

**ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „ОБУЧЕНИЕ ПО МАТЕМАТИКА,
ИНФОРМАТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ”**

МАРИАНА ИВАНОВА КРУШКОВА

**МЕТОДИКА ЗА АКТИВНО ОБУЧЕНИЕ
ПО ПРОГРАМИРАНЕ ЧРЕЗ ИЗПОЛЗВАНЕ
НА ИНФОРМАЦИОННИ
И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ**

А В Т О Р Е Ф Е Р А Т

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 Асен Рахнев

Пловдив, 2014 г.

**ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „ОБУЧЕНИЕ ПО МАТЕМАТИКА,
ИНФОРМАТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ”**

МАРИАНА ИВАНОВА КРУШКОВА

**МЕТОДИКА ЗА АКТИВНО ОБУЧЕНИЕ
ПО ПРОГРАМИРАНЕ ЧРЕЗ ИЗПОЛЗВАНЕ
НА ИНФОРМАЦИОННИ
И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ**

А В Т О Р Е Ф Е Р А Т

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 Асен Рахнев

**Рецензенти: проф. д.п.н. Сава Гроздев
проф. д-р Коста Гъров**

Пловдив, 2014 г.

Дисертационният труд е обсъден и насочен за защита на разширено заседание на катедра „Обучение по математика, информатика и информационни технологии” при Факултета по математика и информатика на ПУ „Паисий Хилендарски“.

Дисертационният труд съдържа 145 страници, от които основна част – 116 и 15 приложения. Използваната библиография включва 140 източника, от които 48 заглавия на кирилица, 82 заглавия на английски език и 10 интернет страници. Приложенията включват анкети, примерни задачи и тестове.

Списъкът на авторските публикации се състои от 6 заглавия.

Защитата на дисертационния труд ще се състои на 28.11.2014 г. от 11:00 ч. в Заседателната зала на новата сграда на ПУ „Паисий Хилендарски“, гр. Пловдив.

Материалите по защитата са на разположение на интересувашите се в секретариата на ФМИ, нова сграда на ПУ, каб. 330, всеки работен ден от 8:30 до 17:00 часа.

Автор: Мариана Иванова Крушкова

Заглавие: „Методика за активно обучение по програмиране чрез използване на информационни и комуникационни технологии “

Университетско издателство „Паисий Хилендарски“

Пловдив, 2014 г.

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Издигането нивото на българското висше образование е една от стратегическите цели на страната ни като равноправен член на европейското семейство. Въвеждането на нови научни методи и съвременни технически средства в процеса на обучение са важна предпоставка за реализирането на тази цел. Програмирането е една от основните дисциплини, чието изучаване е съществен принос в създаване на икономика, основана на знание. Затова тя се изучава в няколко факултета на Пловдивски университет "Паисий Хилендарски". Поради различни причини, голяма част от студентите бавно и трудно овладяват тази дисциплина [17, 23, 27]. Алгоритми, които за напредналите в тази наука таланти [9] изглеждат елементарни, затрудняват голяма част от начинаещите. Това важи особено за студенти от направления като химия, биология, педагогика [19], в които програмирането не е основна дисциплина. Проблеми срещат даже студенти от специалности в направление Информатика и компютърни науки. Особено проблемна е първата година, през която наред с попълване на „дупките“ от предходната образователна степен, трябва да се усвои и новият материал. Наличието на абстрактни структури и понятия в информатиката затруднява студентите.

Трудното начало обезкуражава такива обучаеми и въпреки популярността на дисциплината, те изгубват интерес към нея. В подобни случаи дори и за най-умелия преподавател е цяло изпитание да грабне вниманието в зала, пълна с такива студенти и да го задържи до края на занятието. Възниква нуждата от нови методи и подходи, с които да бъде предизвикан интересът на обучаемите към учебния материал.

Премине ли преподавателят към суха теория, веднага разсеяността в залата се повишава. Вместо това, той може да наблегне на примерите, които дават на теорията по-реални очертания. Един добре подбран и атрактивно представен пример ще бъде по-лесно запомнен и усвоен, отколкото правила и дефиниции, които ще бъдат зазубрени за момента и скоро забравени. Може да се провокират студентите да предложат собствени подобни примери, основавайки се например на метода на аналогията [18]. Стимулирането на обучаемия да бъде активна вместо пасивна част от обучението е едно от полезните и трудни за реализиране действия от страна на преподавателя. В методиката на обучението са описани различни похвати за привличане интереса на обучаемите, като един от най-популярните е участието в образователна игра. Тя има за цел да обогати и задълбочи знанията в дадена област, да помогне на участниците да придобият някакво умение или да проумеят по-добре дадено събитие. Тя дава възможност и за състезание между обучаемите. На такъв принцип са преодолени проблемите във Факултета по Информатика на Университета на Мурсия (Испания) [57]. Голяма част от студентите отпадат по време на следването си поради сложността на материята и липсата на мотивация сред тях. В цитирания университет редица години до втори курс достигат около 30% от първокурсниците. За да се избегне този проблем, авторите разработват методология, базирана на две ключови идеи: замяната на изпитните сесии със серия от оценявания по време на семестъра, като самите изпити се заместват от онлайн състезания, които се оценяват от „електронен съдия“ и са значително по-интересни и мотивиращи. Резултати от проучвания сочат висока ефективност на този подход – във втори курс продължават над 55% от студентите!

Повишаване на мотивацията на обучаемите е важен педагогически похват, който лекторите често пренебрегват. От съществено значение е, студентите да разберат защо им е полезна изучаваната дисциплина, има ли реално приложение в живота, би ли им свършила работа в професията. Обучаемите учат с огромно желание, когато са уверени в приложимостта на усвоените компетентности. Вниманието им може да бъде привлечено, а ентузиазмът им увеличен, като тестват и анализират готови продукти, написани на езика за програмиране, който изучават. Формалните и математически модели, които стоят в основата на такива продукти, провокират интереса им към изучаването на математика и нейните приложения. Всичко това непосредствено доказва, че изучаваният материал е съвременен и

приложим. Трябва да бъдат убедени, че могат занапред сами да реализират подобни продукти, работейки в екип.

В последните години, педагогически експерименти, представени в различни литературни източници, доказват огромната полза от активното обучение (active learning). То включва различни подходи, обединени от едно – поставянето на обучаемия в центъра на образователния процес като активна единица. При него студентът не е пасивен регистратор на лекторските идеи, а активен участник в обучението, което се превръща в обмен на идеи. Подходите включват участие в различни видове образователни игри, експериментиране със симулационни модели, използване на компютърни системи за електронно обучение, проектно-базирано обучение, работа в екип и др. Въпреки че се прилага в различни области [84], най-голям дял то има в природните науки, математиката и информатиката. Бурното развитие на ИКТ, интернет и социалните мрежи допринесе изключително за разширяване на възможностите за активно обучение.

Трудностите, които студентите срещат, намаляващият с всяка изминала година капацитет на студентите за усвояване на знанията и възможностите, които предлагат различните подходи за активно обучение, ни насочиха към педагогическо изследване.

Обект на изследване – учебно-познавателната дейност на студентите в обучението им по програмиране.

Предмет на изследване – възможностите за повишаване успеваемостта и мотивацията на студентите по отношение усвояване на учебното съдържание в обучението им по програмиране.

Цели и задачи на дисертационния труд

Основна цел на дисертационния труд е да се разработи методика за провеждане на активно обучение по базови дисциплини по програмиране и инструментариум за нейното прилагане, които да повишат мотивацията и успеваемостта на студентите.

За реализиране на тази цел е необходимо да бъдат решени следните **задачи**:

1. Да се анализират учебните програми и моделът на обучение по базови дисциплини по програмиране.
 - Да се изследват основните трудности, които студентите срещат при усвояване на базовите дисциплини по програмиране.
 - Да се проучи самоподготовката на студентите и предоставеният им инструментариум за улесняване на нейното провеждане.
 - Да се проучи нагласата на студентите за активно участие в образователния процес и актуалното състояние на активното обучение по програмиране.
 - Да се проучат литературни източници, разглеждащи прилагането на модела на активно обучение в различни направления и по-конкретно в областта на програмирането.
2. Да се разработи дидактически модел и методически инструментариум за провеждане на активно обучение в базовите дисциплини по програмиране.
 - Да се анализират предимствата на активното обучение и се разработи методика за прилагането му при усвояването на базови дисциплини по програмиране.
 - Да се разработят подходящи компютърни методи и средства за подпомагане на модела за активно обучение.
3. Да се проведат педагогически експерименти за определяне на ефективността на предложената методика и разработения инструментариум.
 - Да се разработят критерии и показатели за оценка ефективността на отделните елементи на методиката.
 - Да се създадат и проведат дидактически тестове.
 - Да се анализират резултатите от експериментите.

Основна хипотеза на изследване – повишаване успеваемостта на студентите в овладяване на базови дисциплини по програмиране и мотивацията им за тяхното изучаване

може да се осъществи чрез управление на процеса на обучение по тези дисциплини в условията на конструктивистка и високотехнологична среда.

Структура и обем на дисертационния труд

Дисертационният труд се състои от увод, 3 глави, заключение, използвани библиографски източници и приложения.

В първа глава – „**Теоретични основи на дисертационното изследване**” се разглеждат актуалните проблеми на обучението по програмиране. Анализирани са мненията на анкетирани студенти от различни специалности. След проучване на множество литературни източници са описани изводите от прилагането на активно обучение в различни направления и по-конкретно в областта на програмирането. Обоснована е авторската идея за управление на процеса на обучение по програмиране в условията на конструктивистка и високотехнологична среда.

Втора глава – „**Дидактически инструментариум за реализиране на методиката на обучение по програмиране чрез използване на ИКТ**” анализира предимствата на подхода за обучение чрез забавление (edutainment). Предложена е методика за провеждане на активно обучение по програмиране в стил edutainment чрез използване на ИКТ. Представени са авторски компютърни методи и средства за прилагането ѝ. Описани са предимствата при използването на авторски вместо на готови средства.

Трета глава – „**Организация, провеждане, резултати и анализ на резултатите от педагогически експеримент**” описва провеждането на педагогически експеримент със студенти от специалност Бизнес информационни технологии, първи курс, представя и анализира резултатите от експеримента.

В **Заключението** се анализира изпълнението на поставените задачи и постигането на основната цел. Формулирани са приносите на дисертационния труд и са посочени перспективите за развитието му. Приложен е списък с авторски публикации.

Използваната литература включва 140 заглавия, от които 48 на кирилица, 82 на английски език и 10 интернет източника. Приложенията включват анкети, примерни задачи и тестове.

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

ГЛАВА 1. ТЕОРЕТИЧНИ ОСНОВИ НА ДИСЕРТАЦИОННОТО ИЗСЛЕДВАНЕ

1.1 Процесът на обучение – определение, методи на обучение

Основна цел на всяко обучение е да подготви индивиди, които пълноценно се интегрират в обществената и работна среда, и могат да играят активна роля в нея. За изпълнение на тази цел европейският парламент разработи „Европейската квалификационна рамка за учене през целия живот” (ЕКР). Предложена през април 2008 година, ЕКР включва осем референтни нива, в които се описват умения и компетентности, препоръчителни за покриване във всички страни от ЕС. ЕКР поставя акцент върху резултатите от ученето, а не върху вложените средства, например неговата продължителност. Резултатите от обучението са конкретно описани в три категории – знания, умения и компетентности. Това показва, че квалификациите, в различни съчетания, обхващат широк спектър резултати от обучението, включително теоретични знания, практически опит и технически умения, както и социални компетентности, при водеща роля на уменията за работа в екип. В контекста на ЕКР „знания“ означава резултат от усвояване на информация в процеса на учене. Знанията са съвкупност от факти, принципи, теории и практики, които са свързани с определена сфера на работа или обучение. Знанията се описват като теоретични и/или фактологични. Под „умения“ се разбира способността за прилагане на знанията и използване на ноу-хау при изпълнение на задачи и решаване на проблеми. Уменията се описват като познавателни (включващи прилагане на логическо, интуитивно и творческо мислене) и практически (включващи сръчност и употреба на методи, материали, уреди и инструменти). Понятието „компетентност“ означава доказана способност за използване на знания, умения и личностни, социални и/или методологични дадености в работни или учебни ситуации и в

професионално и личностно развитие. В контекста на ЕКР способностите се описват с оглед степента на поемане на отговорност и самостоятелност [8].

Болонският процес е инициатива, чиято основна цел е стандартизиране на образованието в Европа – въвеждане на лесно разбираеми и сравними образователни степени, въвеждане на система на учебните кредити, въвеждане на европейски измерения за качество и други. Важен компонент в този процес е пренасянето на фокуса от преподавателя и пасивното обучение върху студента и активното обучение.

Процесът на обучение не е обикновен акт на подготовка и представяне на предварително подготвен материал. Преди започване на обучение, преподавателят трябва да е наясно с две неща:

1. възможности, постижения, силните и слабите страни, наличните познания и интересите на своите обучаеми;
2. краткосрочни и дългосрочни цели, които си е поставил за даден лекционен курс.

Така учебната стратегия за всеки лекционен курс ще действа на базата на това, което се знае за обучаемите и разработената тактика за преподаване, съответстваща на това познание.

Основните методи на обучение, които се прилагат понастоящем в сферата на висшето образование, са следните:

- лекция;
- семинарно упражнение;
- лабораторно упражнение;
- курсов проект;
- практика по специалността;
- преддипломен стаж.

