

Пловдивски университет "Паисий Хилендарски"
Факултет по математика и информатика (ФМИ)
Катедра "Компютърни технологии"

Дипломна работа

на тема

Учебно съдържание по обектно- ориентирано програмиране в профилираната подготовка по информатика

Дипломант:

д-р Светлин Иванов Наков

Фак. №: 2001267019

Специалност: Педагогика на обучението по информатика и информационни
технологии в училище (ПОИИТУ)

Степен: **магистър**

Научен ръководител:

гл. ас. д-р Симеон Монов

Пловдив, юли, 2023 г.



Съдържание

Увод	5
Проблем	5
Цели на дипломната работа	6
Задачи, произтичащи от целите	7
Глава 1. Проучване и планиране.....	9
Училищно образование в България	9
Училищата в България.....	10
Начални и основни училища	11
Професионални и профилирани гимназии	11
Обучение по програмиране и ИТ в училищата.....	12
Програмиране и ИТ в началното и основното образование	12
Програмиране и ИТ в гимназиите: профили и професии.....	13
Колко ученици учат програмиране и ИТ в България?	13
Учебни планове по програмиране и ИТ в гимназиален етап.....	14
Задължителни, избираеми и факултативни дисциплини.....	14
Законова рамка и наредби за програмиране и ИТ в профилирана подготовка	15
Законова рамка и наредби за програмиране и ИТ в професионална подготовка	16
Профил “Софтуерни и хардуерни науки”	17
Рамков учебен план за профил “Софтуерни и хардуерни науки”	18
Модули по профилиращ предмет “Информатика”	20
Модули по профилиращ предмет “Информационни технологии”	20
Защо е некоректно заглавието “Софтуерни и хардуерни науки”?	21
Програмиране се учи с много практика, години наред!	21
Недостигът на часове по програмиране в профил “Софтуерни и хардуерни науки”	22
Защо е важен профил “Софтуерни и хардуерни науки”?	27



Модул “Обектно-ориентирано проектиране и програмиране”	27
Официална учебна програма за Модул 1 “ООП”	27
Несъвършенства на учебната програма от МОН	29
Недостигът на учебно съдържание по програмиране и ИТ	29
Добрата подготовка: ключ към качествено преподаване	30
От какво имат нужда учителите?	31
Недостиг на учебни материали и знания	31
Учебници по програмиране и ИТ	32
Учебниците на заместват учебните материали	32
Проект “Обучение за ИТ кариера”	33
Проект “Отворено учебно съдържание” на СофтУни Фондацията	35
Недостиг на учебни материали за профил “Софтуерни и хардуерни науки”	36
Недостиг на учебно съдържание за Модул 1 “ООП”	37
Какво е качествено учебно съдържание?	38
Защо е необходимо качествено учебно съдържание?	39
Проучване на нуждите на ИТ учителите (ноември 2022).....	40
Глава 2. Разработка на учебното съдържание.....	44
Учебна програма	44
Базов, разширен и експертен вариант на учебната програма	45
Разпределения на часовете в учебната програма	46
Структура на учебните ресурси.....	48
Хранилище на проекта	50
Свободен лиценз	52
Разработка на учебното съдържание	52
Теми от учебното съдържание.....	52
Учебни презентации.....	56
Задания за упражнения	59
Система за автоматична проверка	61



Задачи за практически изпити	62
Изпитни тестове във формат ДЗИ	63
Практически проекти.....	64
Статистика за свършеното.....	65
Глава 3. Внедряване и оценка на въздействието	66
Анкета от учители по ООП – май 2023 г.....	66
Участници в анкетата.....	66
Резултати от анкетата	67
Изводи от анкетата	70
Заклучение	71
Планове за развитие	72
Благодарности.....	73
Информационни източници.....	74



Увод

В настоящата дипломна работа са разработени и предоставени за безплатно ползване за българските ИТ учители **учебни програми и учебно съдържание** (учебни ресурси, презентации, демонстрации, упражнения, проекти, изпити, тестове) с високо качество, на български език, за преподаване в “**Модул 1. Обектно-ориентирано проектиране и програмиране**” от профил “Софтуерни и хардуерни науки” в профилираните гимназии.

Специализираната учебна подготовка по **профилиращите предмети** в българските профилирани гимназии се извършва във втори гимназиален етап от средното образование (в 11 и 12 клас). Един от най-търсените профили в днешно време е профил “**Софтуерни и хардуерни науки**”, в който водещите профилиращи учебни предмети са **информатика и информационни технологии (ИТ)**. Задължителното обучение по информатика според наредбите на МОН се състои от най-малко 4 модула, от които първият и най-важен модул е “**Модул 1. Обектно-ориентирано проектиране и програмиране**”. В него учениците усвояват основите на програмирането, работата с класове и обекти и принципите на ООП. Затова темата на дипломната работа е актуална и важна за българското училищно образование по информатика и ИТ.

