. (2000). Education in informatics and information technologies in secondary school - status and perspectives. *Anniversary scientific session - 30 years FMI, PU "Paisiy Hilendarski"*, Plovdiv. Plovdiv. (in Bulgarian)
- Garov, K. (2004). A system of support tasks in the preparation of talented and distinguished students for participation in Olympiads and competitions in informatics. *33 SMB Spring Conference*, (pp. 316-321). Sofia. (in Bulgarian)
- Garov, K. (2010). Tasks in informatics and information technology education. *National conference "Education in the information society*. Plovdiv. (in Bulgarian)
- Garov, K., & Todorova, E. (2006). An example system of supporting tasks on the topic "Algorithms and tasks from number theory" for the preparation of talented students in informatics. *35 SMB Spring Conference*. Borovets. (in Bulgarian)
- Dimitrov, B., & Karastranova, E. (1993). *Statistics for non-mathematicians*. Blagoevgrad: Neofit Rilski Secondary School, Blagoevgrad. (in Bulgarian)
- Dureva, D. (2003). *Problems of the methodology of training in informatics and information technologies*. Blagoevgrad: Neofit Rilski University Publishing House. (in Bulgarian)
- Dureva, D., & Kaseva, M. (2011). Computer-based educational games in primary school education. Information and communication technologies, media and education. Blagoevgrad. (in Bulgarian)
- Kenderov, P. (2009). Bulgaria - the birthplace of the International Informatics Olympiad for students. *Mathematics and Informatics*, vol. 3, 3-7. (E. Sendova, Interviewer) Sofia. (in Bulgarian)
- Manev, K. (2009). The International Informatics Olympiad - the "Bulgarian" Olympiad. *Mathematics and Informatics*(2), 3-11. (in Bulgarian)
- MMIB. (n.d.). International Informatics Olympiads (1989-2008). Downloaded from http://mmib.math.bas.bg/?page_id=20788 (in Bulgarian)
- MON. (2000). State educational requirements for curriculum content. Downloaded from <https://mon.bg/bg/100105>. (in Bulgarian)

- MON. (2005). *National strategy for the introduction of ICT in Bulgarian schools*. Downloaded from <https://www.strategy.bg/StrategicDocuments/View.aspx?lang=bg-BG&Id=398>: (in Bulgarian)
- MON. (2014). *Strategy for effective application of information and communication technologies in education and science of the Republic of Bulgaria (2014-2020)*. Downloaded from <https://www.strategy.bg/StrategicDocuments/View.aspx?Id=904>. (in Bulgarian)
- MON. (2015). *Ordinance No. 4 on the curriculum*. Downloaded from <https://mon.bg/bg/59>. (in Bulgarian)
- MON. (2016). *Ordinance No. 7 on profiled training*. Downloaded from <https://mon.bg/bg/59>. (in Bulgarian)
- MON. (2018). *Study programs for profiled training*. Downloaded from the Ministry of Education and Science: <https://www.mon.bg/bg/100598> (in Bulgarian)
- MON. (2018). *Computer modeling curriculum for third grade (general education)*. Downloaded from the Ministry of Education and Science: <https://www.mon.bg/bg/1689> (in Bulgarian)
- MON. (2019). *Computer Modeling Curriculum for Fourth Grade (General Education)*. Downloaded from the Ministry of Education and Science: <https://www.mon.bg/bg/2190> (in Bulgarian)
- MON. (2020). *Curriculum for computer modeling and information technologies for class V*. Downloaded from the Ministry of Education and Science: <https://www.mon.bg/bg/100884> (in Bulgarian)
- Nakov, S., & Collective. (2017). *C# Programming Basics*. Faber. Downloaded from <https://raw.githubusercontent.com/SoftUni/Programming-Basics-Book-CSharp-BG/master/resources/Programming-Basics-CSharp-v2017.pdf> (in Bulgarian)