Теоретично, така предложените методи на обучение, би трябвало да дадат възможност на студентите да получат нови знания по време на лекции, да ги затвърдят на семинарни и лабораторни упражнения и да ги приложат на практика в условия, близки до тези на реалната работна среда. В действителност, обаче, голяма част от студентите не могат да преминат безпрепятствено през описаната процедура, причините за което са комплексни [20,16]. В настоящия труд си поставихме за цел да проучим студентското мнение в първата година на обучението по програмиране, която е изключително важна за полагане на солидна основа, върху която нататък се изграждат професионалните умения.

1.2 Проучване на студентското мнение

Обратната връзка със студентите е от изключителна важност. Често резултатите от провеждането на анкети дават ясно и еднозначно посоката, в която една организация трябва да поеме [7]. За проверка на нагласата на студентите към активно обучение по програмиране, беше проведена анкета в няколко факултета на Пловдивския Университет.

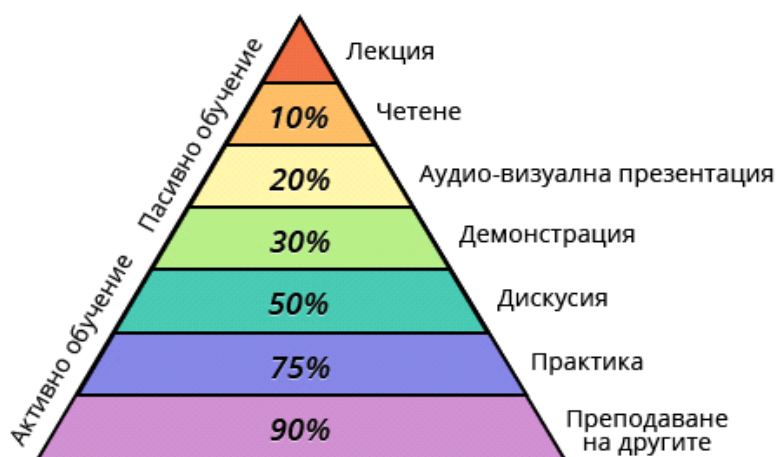
Взелите участие в проучването са студенти първокурсници от три специалности, в които се включва изучаване на информатика – Математика и информатика, Биоинформатика и Компютърна химия. Анкетата включва 20 въпроса, свързани с обучението на студентите както по време на лекции и упражнения, така и при работата им външ по време на самостоятелната им подготовка, изготвяне на домашни и курсови работи.

Проучването стана необходимо, тъй като в последните години посещаемостта на студентите на лекции и упражнения започна все повече да намалява. Поради трудността на научната материя, голяма част от студентите се обезкуражават още в самото начало, въпреки популярността на дисциплината и бързо започват да губят интерес към нея.

След обработка на резултатите от анкетата бяха направени някои съществени изводи, които дадоха насока на по-нататъшната ни работа. Анкетата доказва, че самите студенти високо оценяват методика, в която освен стандартните учебни помагала, чиято полза не се отрича, е подходящо да се използват допълнителни компютърни методи и средства за повишаване на интереса им към програмирането, атрактивността на поднесенния материал и неговата разбираемост.

1.3 Дидактически основи на активното обучение

Едгар Дейл (1900-1985) работи в университета на Охайо от 1929 до 1970 и е международно признат пионер в използването на аудио-визуални материали в обучението. Неговата най-известна концепция "конус на познанието" илюстрира графично връзката между начина на представяне на учебното съдържание и степента на усвояването му. На върха на този конус е слушането на лекция, следва четенето на учебни материали и т.н. Всяко по-долно ниво сваля степента на абстракция и увеличава процентът на възприетия материал. Конкретните проценти са добавени от последователи на Дейл след емпирични изследвания в обучението и резонно са подложени на критика. Естествено, че конкретните проценти зависят от много странични фактори и те не са обект на изследване в настоящия труд. Причината да цитираме конуса на Дейл е, че споделяме неговото мнение за значително увеличаване на усвояемостта на материала при преминаване от пасивна към активна форма на обучение.



Фигура 1 - Конус на познанието

От гледна точка на педагогиката и образователната психология – теорията на конструктивизма подчертава, че обучаемият конструира собствено познание, посредством активно изследване и откриване вместо пасивното му получаване [22, 32].

Според друг американски учен, специалист по психология на образованието – Джеръм Брунър, всички деца имат естествено любопитство и желание да покажат компетентност в решаването на различни задачи в процеса на обучение. Когато поставената им задача е твърде трудна, обаче, те се отегчават и губят интерес. Преподавателят трябва следователно да съобрази поставените задачи с нивото на обучаемите. В изследванията си върху когнитивното развитие на децата [50], Джеръм Брунър предлага три нива на предаване на знанията:

- Enactive – посредством пряко предаване на опит;
- Iconic – посредством образи (картинни, видео, мултимедия);
- Symbolic – посредством думи и символи.

Конструктивистката теория на Брунър твърди, че при поднасяне на нов материал, най-ефективно е това да става в посочената последователност от **enactive** до **symbolic**, дори и при обучение на възрастни. И тук се вижда последователността на преминаване от по-конкретни към абстрактни знания. Съвременни последователи на Брунър посочват значението на теориите за електронно обучение, които подпомагат активната ангажираност на обучаемите и докладват ползата от използването им за подобряване на резултатите от учебния процес [53].

В [37] е подчертано, че конструктивисткият подход е удачен при формиране на новата стратегия на обучение, използваща съвременните ИКТ. С всяка изминала година тяхното прилагане в процеса на обучение ще се увеличава и усъвършенства [3]. Според [33] съвременните ИКТ позволяват на преподавателя да:

- разработи нови методически подходи;
- промени организацията на обучение;
- разшири базата от знания и арсенала от техники за придобиване на умения по съответния учебен предмет;
- осъществи компютърна диагностика, контрол и оценка на знанията и уменията и компетенциите в обучението.

Разработването на виртуални уеббазирани среди за обучение и тяхното използване в образователния процес създава условия за разгръщане на потенциала на обучаемите, на тяхното интелектуално и личностно развитие. Освен удобство в придобиването на знания и умения, те позволяват проверка на нивото на усвояването им, но също така и самоконтрол и самооценка [45]. Конструктивистки подход в духа на теорията на Сеймур Пейпърт (Seimour Papert) и активно обучение се използват при преподаването на компютърна графика и моделиране [80].

В университета Карнеги Мелън, САЩ, за часовете по обектно-ориентирано програмиране е създаден образователен софтуер „Alice”, чрез който студентите създават сами виртуални реалности, игри и анимации, като между другото научават и структурата, функциите и алгоритмите за създаването на едно истинско приложение [52, 81]. Друг подобен продукт е „Greenfoot” – проект на университета в Кент, Англия. Представява интегрирана образователна среда, чиято цел е да обучи потребителите си на обектно-ориентирано програмиране на Java [68]. Greenfoot комбинира програмирането на Java с графичен и интерактивен резултат, който се визуализира на екрана. Интерфейсът му е по-олекотен от този на Alice, като успоредно със създаването на обекти на сцената, обучаемият може да следи и автоматичното генериране на код. Двата продукта са високо оценени от Оракъл. Началното обучение по програмиране в тяхната Оракъл Академия се провежда с използването на тези две системи.

Напоследък с развитието на ИКТ, се прилагат различни подходи за активно обучение като например т.нар. „учебно скеле” (instructional scaffolding) [98], екипно обучение (*group-based learning*) [67], технологично поддържано обучение (Technology Enhanced Learning) [39,49], проектно-базирано обучение [2, 44, 66, 72], активно обучение с елементи на студентско изследване (*inquiry based learning*) [92], колаборативно обучение [55, 56, 60, 82, 90], смесено обучение (*blended learning*) [21, 64, 96], „подгряващо обучение” (*Just-in-Time Teaching*) [54, 77, 102].

Проучвания доказват, че употребата на подходящи технологични средства и активно обучение съвместно с правилния педагогически подход могат да стимулират ученето, да насърчат партньорството и да повишат компетентностите на студентите в различни курсове [42, 58, 61, 63, 65, 69, 70, 78, 79, 83, 85, 87, 99, 100]. За усъвършенстване на този тип обучение се използват и т.нар. виртуални колаборативни пространства [40, 89, 93] и учебни обекти [75, 94, 95, 101].

Ефективни се оказват и учебни среди, в които концепциите на програмирането са представени под формата на игри [43, 86]. Визуализацията на алгоритми [12, 13, 38, 76, 88] и анимацията [91] са често използвани методи за активно обучение.

Peer Instruction (PI) е интерактивен метод на обучение, който спомага за повишаване ангажираността на студентите по време на лекции. Подходът включва промяна на традиционната класна стая чрез получаване на информацията извън клас, а асимилирането ѝ и практическото ѝ прилагане в клас. Снабдени с необходимата за лекцията информация, студентите отговарят на въпроси с множествен избор, като използват необходимата техника, после обсъждат въпросите със своите колеги и отговарят отново. Методът е разработен от харвардския професор Ерик Мазур в началото на 1990 г. [62, 73].

1.4 Резултати и изводи

Направеният обзор на използваните подходи за активно обучение и анализът на прилагането им позволяват да се направят следните изводи:

1. Голяма част от студентите в различни университети по целия свят срещат съществени трудности при усвояването на учебния материал в уводните курсове по програмиране. Немалка част от тях даже отпадат от следването.
2. Активното обучение, приложено в курсовете по програмиране, компютърни науки, информационни технологии, математика и др. науки води до повишаване на успеваемостта на студентите.
3. Подготовката на инструменти, средства и материали за активно обучение изисква значителни допълнителни усилия от страна на преподавателя, което възпрепятства масовото използване на този метод.
4. Наложително е факултетните и университетски ръководства да разработят система за стимулиране на преподавателите, прилагащи иновативни форми на обучение, водещи до повишаване на качеството на обучение, с което да се мотивират по-голям брой преподаватели.

Основните характерни черти на активното обучение са в синхрон с ЕКР:

1. Обучаемите не са само пасивни слушатели.
2. Ангажирани са с допълнителни дейности – събиране на информация по дадена тема, въвличане в дискусии, представяне на собствени материали.
3. Формират се компетентности и умения, вместо да се разчита само на предаване на информация.
4. Отчитат се нагласите, манталитета и ценностната система на обучаемите.
5. Повишава се мотивацията на обучаемия.
6. Обучаемите могат да обменят идеи с преподавателите и в извънучебно време.
7. Обучаемите прилагат анализ, синтез и оценяване.

Постигнати са следните резултати в първа глава:

1. Изследвани са основните трудности, които студентите срещат при усвояване на базовите дисциплини по програмиране.
2. Проучена е самоподготовката на студентите и предоставеният им инструментариум за улесняване на нейното провеждане.
3. Анализирани са нагласата на студентите за активно участие в образователния процес и актуалното състояние на активното обучение по програмиране.
4. Проучени са литературни източници, разглеждащи прилагането на модела на активно обучение в различни направления и по-конкретно в областта на програмирането.

ГЛАВА 2. ДИДАКТИЧЕСКИ ИНСТРУМЕНТАРИУМ ЗА РЕАЛИЗИРАНЕ НА МЕТОДИКАТА НА ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ ЧРЕЗ ИЗПОЛЗВАНЕ НА ИКТ

2.1 Обучение чрез забавление (edutainment) в програмирането

Английският неологизъм **edutainment** [74] се появява за пръв път през 1948 г. като термин, с който компанията Уолт Дисни характеризира документалната си образователна филмова поредица „True Life Adventures”. Терминът се затвърждава от Робърт Хейман през 1973 г. с производството на документални филми за Националното географско дружество. Използвано е също и от д-р Крис Даниелс през 1975 г. в своя проект на хилядолетието (www.elysianworld.com). Възникнало като средство за обучение на деца, методът за обучение чрез забавление се приема позитивно от обучаеми от всякаква възраст. Успешното прилагане на **edutainment** в процеса на обучение се дължи на следните неоспорими факти:

1. Абстракцията се свежда до минимум. Това важи особено при усвояването на голямо количество абстрактни типове данни в началните курсове по програмиране.
2. Теоретичната част се онагледява с множество практически примери, включвайки симулации на процеси, използващи анимации.
3. Психологическият момент на приятно изучаване на учебния материал в игрова обстановка подпомага увеличаването на процента на усвояването му.

Бурното развитие на компютърната техника и технологиите в последните години, позволи направлението да се развие и разшири. То стана обект на множество научни изследвания и средство за постигане на интензификация в обучението. От 2006 година ежегодно се провежда международна конференция EDUTAINMENT по електронно обучение и дигитални развлечения. Основните научни методи и концепции, обсъждани в семинарите към конференциите, са следните:

- Игрово тестване и оценяване;
- Учене с роботи;
- Компютърни игри за изграждане на знания;
- Игрово изучаване на езици;
- Рентабилно производство (авторски средства, подпомагане на непрограмисти; платформи и средства за е-обучение, анимации, гейм сървъри, мидълуер - междинен интеграционен софтуер);
- Проектиране и разработка на игри (физика на игрите, рендъринг и анимация в игрите, виртуални герои);
- Игрови технологии (браузърни игри, онлайн игри с много играчи, мобилни игри, конзолни игри);
- Персонализация и адаптация (контекстни модели, презентирание чрез игра, планиране и изкуствен интелект);
- Съвместно обучение и среди за обучение (сериозни игри, арбитражиране на игри, игри и социални мрежи, сътрудничество, състезателност);
- Интерактивни разказвачи;
- Интерфейси и сензорни технологии (конзоли за игри, жизнени сензори и параметри, добавена реалност, смесена реалност в сценариите за мобилно обучение, невизуални усещания: миризма, докосване, слух) ;
- Оценяване (методологии, модели и показатели за оценяване, инструменти за оценяване, ефективност, ефикасност).