Проблем

Профил “Софтуерни и хардуерни науки” се преподава в **над 150 български училища** [[Наков, 2021](#)]. ИТ учителите в тези училища са принудени **сами да разработват учебно съдържание**, за да преподават профилираните предмети, а това води до **претоварване** на учителите, липси и **несъвършенства в учебните материали** и в крайна сметка до по-ниско качество на обучението по информатика и ИТ.

Учебници по информатика съществуват, но те са предназначени най-вече за самостоятелна подготовка на учениците вкъщи. **За преподаване в клас е необходимо учебно съдържание:** добре разработени **учебни уроци** в готов за презентирание формат, заедно с добре разработени **задания за упражнения** с насоки към учениците и учителите, **изпитни материали** и други учебни ресурси.



Към 2022 г. **не е налично добре разработено учебно съдържание** на български език за обучение по информатика за “Модул 1. Обектно-ориентирано проектиране и програмиране” от профил “Софтуерни и хардуерни науки” и това води до **влошено качество на обучението по информатика в профилираните гимназии**.

Цели на дипломната работа

Настоящата дипломна работа има за цел **да подпомогне българските ИТ учители** от системата на средното образование в профилираните гимназии и паралелки, като им предостави безплатно добре разработени **учебни програми** и качествено **учебно съдържание** за преподаване в първия и най-важен модул от профил “Софтуерни и хардуерни науки”, а именно **“Модул 1. Обектно-ориентирано проектиране и програмиране”**.

Разработването на учебно съдържание е **тежка и времеотнемаща задача**, с която малцина учители се захващат. За да се изготви качествено учебно съдържание (**учебни материали и ресурси** за преподаване в клас, презентационни материали за всеки урок с подходящи демонстрационни примери, задачи за упражнения за всеки урок с указания и решения, системи за оценяване, изпитни задания, практически проекти, изпитни въпроси за държавен зрелостен изпит и други учебни ресурси) са необходими задълбочени **умения**, сериозен учителски **опит** и най-трудното: да се отделят значимо количество **време и усилия**.

Учителите рядко имат уменията, опита и времето, за да разработят качествено учебно съдържание и затова **преподават по каквото намерят** за подходящо: книги, учебници, материали от Интернет (най-често на английски език), видео уроци, учебни платформи (най-често на английски език) и други източници, които обикновено не съответстват точно на учебните програми и изисквания на МОН.

Чрез изграждането на **качествени учебни програми и ресурси за преподаване** и пренасяне на **добре изпитани образователни практики** от автора на настоящата дипломна работа (д-р Светлин Наков) към българските ИТ учители целим значително **да подпомогнем учителите в тяхната образователна кауза** и **да повишим качеството на обучението по програмиране** в профилираните гимназии с профил “Софтуерни и хардуерни науки”.



Резултатите от настоящата дипломна работа са вече **внедрени в практиката** и разработените учебни ресурси **се използват реално от стотици ИТ учители** в България в ежедневната им работа. Това е една от основните цели и тя вече е изпълнена, още преди защитата на настоящата дипломна работа.

Задачи, произтичащи от целите

Във връзка с поставената цел разработване на **учебни програми** и **качествено учебно съдържание** на български език за преподаване в “Модул 1. Обектно-ориентирано проектиране и програмиране” от профил “Софтуерни и хардуерни науки” в профилираните гимназии си поставяме следните **конкретни задачи** в рамките на настоящата дипломна работа:

- Разработка и описание на **концепция за учебни материали** за преподаване на програмиране за ученици в гимназиален етап. Структурата и форматът на учебните материали (презентации, упражнения с автоматизирано оценяване и напътствия, изпитни теми, тестове за ДЗИ и т.н.) се основават на **дългогодишния преподавателски опит** на доцент доктор **Светлин Наков** (авторът на настоящата дипломна работа), който преподава програмиране с огромен успех вече няколко десетилетия в университети, училища, школи и ИТ академии и е автор на десетки учебници, книги, видео уроци и учебни курсове по програмиране и софтуерни технологии.
- Изготвяне на **няколко варианта за учебни програми** за “Модул 1. Обектно-ориентирано проектиране и програмиране”, които съответстват на зададените от МОН **учебни програми** за профил “Софтуерни и хардуерни науки” с **различен хорариум**: 72 учебни часа (минимален базов вариант), 108 учебни часа (разширен вариант) и 144 учебни часа (вариант експерт).
- Изготвяне на **качествено учебно съдържание** на български език за преподаване по “Модул 1. Обектно-ориентирано проектиране и програмиране” в профил “Софтуерни и хардуерни науки”, в три варианта с различен хорариум (базов, разширен и експерт).
 - **Слайдове** за презентиране в клас + **живи примери** (live demos)
 - Задачи за **упражнения** с насоки за решаване и **автоматизирана проверка** (чрез online judge система)



- **Авторски решения на задачите** (сорс код)
- **Практически проекти** (задания + насоки и постъпкови ръководства)
- **Изпитни задания** (примерни изпитни задания за контролна и класна работа в помощ на учителите)
- **Тестове за ДЗИ** (държавен зрелостен изпит по информатика)
- Изготвяне на кратки **методически указания** и напътствания за учителите за целия учебен курс и за всяко учебно занятие.
- **Публикуване в Интернет** за общодостъпно публично ползване **на разработените учебни програми, учебно съдържание и материали** в полза на всички български учители, под **свободен лиценз** с отворен код.
- **Внедряване на резултатите в практиката:** използване на учебните програми, материали и ресурси в **реалната работа на фокус група от български ИТ учители**, които да дадат мнения, оценка и обратна връзка за полезността и качеството на разработените ресурси и препоръки за тяхното подобряване.