- Nikolova, E. (2019). *An integrative approach to informatics education at the high school level*. Blagoevgrad. (in Bulgarian)
- Petrov, A., Valkanova, V., & Dimitrov, I. (2016). *Game-based learning in a virtual educational space*. *Jubilee Scientific Conference with International Participation* (pp. 451-456). Plovdiv: Paisii Hilendarski University of Plovdiv. Downloaded from http://research.bfu.bg:8080/jspui/bitstream/123456789/909/1/451_PDFsam_Final_Tom2.pdf (in Bulgarian)
- Radoslavova, S. (27 04 2018 r.). *Learning as pleasure, provocation of creativity and play*. Downloaded on 31 01 2022. from https://www.capital.bg/specialni_izdaniia/obrazovanie/2018/04/27/3375246_uche_netto_kato_udovolstvie_provokacija_na_kreativnostta/ (in Bulgarian)

- SMB. (n.d.). International Informatics Olympiads. Downloaded from http://mmib.math.bas.bg/?page_id=20786. (in Bulgarian)
- Spirova, M. (2018). Place of game-based learning in school. *XI National Conference "Education and Research in the Information Society"*, (pp. 41-46). Plovdiv. Downloaded from <http://webcache.googleusercontent.com/search?q=cache:MDvQKGmK58MJ:sci-gems.math.bas.bg/jspui/bitstream/10525/2945/1/ERIS2018-book-p04.pdf+&cd=1&hl=bg&ct=clnk&gl=bg&client=firefox-b-d> (in Bulgarian)
- Tabakova-Komsalova, V. (2018). Formation of algorithmic thinking in primary school and junior high school students through informatics and information technology training. Dissertation work. Plovdiv: Paisii Hilendarski University of Plovdiv. (in Bulgarian)
- Teodosiev, T. (2013). Bloom's Taxonomy and Programming Style Learning. *VI National Conference "Education in the Information Society"*, (pp. 227-236). Downloaded from <http://sci-gems.math.bas.bg/jspui/bitstream/10525/2333/1/EIS2013-book-p23.pdf> (in Bulgarian)
- Terzieva, T. (2010). Research results on the concept of algorithmic thinking. . Jubilee International Scientific Conference *"Synergy and Reflection in Mathematics Education"*. September 10-12 (pp. 479-486). Bachinovo: Faculty of Mathematics and Informatics, PU "Paisiy Hilendarski". Downloaded from <http://fmi-plovdiv.org/GetResource> (in Bulgarian)
- Terzieva, T. (2012). Development of algorithmic thinking in informatics education. Dissertation work for awarding the educational and scientific degree "Doctor". Plovdiv: Paisii Hilendarski University of Plovdiv. (in Bulgarian)
- Terzieva, T., Golev, A., & Stavrev, S. (2017). Serious games – an innovative learning tool. *"Innovative Software Tools and Technologies with Applications in Research in Mathematics, Informatics and Learning Pedagogy"*, (pp. 107-114). Pamporovo. (in Bulgarian)
- Tuparova, D., & Georgieva, R. (2019). Educational computer game environments for teaching programming and developing algorithmic thinking - a comparative analysis. *48 SMB Spring Conference*, (pp. 150-155). Borovets. (in Bulgarian)
- Tuparova, D., & Georgieva, R. (2020). Design thinking to aid learning. *Jubilee international scientific conference "Synergy and reflection in the teaching of mathematics"*. Pamporovo: University Publishing House "Paisiy Hilendarski".
- UNIBIT. (2022). *Mission and goals*. Downloaded on 03 02 2022. from <https://www.unibit.bg/about-unibit/overview/mission-goals> (in Bulgarian)
- CIOO - MON. (2015). *Ordinance No. 5 on general education*. Downloaded from <https://cioo.mon.bg/>: https://cioo.mon.bg/wp-content/uploads/2014/07/nrd5_30.11.2015_obshtoobr_podgotovka.pdf (in Bulgarian)