Обучението чрез забавление е атрактивен метод за мотивиране на обучаемите да възприемат по-лесно и приятно поднесен материал [31, 47, 48]. Този метод, обаче, е добре да бъде съчетан със стимулирането им към интерактивна намеса в процеса, а не да бъдат само негови пасивни свидетели [30, 71]. Желателно е също обучаващите среди, използващи този метод, да могат лесно да се интегрират в стандартната преподавателска практика според нуждите на конкретния преподавател [51]. От съществено значение е и възможността за адаптация на тези среди към нивото на обучаемите [29, 41, 48]. Когато им се задават интересни проблеми, активността им е значителна в сравнение с поставянето на рутинни задачи [1]. Участието в образователни игри е средство и за откриване на математически таланти [11].

Подходът за обучение чрез забавление в програмирането се базира на лесно достъпни и разбираеми интерактивни анимирани примери, които представят основни алгоритми, визуализират действието и начина им на работа. Така програмирането става по-достъпно, изясняват се бързо и лесно много от елементите на даден алгоритъм, представляващи сериозна трудност както за обяснение, така и за разбиране. Друго предимство при използването на анимация в обучението или самоподготовката е привличането и задържането на интереса на ученици и студенти към изучавания материал. Благодарение на този подход, основните концепции в програмирането се усвояват по-лесно и то не се струва толкова трудно на обучаемите.

2.1.1 Подобряване на обучението посредством “Edutainment”.

В последните години все повече студенти изпитват сериозни затруднения в усвояване на програмирането. След анализ на срещаните трудности [30] се стигна до заключението, че трябва да се използват нови подходи за обучение. Един от тях е визуализация с елементи на

компютърна анимация. Той навлиза все по-широко в съвременния свят, където online учебниците започнаха да изместват своите хартиени еквиваленти. Визуализацията е удачен подход и при обучението както по математика [59], така и по информатика [14]. В последните години, анимациите от рода на “как да направя дадено нещо” започнаха да се появяват все повече в световната мрежа, като тяхната популярност нараства всеки ден. Доказателство за това са милионите гледания на даден видео клип, за разлика от броя на свалянията на определен online учебник например, които не наброяват повече от няколко десетки хиляди. Основното предимство, предоставено от анимацията е това, че придобиването на дадено умение вече не представлява голяма трудност дори за аматьорите в дадената област. За разлика от голяма част от учебниците, изобилстващи само от теория, гледайки видео клипа, обучаемият може да добие представа и полезен опит. Друго важно предимство в случая е преодоляването на езиковите различия, нещо неизбежно при употребата на online учебниците. С минимални знания по даден език обучаемият може да получи необходимата му информация за функционирането или изработването на предмета, представляващ интерес за него.

2.1.2 Авторски средства за „Edutainment”

След обширно проучване в интернет бяха открити голям брой примери за обучение чрез анимация. За съжаление, малка част от тях е отделена на обучението по информатика. Освен това, тези източници са непълни и до голяма степен неясни, тъй като липсват началните обяснения за инсталиране на съответната програмна среда/платформа, липсват и обяснения на основни понятия, необходими за по-доброто разбиране на извършваните действия. Още повече, почти не се срещат подобни информационни източници на български език, което допълнително влошава разбираемостта и приложимостта на показваното. Друг недостатък на анимираните примери, разпространявани свободно в мрежата е, че те са правени от непрофесионалисти. Въпреки добрите си намерения, тези хора не винаги притежават достатъчна компетентност, за да положат здрави и сигурни основи при обучението на начинаещи програмисти. Случва се дори тези „любителски” анимации да са лишени от всякакъв смисъл и коректност, което още повече би объркало представите на обучаемите. Съществуват, разбира се и продукти в мрежата, както и online курсове, разработени от квалифицирани преподаватели по съответния предмет. За съжаление тези анимации често струват немалко пари, като в повечето случаи само една малка част от продукта се предлага безплатно, за да привлече потенциалните клиенти. Не може да стане ясна и ефикасността на подобни продукти, тъй като чак след плащането, обучаемият разбира дали това е точния източник, от който се нуждае, дали информацията в него е съвременна и полезна, дали версията на продукта е актуална. Често автентичността на такива източници е съмнителна.

След казаното дотук, лесно може да се направи извод, че съществуващите анимираните самоучители в глобалната мрежа не са чак толкова полезни, колкото голяма част от обучаемите очакват, без да се споменават и различните опасности, които съществуват в интернет, като прихващането на вируси, извличане на лични данни от случайни регистрационни форми и др.

С цел, избягване на посочените недостатъци, пристъпихме към създаване на авторски средства за обучение чрез забавление [28]. Те включват интерактивен модул за обучение при използването на подпрограми, система за визуализиране на алгоритми и интегриран графичен редактор. Основната задача, която си постави авторският екип, бе системата за визуализиране на алгоритми да представи тези алгоритми по възможно най-леснодостъпен, разбираем и атрактивен начин в помощ на процеса на изучаване на информатиката. Графичният редактор, освен стандартните функции, като създаване и редактиране на изображения, ротация, симетрия и др., включва и възможности за създаване и манипулиране на графики на функции. С негова помощ лесно и приятно обучаемите получават и затвърждават знания по математика, информационни технологии, компютърна графика и обектно-ориентирано програмиране.

2.2 Модел за преподаване на подпрограми

Проучването на студентските мнения, представено в предишната глава показва, че при началното обучение по програмиране най-трудно се усвояват основните понятия, подпрограмите и масивите. Анализирайки студентските грешки, които те допускат, когато създават или извикват подпрограми, направихме следните изводи:

- механизъм на предаване на параметри остава черна кутия за студентите за дълъг период от време;
- много глобални променливи се използват в подпрограмите;
- локални променливи често се появяват в списъка на формалните параметри.

Въпреки че тези проблеми се обсъждат многократно със студентите по време на лекции и упражнения, необходимостта от допълнителни самостоятелна подготовка за дълъг период от време е очевидна. Ние предлагаме два подхода в настоящия труд за тази цел. Статичният подход е подходящ за писмен текст в хартиен учебник. Динамичният подход е подходящ за интерактивно обучение чрез използване на компютър.

2.2.1 Статичен подход за самообучение

Интегрирането на различни понятия (област на валидност на променливите, механизъм на предаване на параметри) в един пример подпомага процеса на преподаване. Множество от примери е предложено на студентите с цел да усвоят тези понятия и да ги използват. Те са разработени както за езика за програмиране Паскал, така и за C++. Примерите имат за цел да демонстрират разликата между параметри-променливи и параметри-стойности. Също така представят възможност на студентите да осъзнаят областта на валидност на локалните и глобални променливи и програмното разрешаване на колизията, при която и двете имат едно и също име.

2.2.2 Интерактивен модул за самообучение

Динамичният подход е съчетан с подходяща визуализация на процеса на изпълнение на подпрограмата. За тази цел бе създаден интерактивен модул. Той е нова част от компютърна система за подпомагане на цялостния процес на обучение по информатика. Подпрограмата в демонстрационния модул има два формални параметъра. За разлика от статичния подход, динамичният позволява на студентите да изберат интерактивно вида на всеки параметър – параметър-стойност или параметър-променлива.

2.3 Система за динамично обучение по програмиране

Съществен елемент на методиката на обучението е комбинирането на опорни задачи по програмиране [25, 26] със система за динамично обучение, чието създаване е по силите на студентите, преминали уводния курс на обучение по програмиране.

2.3.1 Обща идея

При разработването на програмната система за обучение по програмиране си поставихме като основна цел да демонстрираме на студентите, които ще ползват програмния текст като образец, реализиране на стегнат код, минимален размер на изпълнимия файл и максимална разбираемост.

Концептуално системата е разделена на четири основни модула според типа на визуализираните алгоритми:

- Базови алгоритми;
- Вектори;
- Матрици;
- Динамични структури от данни.



2.3.2 Базови алгоритми

2.3.2.1 Цикли

В програмата е визуализирано действието на трите вида цикли – for, while и do while. Обучаемият може да наблюдава изпълнението на всеки един от тях поотделно, избирайки желания алгоритъм от падащия списък, задавайки сам началните стойности и следейки техните промени в хода на програмата. Обучаемият може и сам да настройва скоростта на изпълнение на алгоритъма за по-прецизно наблюдение на действието му, тъй като съществуват няколко възможни режима. Настройките за бързина и изпълнение по стъпки са разработени и в останалите модули на програмния продукт. След задълбоченото изследване и на трите алгоритъма за цикъл, обучаемият може сам да направи изводи за начина на тяхната работа, прилики и разлики между тях, както и бързина, ефективност и приложение.

2.3.2.2 Сума и средноаритметично на числа

Един и същи интерфейс се ползва за демонстрация на сума и средноаритметично. Изборът на действия се осъществява от комбинирана кутия. Съществени елементи при визуализацията са масивът и суматорът, който „събира” в себе си стойностите на елементите, движейки се под тях. При всяко добавяне на елемент към него, този елемент се оцветява в червено за подчертаване на действието добавяне към суматор. В този модул обучаемият отново може сам да въведе брой и стойности на елементите от масива, с който оперира програмата.

2.3.2.3 Намиране на най-голям общ делител (НОД) и най-малко общо кратно (НОК) на две или повече числа

Реализацията на програмния модул интерфейсно наподобява на модула за намиране на сума и средноаритметично. Отново използваме един и същ интерфейс за демонстрация на НОК и НОД, като изборът на действия се осъществява от комбинирана кутия. Разглеждат се първите две въведени числа. За улеснение на обучаемия, те се оцветяват в червено. Намира се техният НОД (или НОК съответно). Неговата моментна стойност се „складира” в суматора. След това се преминава към разглеждане на второто и третото число, извършвайки същата дейност и се продължава така до достигане края на масива.

2.3.2.4 Решето на Ератостен

Задача: По метода на решето на Ератостен [34, 35], отделете всички прости числа, принадлежащи на интервала $[1, n]$, в даден едномерен масив.

Визуализация на алгоритъма: При стартиране на програмния модул се появяват всички числа от 2 до n (числото 1 не участва, тъй като е просто и същевременно делител на всички останали). Подчертаваме 2 и зачеркваме през едно всички числа след 2. Това са именно числата, които се делят на 2 и следователно не са прости. Първото незачеркнато число е 3 и то е просто. Подчертаваме 3 и зачеркваме през две всички числа след 3. Продължаваме по същия начин до края, търсейки винаги първото незачеркнато число и зачертавайки кратните на него (ако има такива). Зачертаването става, като кратните числа се оцветяват в червено, за да бъдат наблюдавани от обучаемия, а после изчезват, като числото, на което те са кратни, се визуализира в долната част на интерфейсия прозорец. След приключване работа на програмата, във формата се визуализират всички прости числа, принадлежащи на дадения интервал.

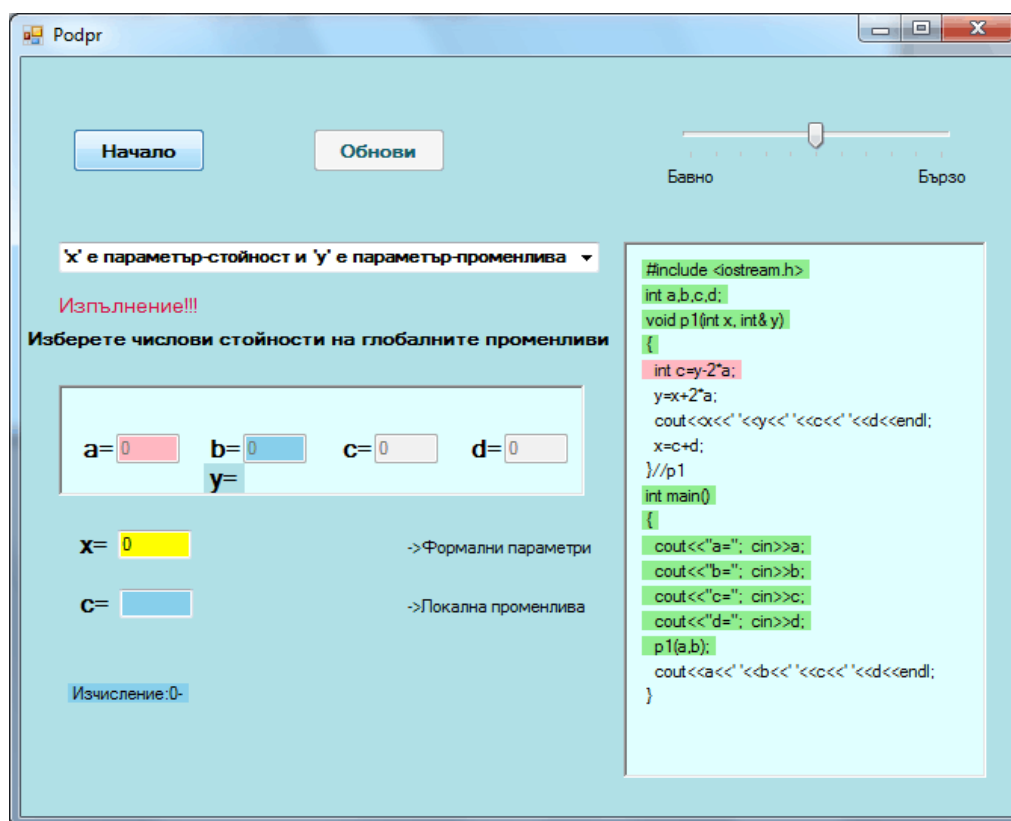
2.3.2.5 Числа на Фибоначи

Представява редица (масив) от числа, като всеки следващ елемент на тази редица се получава от сбора на предходните два [36]. Елементите, които се събират се оцветяват в жълт цвят за по-нагледно, след което се извършва тяхното събиране и добавяне на новополучения елемент към редицата.

2.3.2.6 Подпрограми

В настоящия програмен модул подробно и нагледно е обяснена работата на програмен фрагмент, включващ използването на подпрограма с два параметъра x и y . От комбинираната кутия може да бъде избран видът на тези параметри – променлива или стойност. След стартиране изпълнението на програмата, по програмния фрагмент в дясната част на

прозореца започва да се движи маркер. Изпълняваният в момента ред (или част) се оцветява в червен цвят, след като бъде преминато през него, променя цвета си в зелено. Така обучаемият лесно може да проследи хода на програмата. При достигане на частта, в която се използва подпрограма, в лявата част на прозореца се появява набор от необходимите за програмата глобални променливи. Обучаемият може сам да им зададе стойности, ако това не се случи до определен момент, тези променливи автоматично получават нулеви стойности. Същото се случва и с формалните параметри и локалната променлива. В най-долната част на прозореца се показва изпълнението на програмата, изчисленията, които се получават и междинните резултати.



Фигура 2 - Подпрограми

2.3.2.7 Ханойски кули

Задачата за Ханойските кули винаги е представлявала голям интерес за повечето студенти, но тя трудно може да бъде обяснена без достатъчно нагледен пример за решаването ѝ. В настоящия програмен модул чрез анимация е показано движението на различните по големина дискове, като премествания в даден момент диск се оцветява в червено. Допълнително, в горната част на прозореца се изписва като инструкция кой диск от къде трябва да бъде взет и на кой пилон трябва да бъде преместен до окончателното преместване на цялата кула.

2.3.3 Вектори

2.3.3.1 Сортировки

- Сортировка по метода на мехурчето

Задача: Да се напише програма, която да подрежда елементите на даден едномерен масив във възходящ ред по метода на мехурчето.

Визуализация на алгоритъма: При визуализирането на този алгоритъм се използват допълнително правоъгълници за моделиране на елементите на масива. Идеята е обучаемите да получат нагледна представа за сортирането като „подреждане на аптекарски шишета”. Понеже при сортирането се използват вложени цикли, то в тази демонстрация се използва и постъпково изпълнение – разделени са външният и вътрешен цикъл. Всяка стъпка е едно

изпълнение на вътрешния цикъл. Двата съседни елемента, които се сравняват и евентуално разменят са оцветени в различни цветове. Първоначалният неподреден масив е оцветен в черно. Сортираната част отляво започва да се оцветява в синьо. На следващата фигура е показана визуализация на алгоритъма в процес на действие, при което два съседни елемента (71 и 30) се сравняват и предстои да разменят стойностите си. Отляво – подредената част (до елемента със стойност 42) е оцветена в синьо.

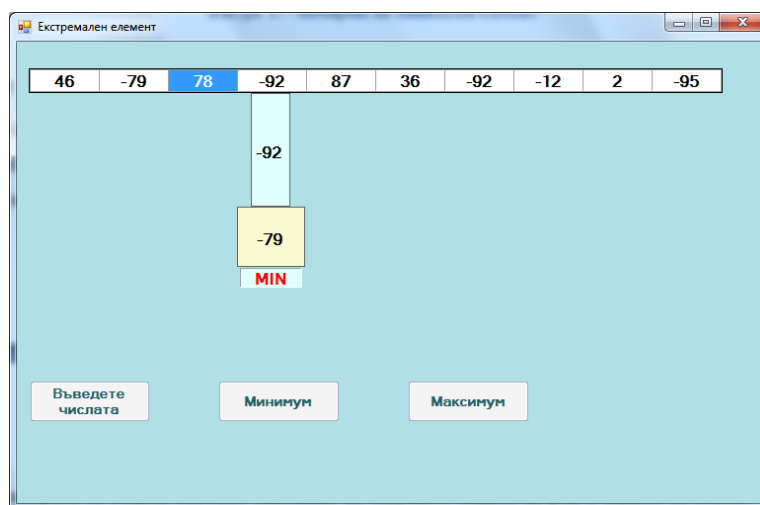


Фигура 3 - Сортировка по метода на мехурчето

- Сортировка по метода на пряката селекция

2.3.3.2 Намиране на екстремален елемент в масив (минимум или максимум)

Маркираща лента с прикрепена в долната си част кутия се движи по елементите на масива. При стартиране на алгоритъма, в кутията се „изсипва” стойността на първия елемент от масива. При намиране на стойност, която е по-голяма от съдържанието на кутията Max, се симулира промяна на тази стойност посредством изсипване на новата стойност в кутията. Отделно съдържанието на максималния елемент се показва в по-голяма кутия. Визуализацията на алгоритъма за намиране на минимален елемент е същата, с тази разлика, че кутията приема стойността на най-малкия елемент от редицата. На Фигура 4 е визуализиран алгоритъмът за търсене на минимален елемент. Уловен е моментът, когато елемент от масива, чиято стойност е -92, се спуска в панела, за да замени досегашния минимален елемент -79.



Фигура 4 - Намиране на минимален елемент

2.3.4 Основни алгоритми с матрици

Основните алгоритми с матрици обхождат отделни елементи на матрицата като стълб, ред, главен, второстепенен диагонал, обособените от тях триъгълници, а и визуализират тези елементи, оцветявайки ги в различни цветове. При обхождането на структурите се отделя особено внимание на текущите индекс на ред и индекс на стълб. Основни функции на визуалните приложения са:

- а) манипулиране на елементите на матрицата, което включва добавяне, изтриване, редактиране или случайно запълване;
- б) избор на скорост на изпълнение на алгоритъма с възможност за спиране
- в) подчертаване на отделни структури в алгоритъма посредством специфичното им оцветяване.

2.3.4.1 Обхождане елементите на матрица

Задача: Даден е двумерен масив $a(5,5)$. Да се напише програма, която:

- а) да обхожда матрицата по редове;
- б) да обхожда матрицата по стълбове;
- в) да обхожда матрицата тип „змия”
- г) да обхожда матрицата тип „охлюв”.

Визуализация на алгоритъма: При обхождане по редове и по стълбове, разглежданият в момента стълб/ред се оцветява в бледозелено, преминава се през всичките му елементи, като те се оцветяват последователно в червено. По подобен начин се обхожда и матрицата и тип „змия” и „охлюв”. След като червения маркер премине през даден елемент, този елемент се оцветява в тъмозелено. В дясната част на прозореца са разположени три панела, в които обучаемият може постоянно да следи индекса на реда и на стълба, както и стойността на елемента, през който преминава маркера.

2.3.4.2 Суми на елементите на матрица под и над главен или второстепенен диагонал

Задача: Даден е двумерен масив $a(5,5)$. Да се напише програма, която:

- а) намира сумата на елементите на матрицата над главния диагонал;
- б) намира сумата на елементите на матрицата под главния диагонал;
- в) намира сумата на елементите на матрицата над второстепенния диагонал;
- г) да намира сумата на елементите на матрицата под второстепенния диагонал.

Визуализация на алгоритъма: Изборът на един от четирите описани в условието на задачата варианти се осъществява от комбинирана кутия. При визуализирането на този алгоритъм се използват червен цвят за оцветяване на диагонала, зелен – за оцветяване на обхождания триъгълник, тъмно зелен за маркиране на вече добавен към сумата елемент. В дясната част на прозореца има панел за сумата, към който последователно се добавят съответните елементи. В двата панела отдолу отново могат да се наблюдават индексите на реда и стълба на добавения елемент.

2.3.4.3 Намиране на минимален и максимален елемент в матрици

Задача: Даден е двумерен масив $a(5,5)$. Да се напише програма, която:

- а) да намира максималния от минималните елементи по редове;
- б) да намира минималния от максималните елементи по редове;
- в) да намира максималния от минималните елементи по стълбове;
- г) да намира минималния от максималните елементи по стълбове.

Визуализация на алгоритъма: И тук изборът на един от четирите описани в условието на задачата варианти се осъществява от комбинирана кутия. При визуализирането на този алгоритъм се използват: зелен цвят за оцветяване на реда (стълба), в който се търси минимален (максимален) елемент, самият екстремален елемент, както и проверяваният за такъв се оцветяват в червено. След намирането на поредния екстремален елемент, той се прехвърля в едномерен масив. После от така получения масив отново се определя екстремален елемент. На представената фигура е показана визуализация на алгоритъма в последната му фаза, когато минималните елементи от всеки ред са прехвърлени във вектор и

е намерен максималният елемент в този вектор. Въпреки че задачата може да се прилага и за правоъгълни матрици, сме се спрели на варианта квадратна матрица за да може да се използва една и съща помощна структура – едномерният масив от 5 елемента. Визуализацията се използва и в обучението по обектно-ориентирано програмиране и изграждане на програми с ГПИ. Там студентите разширяват алгоритъма така, че да се прилага и за правоъгълни матрици. Предлага им се и разработването на вариант без използване на допълнителен масив.



Фигура 5 - Намиране на максимална стойност от минимумите на редовете

2.3.4.4 Намиране норми на матрици

Задача: Даден е двумерен масив $a(5,5)$. Да се напише програма, която намира норма на матрица по формулите: $N_1 = \max_{i=1,n} \sum_{j=1,m} |a_{ij}|$, $N_2 = \max_{j=1,m} \sum_{i=1,n} |a_{ij}|$.

Визуализация на алгоритъма: От комбинираната кутия избираме опция, във вектора в дясната част на прозореца се натрупват сумите от модулите на елементите по ред/стълб и се намира максималния от тях, като се оцветява в червено.

2.3.5 Алгоритми с динамични структури от данни

2.3.5.1 Линеен списък

Задача: Даден е линеен списък. Да се извършат върху него операциите добавяне на елемент, вмъкване на елемент, изтриване на междинен елемент, изключване на първия елемент, изтриване на елемент в края на линейния списък.

Визуализация на алгоритъма: Преди обучаемият да започне да оперира с линейния списък, първо трябва да го създаде. Това става лесно чрез натискането на бутона „Създаване на списък“. Ако списъкът не бъде създаден, излиза съобщение. След това обучаемият може да продължи с операциите върху списъка, като за вмъкването и изтриването на елемент трябва да посочи след кой елемент точно желае да се извърши действието. За още по-голямо удобство, в кутия, разположена в долната част на прозореца, се генерира код, съответстващ на извършваната операция.

2.3.5.2 Стек и опашка

Двете структури се обработват паралелно, за да се видят приликите и разликите. Операциите са добавяне на елемент и обработка на елемент (елементът се извлича, след което се премахва).

2.3.5.3 Двоично дърво

Първоначално се инициализира списък с 10 числа. Операциите са създаване на двоично дърво за търсене, търсене на елемент в дървото и изтриване на елемент от дървото. Всяка операция може да се извършва и постъпково. При търсенето и изтриването съответният елемент трябва да бъде въведен от обучаемия.

2.3.6 Тестов модул

Този модул се състои от две части. Едната е за създаване, редактиране и запазване на тестове в кодиран вид. Модулът е част от авторска разработка. Администрирането е опростено и не изисква специални права за достъп и настройка. Удобство за потребителя е възможността да импортира и готови текстови файлове, които могат да се създават и коригират с елементарен текстов редактор (напр. Notepad). Втората част е за провеждане на тестове. Тестовият модул работи с множествен избор от готови отговори.

2.4 Модел за преподаване на обектно-ориентирано програмиране

В настоящия труд е предложен модел за въвеждане в обектно-ориентираното програмиране. И в този случай заложихме на създаване на авторски софтуер, който да бъде атрактивен за ползване и същевременно да служи като образец за програмиране. В модела се предлагат две нива на усвояване на ООП – въвеждащо и приложно. В първо ниво се усвояват основните понятия в ООП и се правят “конзолни” програми. Във второто ниво се използват възможностите на визуалното програмиране, комбинирано с елементи на компютърната графика. Реализират се обектноориентирани програми за изобразяване на равнинни криви и компютърна анимация.

2.4.1 Обектноориентирано програмиране

Възникнало в края на шестдесетте години на двадесети век, утвърдило се през осемдесетте години и особено бурно развило се през последното му десетилетие, ООП се наложи като водещ стил на програмиране в началото на новия век [6].

2.4.2 Задачи.

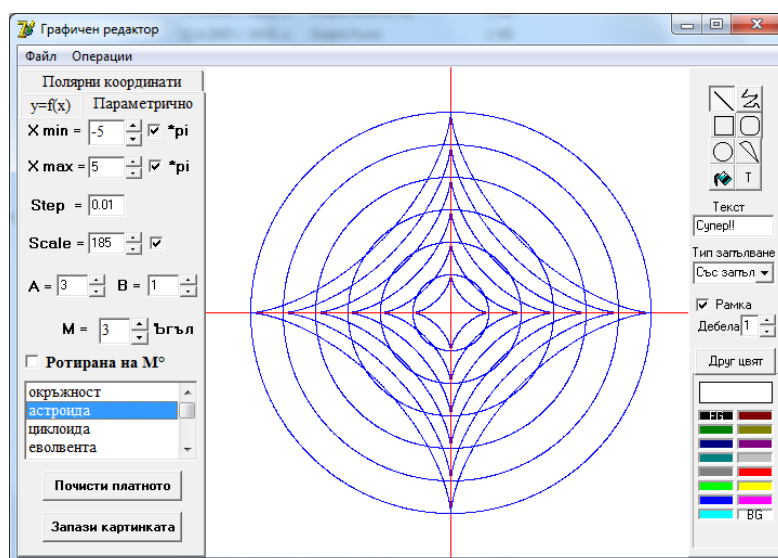
Примерните задачите се подбират така, че да могат да се решават на части след усвояването на всяко основно понятие. Така например след темата “Деклариране на класове от обекти” могат да се декларират първите класове (които нямат предшественици) и да се дефинират техните методи. След “Наследяване” – да се декларират техните наследници и т.н. След усвояване на последното основно понятие “Полиморфизъм”, тези задачи могат да се решават изцяло. Представени са две примерни задачи, които могат да се използват като образец за други подобни

2.4.3 Компютърна графика

Компютърната графика се занимава с изграждането (синтеза), представянето, обработката и изобразяването на графична информация. Тя е основна дисциплина в учебните планове на специалности по направление 4.6 „Информатика и компютърни науки”. Средства на компютърната графика се използват активно и в други учебни дисциплини като обектно-ориентирано програмиране [27], информационно моделиране [24, 97] и др. В процеса на обучение се подчертава спецификата на изобразяването на такава информация на монитор, състоящ се от множество точки (пиксели), които могат да имат различен цвят. Анимацията позволява по-добро усвояване на знанията. Декларира се абстрактен клас, който е базов за всички анимирани обекти. След това се дефинират конкретни наследници, които се движат посредством един полиморфен обект.

За чертаене на графики на равнинни криви се дефинира клас *Grafika*, съдържащ три полиморфни метода. Първият чертае крива, представена като функция на една променлива, вторият такава, представена параметрично и третият – в полярни координати. Формалните параметри на всеки метод са *ClientWidth* (Ширина на платното), *ClientHeight* (Височина на платното), *[Xmin, Xmax]* (Интервал, в който се чертае кривата), *Step* (Стъпка), *Scale* (Машабиращ коефициент). Всички тези параметри могат да се манипулират от левия панел на графичния редактор (Фигура 6). Ако е маркирана кутията за избор след краищата на

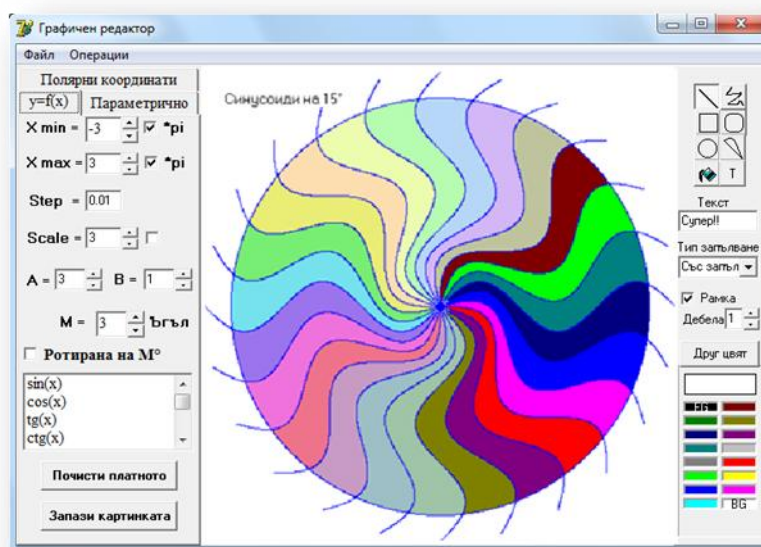
интервала $X_{\min}$, $X_{\max}$, то тези краища се умножават по $\pi=3,1415$. Оттам може да се избере и какъв вид функция да се чертае ($y=f(x)$, параметрично или полярни координати). След избора на тип функция, в долната част на панела се избира и самата функция, чиято графика да се чертае. Функциите, чиито графики се чертаят, са абстрактни методи и се предефинират в конкретните наследници. За нуждите на обектно-ориентираното програмиране, с демонстрирането на тази програмна част преподавателят мотивира необходимостта от използване на абстрактни методи, извикването им в полиморфните методи, без да са дефинирани и задължителното им предефиниране като виртуални методи в наследниците. Също така в частта удачно се илюстрира използването на един полиморфен обект за чертане на различни видове графики. За затвърждаване на основите на компютърната информатика, обучаемите успяват да видят разликата между понятия като дискретно (цифрово) и непрекъснато (аналогово) представяне на информацията като експериментират с различни стъпки (променяйки параметъра Step). При чертане на функция на една променлива, могат допълнително да я представят ротирана на определен ъгъл, който се подава като допълнителен незадължителен параметър на метода. Промяна на началната и крайна стойност на интервала позволяват изясняването на понятие като периодичност на функция и да се проследи проявата му при тригонометричните функции. Промяна на мащабиращия коефициент позволява да се чертаят концентрични окръжности, астроиди, елипси и други функции, представени параметрично. На фигура 6 са начертани концентрични окръжности и астроиди, представени параметрично съответно с формулите $x=\cos t$; $y=\sin t$ (окръжност) и $x=\cos^3 t$; $y=\sin^3 t$ (астроида). Маркирано е използването на мащабиращ коефициент (Scale) със последователно избрани стойности 35,65,95,125,155,185.



Фигура 6- Концентрични окръжности и астроиди

При интерактивната компютърна графика (ИКГ) потребителят има възможност непосредствено да създава графични изображения посредством различни периферни устройства (клавиатура, мишка, електронна писалка и др.). Има много системи с редактори за създаване на графични изображения, средства за преобразуването им, както и тяхното записване в различни графични формати. Авторският опростен графичен редактор позволява на обучаемия да го допълва самостоятелно. С цел повишаване на интереса му към изследване на графики на функции, графичният редактор е интегриран в програмата, която ги чертае. Към нея отдясно е добавен панел за чертане и оцветяване. Графичният редактор е средство, което позволява студентите да получават интердисциплинарни знания по математика, информатика, информационни технологии и дизайн. Същевременно работата с него е забавление и удоволствие напълно в стила на edutainment направление. На Фигура 7 е представена възможността за използването на редактора в стил edutainment с дизайнерска

цел в изчертаването на окръжност, синусоиди на 15° без координатни оси, и запълването с различни цветове.



Фигура 7 - Комбинация от окръжност и синусоиди на 15°

Редакторът демонстрира и други възможности за операции над графични обекти. Това са ротация на цялата картинка на произволен ъгъл и симетрия относно абсцисата, ординатата и техните ъглополовящи прави. Вижда се, че при ротация на ъгли, които не са кратни на 90 градуса, се губи от картината поради нейното дискретно представяне и се проявява ефектът на растеризация. Редакторът има две реализации – Delphi и C#.

2.5 Методика за активно обучение с използване на ИКТ

Създаденият инструментариум от компютърни средства и опорни задачи залегнаха в основата на методика за обучение, която бе приложена в редица уводни курсове по програмиране. За съответната учебна програма бяха поставени образователни цели, които обучаемите трябва да постигнат в съответствие с таксономията на Блум [46]. Ще дадем конкретен пример за прилагането ѝ в дисциплината „Програмиране”. Темите за десетте занятия са следните:

1. Стандартни типове данни, операции и функции. Основни конструкции за управление. (Преговор с входно контролно ниво)
2. Работа с потребителски функции. Дефиниране и деклариране на функция и нейните формални параметри. Използване на функция.
3. Опорни задачи за обработка на едномерни масиви – реализация с потребителски функции.
4. Сортировка и търсене.
5. Контролно упражнение и дискусия.
6. Знаков низ. Стандартни функции за обработка на низове.
7. Логически записи (структури).
8. Масиви от структури.
9. Масиви от структури. Оценка на ефективността на методите за създаване на програми без и със използване на потребителски функции.
10. Контролно упражнение и дискусия.

Вземайки под внимание невъзможността да се обхванат в детайли всички аспекти и специфики на обучението по програмиране, прилагайки тази таксономия, смятаме че тя е достатъчна за отчитане на степента на усвояване на учебния материал на базово ниво. В следващата таблица са описани образователните цели, които трябва да постигнат студентите в обучението си по програмиране. Всяка тема от учебната програма предвижда постигане на

няколко цели от таксономията. Срещу всяка от целите, обаче, е поставена тази тема от програмата, която в най-голям процент позволява постигането ѝ

Образователни цели по Блум	Учебна програма
1. Знание: включва запомняне и възпроизвеждане на фактическа информация –термини, дефиниции и други формулировки.	1. Типове данни в езика C++. Създаване на нови типове данни чрез изброяване (enum) 2. Деклариране на променливи в C++. 3. Операции, изрази и стандартни функции.
2. Разбиране: включва способността да се интерпретира и трансформира информацията с цел нейното структуриране и систематизиране.	4. Основни конструкции за управление. 5. Структури от данни в езика C++. Масиви и логически записи (struct). Деклариране на масиви и структури. 6. Низове
3. Прилагане: включва способността за използване на усвоените знания за решаването на непознати задачи.	7. Опорни задачи за обработка на едномерни масиви. 8. Сортировка и търсене 9. Текстообработка
4. Анализ: включва декомпозиране на общата материя на по-малки части и разглеждането им поотделно като самостоятелни единици. Това ниво е стъпката към решаването на по-сложни задачи.	10. Работа с потребителски функции. Дефиниране и деклариране на функция и нейните формални параметри. Използване на функция.
5. Синтез: включва способността за обединение на вече анализирани части в едно цяло. Творческо приложение на знанията, водещо до създаването на нов продукт.	11. Работа с потребителски функции. Преобразуване на задачи, при които не са използвани функции в такива чрез използване на функции. Решаване на допълнителни задачи за създаване на програми.
6. Оценка: включва способността да се обобщава и оценява стойността на решена (синтезирана) задача. Оценката винаги трябва да се дава по точно определени критерии и норми.	12. Оценка на ефективността на методите за създаване на програми без и със използване на потребителски функции.

Всяко занятие започва с демонстрационен пример, който илюстрира прилагането на знанията, които трябва да бъдат усвоени по време на това занятие. Той е необходим за да повиши мотивацията и изостри любопитството на студентите към новия учебен материал. Следва визуална презентация с помощта на системата за динамично обучение по програмиране. Тя позволява да се разкрие механизмът на действие на определена програмна конструкция или алгоритъм. Чак след това се представя теоретичният материал (понятия, термини, структури от данни, алгоритми и програми) вербално от преподавателя. Решава се опорна задача, като по-добрият вариант е да не се ползва готов пример, а да се тръгва от нулата – анализ, моделиране, алгоритъм, програмиране. Набляга се на проблемни места, където студентите най-често допускат грешки. В края на този етап преподавателят остава отворени въпроси, по които студентите трябва да изразят становище по време на дискусиата. Тези три стъпки следват схемата на обучение на Джеръм Брунър. Същинската част на активното обучение стартира с дискусиата, в която студентите обсъждат под ръководството на преподавателя всички елементи на поднесения в различни форми в трите предишни етапа учебен материал, опитват се да дадат отговор на поставените от преподавателя въпроси, задават други въпроси, предлагат алтернативни решения. В края на този четвърти етап

студентите трябва да разбират и анализират готов програмен код. Той е подготовка за следващия етап, предвиждащ да се постигне целта синтез – самостоятелната практика, по времето на която трябва да реализират програма. Този етап е комбиниран с литературни справки в учебници и Интернет. За пълното ѝ осъществяване, преподавателят отговаря на въпроси за доизясняване на проблемни ситуации.



Фигура 8 Схема за провеждане на занятие по метода на активното обучение

Самоподготовката е съществена част от учебния план. Във ФМИ при ПУ се предвижда при провеждането ѝ студентите да отделят два пъти повече време в сравнение с аудиторните занятия за редовно обучение и пет пъти повече за задочно обучение. Като се вземат предвид тези данни и отсъствията на част от студентите от аудиторни занятия, създаването на инструментариум за самообучение и предлагането на достъпни и разбираеми литературни източници е от съществено значение. По време на самоподготовката студентите подготвят

домашни работи, курсови проекти и презентации на допълнителен учебен материал по темата на занятието. Възможността за комуникация с преподавателя и топлата обратна връзка на този етап са основен фактор за успешното му реализиране.

Постигнати резултати:

1. Направен е анализ на основните преимущества на обучението чрез забавление (edutainment).
2. Създадени са авторски компютърни средства и методически инструментариум за активно обучение по програмиране в стил **edutainment**.
3. Разработен е дидактически модел за активно обучение посредством прилагане на предложения методически инструментариум.
4. Предложена е методика за активно обучение по програмиране чрез използване на ИКТ.

ГЛАВА 3. ОРГАНИЗАЦИЯ, ПРОВЕЖДАНЕ, РЕЗУЛТАТИ И АНАЛИЗ НА РЕЗУЛТАТИТЕ ОТ ПЕДАГОГИЧЕСКИ ЕКСПЕРИМЕНТ

3.1 Организация на педагогическия експеримент

За проверка на основната хипотеза на провежданото изследване, че повишаване успеваемостта на студентите в овладяване на базови дисциплини по програмиране и мотивацията им за тяхното изучаване може да се осъществи чрез управление на процеса на обучение по тези дисциплини в условията на конструктивистка и високотехнологична среда, бе проведен педагогически експеримент през втори триместър на учебната 2013-2014 година със студенти от първи курс, специалност Бизнес информационни технологии (БИТ) по учебната дисциплина Програмиране. Етапите на експеримента [5] бяха следните:

1. Констатиращ
2. Формиращ
3. Заключителен

На първия етап, на базата на поставените в учебната програма цели, бе разработен стандартизиран дидактически тест [4], който се подложи на анализ за надеждност и валидност. От този анализ става ясно, че включените в теста задачи имат приемливи стойности на изследваните показатели, като единственият оказал се силно проблематичен въпрос е следният:

5. Какъв е резултатът от операцията $10+5/2-6$?
- а) 1.5
 - б) 6.5
 - в) 6
 - г) Ще даде грешка при изпълнение

За езика C++, който е обект на теста верният отговор е в) 6. Масовите отговори, които са дали тестваните студенти, обаче, са б) и г). Този съществен пропуск в знанията на студентите при входното ниво би могъл да доведе до редица семантични грешки при създаването на програми от тях и затова не смятаме този въпрос за излишен.

Бе разработен модел за количествена оценка на придобитите компетентности. На базата на административно оформените подгрупи за лабораторни занятия бяха сформирани експериментална група (ЕГ), състояща се от 29 студента, и контролна група (КГ), наброяваща 32 студента. С помощта на Тест 1 бе оценено входното ниво по Програмиране на двете групи. По този начин бе направена обективна оценка на нивото, което студентите са достигнали в проведения през първи триместър курс по Основи на компютърната информатика.

По време на формирания експеримент се проведе обучение на двете групи, като КГ се обучаваше по стандартна методика, а с ЕГ се проведе активно обучение с използване на ИКТ. На петата седмица бе осъществен текущ контрол на двете групи с Тест 2, за да се проследи степента на усвояване на компетентностите по време на обучението, след което да се отстранят констатираните пропуски.

Заключителният експеримент се проведе в края на триместъра. Усвоените знания на студентите бяха проверени с Тест 3, резултатите от който бяха подложени на статистически анализ.

3.2 Резултати и анализ на резултатите от експеримента

Смятаме, че използването на тест с множествен избор не може да бъде единствен критерий за проверка на успеваемостта на студентите по време на тяхното обучение по програмиране. От съществено значение за тази проверка е да оценим създадения формален модел на задачата, последвани от алгоритъм и програмна реализация. Задачният подход е важна част от обучението и по други ИТ дисциплини [10, 15]. За осъществяването на тази идея бяха разработени критерии за точкуване на реализираните фрагменти от компютърната програма, което дава възможност за количествена оценка на постигнатите от обучаемите образователни цели според таксономията на Блум.

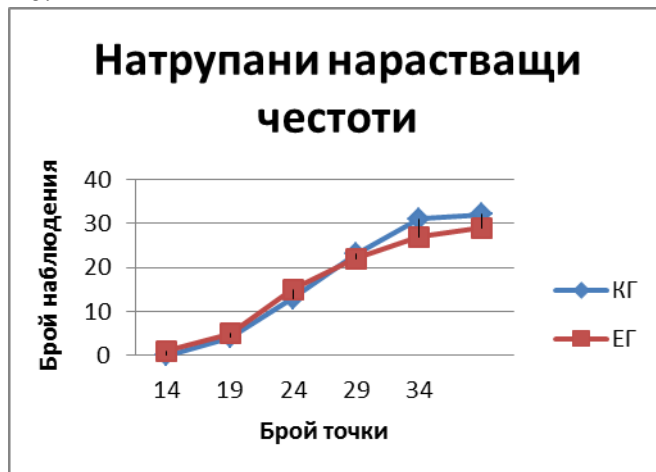
3.2.1 Констатиращ експеримент

Тестът за проверка на входното ниво се състои от: 12 въпроса с четири отговора, един от които е верен; задача от типа „Открийте грешките в програмата” със свободен отговор, която носи от 0 до 19 точки; задача за създаване на програма, основана на изучавания материал от предходния триместър. Задачата носи от 0 до 21 точки в зависимост от постигнатите от студента резултати, оценени посредством модел за количествена оценка. Крайната оценка се изчислява по формулата $ROUND(2+T*4/M;1)$, където Т е полученият брой точки от студента, М – максималния брой точки (в случая 52), закръглени до първия знак след десетичната запетая. Тестът се провежда в рамките на един учебен час. Резултатите от теста на студентите от двете групи са представени в следващата таблица.

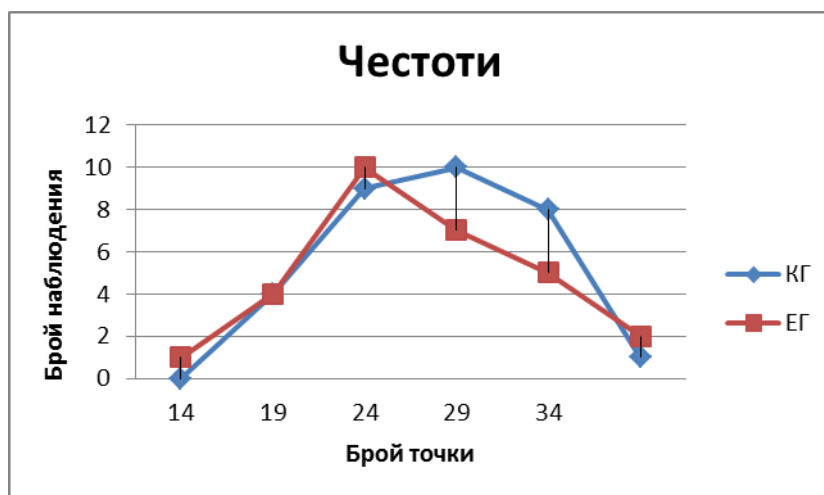
	Контролна група			Експериментална група		
Интервали	Честота	Натрупана честота	Относителна честота	Честота	Натрупана честота	Относителна честота
0-14	0	0	0.00%	1	1	6.90%
15-19	4	4	12.50%	4	5	17.24%
20-24	9	13	40.63%	10	15	51.72%
25-29	10	23	71.88%	7	22	75.86%
29-34	8	31	96.88%	5	27	93.10%
Над 34	1	32	100.00%	2	29	100.00%

Таблица 1 Обобщени резултати на двете групи при Тест 1

За да продължим изследването, трябваше да проверим дали двете групи имат резултати, които не могат статистически да отличат едната група от другата. Сравнявайки графиките на натрупаните честоти на двете групи, както и графиките на честотните им разпределения можем да изкажем предположение, че няма разлика в представянето им на входното ниво при провеждане на Тест 1. На Фигура 9 преплитането на двете криви е ясен знак за липсата на съществени различия. Няма явно доминиране в резултатите на никоя група и при разглеждане на Фигура 10.



Фигура 9 Натрупани честоти при Тест 1



Фигура 10 Честотно разпределение при Тест 1

За да сме сигурни в предположенията си, направихме и необходимите за целта статистически тестове на Фишер и Стюдънт. Резултатът показва, че средноаритметичните постижения на студентите от двете групи тествани лица при Тест 1 (входно ниво) са статистически неразличими стойности

3.2.2 Заключителен експеримент

Тестът за проверка на резултатите от проведеното обучение се състои от 12 въпроса и задача за създаване на програма основана на изучавания материал през изминалия тримесър. Задачата носи от 0 до 40 точки в зависимост от постигнатите от студента резултати, оценени посредством модел за количествена оценка. Крайната оценка се изчислява по формулата $\text{ROUND}(2+T*4/M;1)$, където T е полученият брой точки от студента, M – максималния брой точки (в случая 52), закръглени до първия знак след десетичната запетая. Тестът се провежда в рамките на два учебни часа.

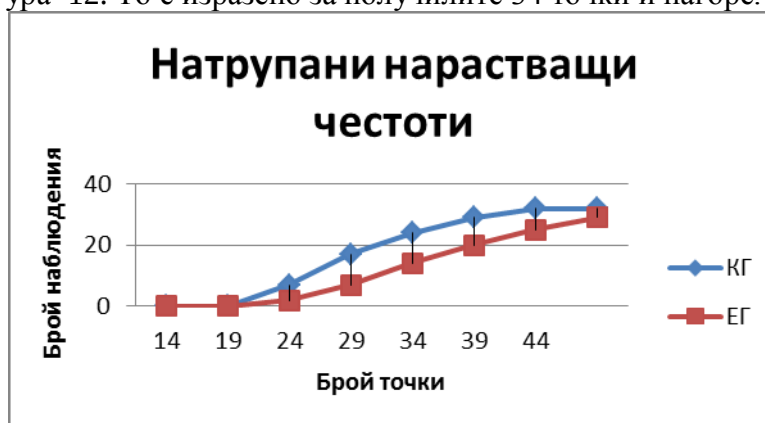
Резултатите от теста на студентите от двете групи са представени в следващата таблица.

Интервали	Контролна група			Експериментална група		
	Честота	Нагрупана честота	Относителна честота	Честота	Нагрупана честота	Относителна честота
0-14	0	0	0.00%	0	0	0.00%
15-19	0	0	0.00%	0	0	0.00%
20-24	7	7	21.88%	2	2	6.90%
25-29	10	17	53.13%	5	7	24.14%
29-34	7	24	75.00%	7	14	48.28%
35-39	5	29	90.63%	6	20	68.97%
40-44	3	32	100.00%	5	25	86.21%
Над 44	0	32	100.00%	4	29	100.00%

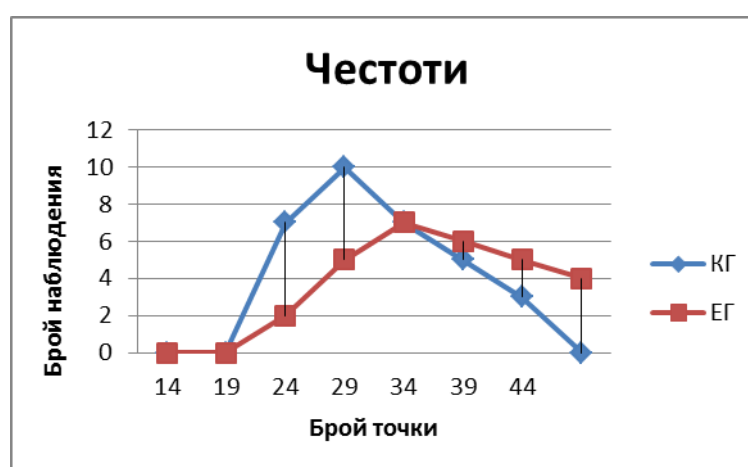
Таблица 2 Обобщени резултати на двете групи при Тест 3

За да продължим изследването, трябваше да проверим дали резултатите на ЕГ са по-добри от тези на КГ. За тази цел е необходима проверка дали те могат статистически да отличат едната група от другата. Сравнявайки графиките на натрупаните честоти на двете групи, както и графиките на честотните им разпределения можем да изкажем предположение, че има съществена разлика в представянето им при провеждане на Тест 3. На Фигура 11 по-голямото нарастване на кривата на КГ е ясен знак за по-слабите резултати на тази група в

сравнение с тези на ЕГ. Явно доминиране в резултатите на ЕГ се наблюдава и при разглеждане на Фигура 12. То е изразено за получените 34 точки и нагоре.



Фигура 11 Натрупани честоти при Тест 3



Фигура 12 Честотно разпределение при Тест 3

За да сме сигурни в предположенията си направихме и необходимите за целта статистически тестове. Те показаха, че средноаритметичните постижения на студентите от ЕГ надхвърлят тези на студентите от КГ при провеждането на Тест 3 на базата на статистически различни стойности. Проведеният заключителен експеримент потвърди основната хипотеза, че повишаване успеваемостта на студентите в овладяване на базови дисциплини по програмиране и мотивацията им за тяхното изучаване може да се осъществи чрез управление на процеса на обучение по тези дисциплини в условията на конструктивистка и високотехнологична среда.

ЗАКЛЮЧЕНИЕ И ИЗВОДИ

Изпълнена е основната цел на дисертационното изследване – да се разработи методика за провеждане на активно обучение по базови дисциплини по програмиране и инструментариум за нейното прилагане, които да повишат мотивацията и успеваемостта на студентите. Основните приноси на дисертационния труд са от научно-приложен и приложен характер:

- I. Анализирани са текущият модел на обучение по програмиране, основните затруднения на студентите при усвояване на базови знания и необходимостта от прилагане на активно обучение.
- II. Чрез проучване на научно-приложна литература са анализирани методите и средствата за провеждане на активно обучение и са определени основни учебни дейности за прилагането му в обучението по програмиране.
- III. Разработен е дидактически модел и методически инструментариум за провеждане на активно обучение в базовите дисциплини по програмиране.

IV. Създадени са подходящи компютърни методи и средства за подпомагане на модела за активно обучение.

V. Разработени са и са апробирани дидактически тестове, критерии и показатели за оценка ефективността на обучението проведено по описаната методика.

Връзката между приносите, задачите, мястото на описание в дисертационния труд и направените публикации са следните:

Принос	Задача	Секция в дисертацията	Публикация
I.	1	1.2	5
II.	1	1.3, 1.4	4
III.	2	2.1	1, 4
IV.	2	2.2, 2.3	2,3, 6
V.	3	3.1, 3.2	

ПЕРСПЕКТИВИ ЗА РАЗВИТИЕ

Методиката е експериментирана и подобрявана в продължение на 5 години със студенти от ФМИ (специалност Математика и информатика и Бизнес информационни технологии) и Биологически факултет (специалност Биоинформатика). Процесът на актуализация е постоянен.

В перспектива създадената методика за активно обучение би могла да се разшири и приложи:

- в други университетски курсове в областта на информатиката и ИТ.
- в профилираната подготовка по информатика и ИТ в училище.

Методиката може да бъде подкрепена с разработка на повече практически задачи, решаването на които да доближи студентите до реална работна среда. Това още повече би увеличило тяхната мотивация.

Перспективите за усъвършенстване на авторската система за обучение са насочени в няколко направления.

Първото е свързано с допълнително улеснение на обучаемия, посредством добавяне на програмни примери към всеки отделен модул, показвайки програмното решение на съответната задача (както в случая е реализирано само за функциите на линеен списък). Предвидено е поставянето на бутон „Виж пример” на всеки модул, като при натискането му ще се отваря отделен прозорец с няколко примерни реализирани програми, съответстващи на използвания алгоритъм.

Второто направление е по-сложно и предвижда построяване на визуално представяне на алгоритъма на базата на създаден програмен фрагмент от самия обучаем, както и помощник за създаване на елементарен програмен текст. В началото, обучаемият ще работи само с визуални компоненти, които автоматично ще генерират програмен код. Така дори лице, което няма представа от програмиране, ще бъде способно да създава елементарни програми.

Съществува възможност и за доразвиването на програмната система до уеб приложение, като по този начин ще бъде осигурен по-голям достъп до създадените компютърни средства за обучение.

Резултатите от дисертационния труд могат да бъдат успешно мултиплицирани от други преподаватели в областта на ИКТ. Разработеният инструментариум и неговите периодични актуализации са достъпни след регистрация на адрес <http://krushkov.fmi-plovdiv.org/>.

АПРОБАЦИЯ

Части от дисертационния труд са изготвени в изпълнение на задачи по следните проекти:

1. Университетски проект „Междуфакултетен разпределен център за електронно обучение“, договор №ИС-М-4/2008, (2008–2010 г.).

2. Научен проект № РС09-ФМИ-014/17.04.2009 (2009-2010) към фонд "Научни изследвания" при ПУ "Паисий Хилендарски": Разработване на интелигентни компютърни методи и средства за обучение.
3. Научен проект НИ11-ФМИ-004 (2011-2012) към фонд "Научни изследвания" при ПУ "Паисий Хилендарски": "Разработка и приложение на иновативни ИКТ за провеждане на качествени конкурентноспособни научни изследвания и цялостно осъвременяване процеса на обучение във ФМИ".
4. Научен проект НИ13 ФМИ-002 (2013-2014) към фонд "Научни изследвания" при ПУ "Паисий Хилендарски": „Интеграция на ИТ в научните изследвания по математика, информатика и педагогика на обучението”.

Части от дисертационния труд са докладвани на следните научни сесии и конференции:

1. XXXIV пролетна конференция на СМБ, 6-9 април 2005г.,
2. XXXIX пролетна конференция на СМБ, 6-10 април 2010 г., Албена.
3. Anniversary International Conference REMIA 2010, December 10-12, 2010, Plovdiv.
4. Научна сесия „Дни на науката 2011” на СУБ-секция Пловдив, 10-11.11. 2011.

ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД:

1. Крушков, Хр., **М. Крушкова**. “Въведение в обектноориентираното програмиране с Object Pascal и Delphi”, XXXIV пролетна конференция на СМБ, 6-9 април 2005г., Боровец, стр. 339-343, ISBN 954-8880-19-9.
2. Крушков, Хр., **М. Крушкова**, В. Атанасов, М. Крушкова, “Компютърна система за обучение по програмиране”, XXXIX пролетна конференция на СМБ, 6-10 април 2010 г., Албена, стр.354-358, ISSN 1313-3330.
3. **Krushkova, M.**, Y. Stoyanova, Hr. Krushkov, “Teaching Subroutines: as Early as Possible”, Anniversary International Conference REMIA 2010, December 10-12, 2010, Plovdiv, 451-457, ISBN 978-954-423-648-9.
4. Крушкова Д., Х. Крушков, В. Атанасов, **М. Крушкова**, М. Крушкова, "EDUTAINMENT (обучение чрез забавление) в програмирането", Научни трудове на СУБ, т. XIV, Пловдив, 10-11. ноември 2011 г., стр.259-262, ISSN 1311-9192.
5. **Крушкова М.**, А. Рахнев, Х. Крушков, "Съвременни методи и средства за активно обучение", годишно научно-методическо списание „Образование и технологии”, бр. 5, 2014 г., стр. 211-215 (ел. вариант стр. 427-433), ISSN 1314 – 1791 (Print).
6. Крушков Хр., А. Рахнев, **М. Крушкова**, "Обучение в стил Edutainment с използване на компютърна графика", научно-методическо списание „Математика и информатика”, бр. 4, 2014 г., стр.364-383, ISSN 1314 – 8532 (Online), ISSN 1310 – 2230 (Print).

БИБЛИОГРАФИЯ

- [1] Ангелова Е., Р. Маврова, "За професионалната подготовка на бъдещите учители по математика и информатика", *Юбилейна национална научна конференция с международно участие „Традиции, посоки, предизвикателства”*, Смолян, 2012, стр. 133-138.
- [2] Ангелова, Е., "Подготовка на учители за обучение на ученици по информационни технологии", София, Дисертация за присъждане на образователната и научна степен "доктор" 2009.
- [3] Асенова, П., "Информатика. Приложение на информационните и комуникационните технологии в обучението", *Методология и технология за създаване на държавните образователни изисквания*. Национален институт по образование, 2000, стр. 187 – 195.

- [4] Бижков, Г., *Теория и методика на дидактическите тестове*. София: Просвета, 1992.
- [5] Бижков, Г., В. Краевски, *Методология и методи на педагогическите изследвания*. София: УИ „Св. Климент Охридски“, 2002.
- [6] Буч Г., *Объектно-ориентированное проектирование с примерами применения*. Москва: Конкорд, 1992.
- [7] Вълчанов, Н., А. Илиев, "Общ подход при разработка на уеб базирани системи за управление на анкети", *Националната конференция „Образованието в информационното общество“*, Пловдив, 2010, стр. 54–60.
- [8] ГД "Образование и култура". (2008) ЕВРОПЕЙСКА КВАЛИФИКАЦИОННА РАМКА ЗА УЧЕНЕ. [Online]. HYPERLINK
["http://ec.europa.eu/education/pub/pdf/general/eqf/broch_bg.pdf"](http://ec.europa.eu/education/pub/pdf/general/eqf/broch_bg.pdf)
http://ec.europa.eu/education/pub/pdf/general/eqf/broch_bg.pdf
- [9] Гроздев С., К. Гъров, "За системите от опорни задачи при подготовката за участие в олимпиади по информатика (комбинаторни обекти и алгоритми)", *XXXVII пролетна конференция на СМБ*, Боровец, 2008, стр. 304-311.
- [10] Гроздев, С., Е. Ангелова, "Задачният подход при подготовка на учители за преподаване на "Информационни технологии" в училище" *Педагогика*, № 2, стр. 16-25, 2010.
- [11] Гроздев, С., С. Дойчев, "Математическите игри като средство за откриване на математически таланти" in *XXXVIII пролетна конференция на СМБ*, Боровец, 2009, стр. 237-244.
- [12] Гроздев, С., Т. Терзиева, "Визуализация методов сортировки массивов" *Электронный журнал Российской Академии Образования "Информационная среда образования и науки"*, vol. 5, 2011.
- [13] Гроздев, С., Т. Терзиева, "Исследование концепции алгоритми-ческого мышления при обучении информатике", *Международной научно-практической конференции „Информатизация образования 2011“*, Елец, 2011, стр. 112 - 119.
- [14] Гроздев, С., Т. Терзиева, "Статичные и динамичные средства для визуализации методов сортировки масивов" *Педагогическая информатика*, Выпуск 1, 2012.
- [15] Гъров, К., "За задачите в обучението по информатика и информационни технологии", *Национална конференция "Образованието в информационното общество"*, Пловдив, 2010, стр. 95-101.
- [16] Гюдженев, И., *Методиките на обучение във висшите училища. Необходимост и възможност*. Благоевград: УИ ЮЗУ „Неофит Рилски“, 2007.
- [17] Дончев, И., "Систематизиране на специфичните за C++ трудности при усвояването на клас и обект в обучението по програмиране", *XXXVIII пролетна конференция на СМБ*, Боровец, 2009, стр. 257-267.
- [18] Дурева, Д., "Метод на аналогията при реализацията на дидактическия принцип за достъпност в обучението по Информатика и ИТ" сб. *"Математика и математическо образование"*, 2004, стр. 329-333.
- [19] Дурева, Д., Г. Тупаров, "За подготовката на студентите - бъдещи учители по информатика и информационни технологии", *Сборник доклади на Национална конференция "Образованието в информационното общество"*, Пловдив, 2011, стр. 337-341.
- [20] Дурева-Тупарова, Д., *Проблеми по методиката на обучение по информатика и информационни технологии*. Благоевград: Университетско издателство "ЮЗУ Неофит Рилски", 2003.

- [21] Дурева-Тупарова, Д., "Реализация на електронно-базирани курсове в магистърска програма за педагози", *39-а пролетна конференция на СМБ*, Албена, 2010, стр. 318-324.
- [22] Дюкре, Ж., "Конструктивистки подходи: приложение и перспективи в областта на образованието" *Перспективи*, vol. XXXI, № 2, стр. 109-125, 2001.
- [23] Иванов, К., "За ефективността на обучението по обектно-ориентирано програмиране", *XXXVIII пролетна конференция на СМБ*, Боровец, 2009, стр. 282-288.
- [24] Илиев, А., Н. Вълчанов, Т. Терзиева, "Споделен опит от използване на софтуерна тестова система при провеждане на изпити в курса по Информационно моделиране", *Юбилейната конференция „Науката, образованието и времето като грижа“*, Смолян, 2007, стр. 77-81.
- [25] Крушков, Хр., *Програмиране на C++*. Пловдив: Коала Прес, 2010.
- [26] Крушков, Хр., А. Илиев, *Програмиране на Паскал*. Пловдив: Университетско издателство "Паисий Хилендарски", 2007.
- [27] Крушков, Хр., М. Крушкова, "Въведение в обектноориентираното програмиране с Object Pascal и Delphi", *XXXIV пролетна конференция на СМБ*, Боровец, 2005, стр. 339-343.
- [28] Крушков, Хр., М. Крушкова, В. Атанасов, М. Крушкова, "Компютърна система за обучение по програмиране", *XXXIX пролетна конференция на СМБ*, Албена, 2010, стр. 354-358.
- [29] Крушков, Хр., М. Крушкова, Д. Коленцов, Д. Крушкова, Й. Стойнова, "Разработване на интелигентни компютърни методи и средства за обучение", *Научни трудове на СУБ, т. VII*, Пловдив, 2009, стр. 412-415.
- [30] Крушков, Хр., М. Крушкова, Д. Крушкова, "Как да подобрим обучението по програмиране", *Национална конференция "Образованието в информационното общество"*, Пловдив, 2010, стр. 110-116.
- [31] Крушкова Д., Х. Крушков, В. Атанасов, М. Крушкова, М. Крушкова, "EDUTAINMENT (обучение чрез забавление) в програмирането" *Научни трудове на СУБ*, vol. XIV, стр. 259-262, Nov. 2011.
- [32] Пиаже Ж., *Избранные психологические труды: Психология интеллекта*. Москва: Просвещение, 1994.
- [33] Рахнев, А., "Интензификация на обучението по математика и информатика чрез използване на информационни технологии" Пловдовски Университет "Паисий Хилендарски", София, Хабилизационен труд за получаване научното звание „професор“ 2010.
- [34] Рахнев, А., "Метод на решетото" *Математика*, бр. 5, стр. 38-43, 1988.
- [35] Рахнев, А., "Още за метода на решетото" *Математика*, бр. 7, стр. 36-37, 1988.
- [36] Рахнев, А., К. Гъров, "Някои задачи по програмиране, свързани с числата на Фибоначи" *Математика*, бр. 8, стр. 35-37, 1988.
- [37] Старибратов, И., Е. Ангелова, "Методически подходи за обучение чрез използване на електронни учебни ресурси", *IV национална конференция „Образованието в информационното общество“*, Пловдив, 2011, стр. 329-336.
- [38] Терзиева, Т., "Развитие на алгоритмичното мислене в обучението по информатика" Пловдив, Дисертационен труд за присъждане на образователна и научна степен "доктор" 2012.
- [39] Agre, G., "SINUS – A Semantic Technology Enhanced Environment for Learning,

Humanities” vol. 12, no. 4, 2012.

- [40] Alanis-Funes, G.J., Neri, L., Noguez, J., "Virtual collaborative space to support active learning” in *Proceedings - Frontiers in Education Conference, FIE*, Campus Ciudad de Mexico, Mexico ; Tecnológico de Monterrey, Mexico, 2011.
- [41] Aleksieva-Petrova, A., Petrov, M., "Mapping games and gaming style with learning goals and style” in *1st International Multi-Conference on Innovative Developments in ICT, INNOV 2010*, Athens, 2010, pp. 169-172.
- [42] Alemán, J.L.F., Oufaska, Y., "SAMTool, a tool for deducing and implementing loop patterns” in *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*, Espinardo, Spain, 2010, pp. 68-72.
- [43] Anderson, N., Gegg-Harrison, T., "Learning computer science in the "comfort zone of proximal development"“ in *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, Winona, United States, 2013, pp. 495-500.
- [44] Angelova, E., A. Rahnev, "Boosting teaching and learning effectiveness in training teachers of Information Technology” *Scientific Works, Plovdiv University*, vol. 36, no. 3, pp. 5-18, 2009.
- [45] Angelova, E., I. Staribratov, "On a Virtual Learning Environment” in *Anniversary International Conference REMIA2010*, Plovdiv, 2010, pp. 403-410.
- [46] Bloom, B.S., *Taxonomy of Educational Objectives: The Classification of Educational Goals*. New York: David McKay Company, 1969.
- [47] Bontchev, B., Vassileva, D., "Courseware authoring for adaptive e-learning” in *International Conference on Education Technology and Computer, ICETC 2009*, Singapore, 2009, pp. 176-180.
- [48] Bontchev, B., Vassileva, D., Chavkova, B., Mitev, V., "Architectural design of a software engine for adaptation control in the ADOPTA e-learning platform” in *International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, CompSysTech'09*, Ruse, 2009.
- [49] Brooks, D.C., "Space matters: The impact of formal learning environments on student learning” *British Journal of Educational Technology*, vol. 42, no. 5, pp. 719-726, 2011.
- [50] Bruner, J., "Studies in Cognitive Growth” in *Studies in Cognitive Growth*. New York: John Wiley & Sons Inc., 1966, p. 374.
- [51] Clark, J., Fletcher, A., Fritz, T., Schreiver, J., "Modular educational game system: A customizable framework for learning” in *16th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational and Serious Games, CGAMES'2011*, Louisville, 2011, pp. 248-253.
- [52] Cooper, S., Dann, W., Pausch, R., "Alice: An introductory tool for teaching programming concepts” in *CCSCNE*, Ramapo, NJ, 2000.
- [53] Da Silva Gomes De Oliveira, E., Nunes, J.M.G., "Cognitivism and e-learning: A view from the learning theories of Ausubel and Bruner” in *Proceedings of the IADIS International Conference e-Learning 2011, Part of the IADIS Multi Conference on Computer Science and Information Systems 2011, MCCSIS 2011*, Rio de Janeiro, Brazil, 2011.
- [54] Davis, J., "Experiences with just-in-time teaching in systems and design courses” *SIGCSE Bulletin Inroads*, vol. 41, no. 1, pp. 71-75, 2009.
- [55] Detmer, R., Li, C., Dong, Z., Hankins, J., "Incorporating real-world projects in teaching computer science courses” in *Proceedings of the Annual Southeast Conference*, Murfreesboro, United States, 2010.

- [56] Ellis, H.J.C., Chua, M., Jadud, M.C., Hislop, G.W., "Learning through open source participation" in *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, Springfield, St. Meadville, Philadelphia - United States, 2011, pp. 83-84.
- [57] García-Mateos, G., Fernández-Alemán, J.L., "A course on algorithms and data structures using on-line judging" in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, Paris, France, 2009, pp. 45-49.
- [58] Gehringer, E.F., Miller, C.S., "Student-generated active-learning exercises" in *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, Chattanooga, United States, 2009, pp. 81-85.
- [59] Grozdev, S., "On the Visualness in Mathematics Education" in *4th Mediterranean Conference on Mathematics Education*, Palermo – Italy, 2005, pp. 303 – 313.
- [60] Guo, M., Chai, T., Qian, K., "Design of online runtime and testing environment for instant java programming assessment" in *ITNG2010 - 7th International Conference on Information Technology: New Generations*, Marietta, GA, United States, 2010, pp. 1102-1106.
- [61] Hakimzadeh, H., Adaikkalavan, R., Batzinger, R., "Successful implementation of an active learning laboratory in computer science" in *Proceedings ACM SIGUCCS User Services Conference*, South Bend, IN, United States, 2011.
- [62] Herold, M.J., Lynch, T.D., Ramnath, R., Ramanathan, J., "Student and instructor experiences in the inverted classroom" in *Proceedings - Frontiers in Education Conference, FIE*, Columbus, United States, 2012.
- [63] Hoffman, D., Lu, M., Pelton, T., "A web-based generation and delivery system for active code reading" in *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, Victoria, Canada, 2011, pp. 483-488.
- [64] Hoic-Bozic, N., Mornar, V., Boticki, I., "A blended learning approach to course design and implementation" *IEEE Transactions on Education*, vol. 52, no. 1, pp. 19-30, 2009.
- [65] Kember, D., Ho, A., Hong, C., "Characterising a teaching and learning environment capable of motivating student learning" *Learning Environments Research*, vol. 13, no. 1, pp. 43-57, 2010.
- [66] Kilic, H., Koyuncu, M., Rehan, M., "Designing senior graduation project course for computing curricula: An active learning approach" *International Journal of Engineering Education*, vol. 26, no. 6, pp. 1472-1483, 2010.
- [67] Kim, K., Sharma, P., Land, S.M., Furlong, K.P., "Effects of Active Learning on Enhancing Student Critical Thinking in an Undergraduate General Science Course" *Innovative Higher Education*, vol. 38, no. 3, pp. 223-235, 2013.
- [68] Kölling, M., "The Greenfoot Programming Environment" *ACM Transactions on Computing Education*, vol. 10, no. 4, p. Article number 14, 2010.
- [69] Kollmansberger, S., "Helping students build a mental model of computation" in *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*, Olympia, United States, 2010, pp. 128-131.
- [70] Köppe, C., "Continuous activity - A pedagogical pattern for active learning" in *ACM International Conference Proceeding Series*, Hogeschool Utrecht, Institute for Information and Communication Technology, Netherlands, 2012.
- [71] Krushkova, M., Y. Stoyanova, Hr. Krushkov, "Teaching Subroutines: as Early as Possible" in *Anniversary International Conference REMIA 2010*, Plovdiv, 2010, pp. 451-457.

- [72] Marasheva-Delinova, I., E. Angelova, "Main Activities of the Teacher and the Students in the Process of Project Work" *Scientific Works, Plovdiv University*, vol. 38, no. 3, pp. 79-86, 2011.
- [73] Mazur, E., *Peer Instruction: A User's Manual*, Series in Educational Innovation ed. NJ, Upper Saddle River: Prentice Hall, 1997.
- [74] Merriam-Webster-dictionary. [Online]. HYPERLINK "http://www.merriam-webster.com/dictionary/edutainment" <http://www.merriam-webster.com/dictionary/edutainment>
- [75] Miller, L.D., Soh, L.-K., Samal, A., Nugent, G., Kupzyk, K., Masmaliyeva, L., "Evaluating the use of learning objects in CS1" in *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, Lincoln, United States, 2011, pp. 57-62.
- [76] Naps, T., "JHAVE: Supporting algorithm visualization" *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 49-55, 2005.
- [77] Novak, Gregor M., Patterson, Evelyn T., Gavrinn, Andrew D., and Christian, Wolfgang, *Just-In-Time Teaching: Blending Active Learning with Web Technology*. Upper Saddle River, NJ: Prentice Hall, 1999.
- [78] Ola, A.G., Bada, A.O., Omojokun, E., Adekoya, A.A., "Creating the classroom environment for asynchronous E-learning" *WSEAS Transactions on Information Science and Applications*, vol. 7, no. 2, pp. 207-219, 2010.
- [79] Pahl, C., Kenny, C., "Interactive correction and recommendation for computer language learning and training" *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 6, pp. 854-866, 2009.
- [80] Pasko, A., Adzhiev, V., "Constructive function-based modeling in multilevel education" *Communications of the ACM*, vol. 59, no. 9, pp. 118-122+10, 2009.
- [81] Pausch, R., Burnette, T., Capeheart, A.C., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Gossweiler, R., Koga, S., White, J., "Alice: Rapid Prototyping System for Virtual Reality" *IEEE Computer Graphics and Applications*, 1995.
- [82] Qian, K., Dan Lo, C.-T., Yang, L., Liu, J., "Inquiry-based active learning in introductory programming courses" in *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*, Southern Polytechnic State University, University of Tennessee, Metropolitan State University - United States, 2010, p. 312.
- [83] Regueras, L.M., Verdu, E., Mucoz, M.F., Perez, M.A., de Castro, J.P., Verdu, M.J., "Effects of competitive E-learning tools on higher education students: A case study" *IEEE Transactions on Education* 52, vol. 52, no. 2, pp. 279-285, 2009.
- [84] Riek, L.D., "Embodied computation: An active-learning approach to mobile robotics education" *IEEE Transactions on Education* 56 (1), pp. 67-72, 2013.
- [85] Santos, A., A.Gomes, A.Mendes, "Integrating New Technologies and Existing Tools to Promote Programming Learning" *Algorithms* , vol. 3, no. 2, pp. 183-196, 2010.
- [86] Schiller, S.Z., Goodrich, K., Gupta, P.B., "Let Them Play! Active Learning in a Virtual World" *Information Systems Management*, vol. 30, no. 1, pp. 50-62, 2013.
- [87] Schweitzer, D., Boleng, J., "A simple machine simulator for teaching stack frames" in *SIGCSE'10 - Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, USAFA, United States , 2010, pp. 361-365.
- [88] Shaffer, C.A., Akbar, M., Alon, A.J.D., Stewart, M., Edwards, S.H., "Getting algorithm visualizations into the classroom" in *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, Virginia, United States, 2011,

pp. 129-134.

- [89] Simon, B., Bales, E., Griswold, W.G., Cooper, S., "Case study: Faculty professional development workshops for innovation diffusion" in *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, San Diego, Stanford - United States, 2011, pp. 673-678.
- [90] Song, J., S. Hahn, K. Tak, J. Kim, "An intelligent tutoring system for introductory C language course" *Computers & Education*, vol. 28, no. 2, pp. 93-102, 1997.
- [91] Taylor, D.S., Lurie, A.F., Horstmann, C.S., Johnson, M.B., Sharma, S.K., Yin, E.C., "Predictive vs. passive animation learning tools" in *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, Stanford, United States , 2009, pp. 494-498.
- [92] Trivedi, B.H., "Impact of effective pedagogy on information technology students with weak academic backgrounds" *International Journal of Engineering Education*, vol. 26, no. 3, pp. 708-715, 2010.
- [93] Tront, J.G., "Workshop - Developing active learning classroom exercises for use with tablet PCs" in *Proceedings - Frontiers in Education Conference, FIE*, Virginia Tech., United States, 2011.
- [94] Tuparov, G., D. Tuparova, A. Tsarnakova, "Using Interactive Simulation-Based Learning Objects in Introductory Course of Programming" *Procedia - Social and Behavioral Sciences*, vol. 46, pp. 2276-2280, 2012.
- [95] Tuparov, G., D. Tuparova, V. Jordanov, "Teaching Sorting and Searching Algorithms through Simulation-based Learning Objects in an Introductory Programming Course" *Procedia - Social and Behavioral Sciences*, vol. 116, pp. 2962–2966, February 2014.
- [96] Tuparova, D., G.Tuparov, "Implementation of blended learning scenarios for training of school teachers" in *14th International Conference on Interactive Collaborative Learning (ICL)*, Piestany, 2011, pp. 285-289.
- [97] Valchanov, N., A. Iliev, T. Terzieva, "One Methodological Approach for Teaching in Information Modeling" in *Синергетика и рефлексия в обучението по математика*, Бачиново, 2010, pp. 487–494.
- [98] Vega, C., Jimenez, C., Villalobos, J., "A scalable and incremental project-based learning approach" *Education and Information Technologies*, vol. 18, no. 2, pp. 309-329, 2013.
- [99] Velazquez-Iturbide Angel, J. , Pérez-Carrasco, A., "Active learning of greedy algorithms by means of interactive experimentation" in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, Madrid, Spain, 2009, pp. 119-123.
- [100] Velázquez-Iturbide, J.A., Debdi, O., "Experimentation with optimization problems in algorithm courses" in *EUROCON 2011 - International Conference on Computer as a Tool - Joint with Conftele 2011*, Madrid, Spain, 2011.
- [101] Villalobos, J.A. , Calderon, N.A. , Jimenez, C.H., "Developing programming skills by using interactive learning objects" in *ITiCSE '09 Annual Conference on Innovation and Technology in Computer Science Education* , Paris, 2009, pp. 151-155.
- [102] Wang, G., "Active Learning through WWW: Just-in-Time Teaching in Digital Systems Design" in *ASEE Annual Conference & Exposition, AC-94*, Honolulu, 2007.

СЪДЪРЖАНИЕ

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД	3
Цели и задачи на дисертационния труд	4
Структура и обем на дисертационния труд	5
ГЛАВА 1. Теоретични основи на дисертационно-то изследване.....	5
1.1 Процесът на обучение – определение, методи на обучение	5
1.2 Проучване на студентското мнение.....	6
1.3 Дидактически основи на активното обучение	7
1.4 Резултати и изводи	8
ГЛАВА 2. Дидактически инструментариум за реализиране на методиката на обучение по програмиране чрез използване на ИКТ.....	9
2.1 Обучение чрез забавление (edutainment) в програмирането.....	9
2.1.1 Подобряване на обучението посредством “Edutainment”	10
2.1.2 Авторски средства за „Edutainment”.....	11
2.2 Модел за преподаване на подпрограми.....	12
2.2.1 Статичен подход за самообучение	12
2.2.2 Интерактивен модул за самообучение	12
2.3 Система за динамично обучение по програмиране	12
2.3.1 Обща идея.....	12
2.3.2 Базови алгоритми	14
2.3.3 Вектори.....	15
2.3.4 Основни алгоритми с матрици.....	17
2.3.5 Алгоритми с динамични структури от данни	18
2.3.6 Тестов модул	19
2.4 Модел за преподаване на обектно-ориентирано програмиране	19
2.4.1 Обектноориентирано програмиране.....	19
2.4.2 Задачи.	19
2.4.3 Компютърна графика	19
2.5 Методика за активно обучение с използване на ИКТ	21
ГЛАВА 3. Организация, провеждане, резултати и анализ на резултатите от педагогически експеримент.....	24
3.1 Организация на педагогическия експеримент	24
3.2 Резултати и анализ на резултатите от експеримента	25
3.2.1 Констатиращ експеримент.....	25
3.2.2 Заключителен експеримент	26
ЗАКЛЮЧЕНИЕ И ИЗВОДИ	27
БИБЛИОГРАФИЯ	29
СЪДЪРЖАНИЕ	36