Глава 1. Проучване и планиране

Във връзка с поставената цел разработване на **учебни програми и учебно съдържание** за преподаване в модул “Обектно-ориентирано проектиране и програмиране” от профил “Софтуерни и хардуерни науки” в профилираните гимназии извършихме следните **проучвания и анализи** на проблемната област:

- Проучване на **образователната система** в българските училища и **обучението по програмиране и ИТ** в различните образователни етапи.
- Проучване на **законовата рамка и наредбите на МОН**, свързани с преподаването на програмиране и ИТ в училищата, учебните дисциплини и структура на **учебните планове** и програми в различните образователни етапи според наредбите на МОН.
- Анализ на текущото състояние на преподаването в **профил “Софтуерни и хардуерни науки”** в профилираните гимназии: учебен план, учители, учебници, нерешени проблеми.
- Анализ на текущото състояние на преподаването в модул “**Обектно-ориентирано проектиране и програмиране**” – учебна програма и учебни материали.
- Проучване на **проблемите на учителите** при преподаване на информатика и ИТ в училище и по-конкретно в профил “Софтуерни и хардуерни науки” и модул “Обектно-ориентирано проектиране и програмиране”.

Училищно образование в България

Училищното образование в България е разделено на **три етапа** (по дефинициите на Закона за предучилищното и училищното образование [\[ЗПУО, 2023\]](#)):

- **Начален етап** на образование: **1–4 клас**
- **Прогимназиален етап** на образование: **5–7 клас**
- **Гимназиален етап** на образование: **8–12 клас**
 - **Първи гимназиален етап: 8–10 клас**
 - **Втори гимназиален етап: 11–12 клас**



Училищата в България

Училищата в България са **неспециализирани** (общообразователни, профилирани, професионални) и **специализирани** (спортни, духовни, по култура и изкуства).

Според етапа или степента на образование **училищата в България** са:

- **Начални училища (1–4 клас)**
 - **Общообразователна насоченост**
 - **127** начални училища (НУ) за 2022/2023 по данни от [\[НСИ, 2023a\]](#)
- **Основни училища (1–7 клас)**
 - **Общообразователна насоченост**
 - **1128** основни училища (ОУ) за учебната 2022/2023 [\[НСИ, 2023a\]](#)
- **Гимназии (8–12 клас)**
 - **Профилирани и професионални** гимназии (ПГ) – езикови, математически, природоматематически, хуманитарни, за изкуства и други
 - **116** гимназии (ПГ) за учебната 2022/2023 по данни от [\[НСИ, 2023a\]](#)
- **Обединени училища (1–10 клас)**
 - **Общообразователни и професионални** обединени училища (ОУ), **75** обединени училища (ОУ) за 2022/2023 по данни от [\[НСИ, 2023a\]](#)
- **Средни училища (1–12 клас)**
 - **Общообразователни** в началния и прогимназиалния етап и с **профилирани и професионални** паралелки в гимназиален етап
 - **496** средни училища (СУ) за 2022/2023 по данни от [\[НСИ, 2023a\]](#)

Специализираните училища могат да са също начални, основни и средни:

- **Специализирани училища (1-12 клас, различно според училището)**
 - **Спортни, по култура и изкуства, духовни**
 - **47** специализирани училища за 2022/2023

Според **формата на обучение** училищата в България могат да бъдат :

- **Дневни** – в сутрешна или следобедна смяна
- **Вечерни** – за обучение на работещи, във вечерните часове

Според **собствеността** училищата в България могат да бъдат:

- **Държавни** – финансират се и се управляват от държавата



- **Общински** – управляват се и се финансират от местната община
- **Частни** – управляват се и се финансират от частна фирма
- **Духовни** – управляват се и се финансират от религиозна институция

Начални и основни училища

В **началните училища** (1-4 клас) се осигурява **базово образование**, фокусирано върху развитието на основните умения и познания в различни области (български език и литература, математика, история, природни науки, чужди езици, физическо възпитание и други). Обучението по **информационни технологии (ИТ)** започва още в **3 клас** с учебния предмет “**Компютърно моделиране**”.

В **основните училища** (1-7 клас) се осигурява основна подготовка по общо-образователните предмети (български език и литература, математика, история, природни науки, чужди езици, физическо възпитание и други), а **компютърно моделиране и информационни технологии** се изучават задължително **до 7 клас** във всички български училища.

Според статистика от НСИ от 2023 г. в начален и прогимназиален етап (**1-7 клас**) в България през учебната 2022/2023 г. учат около **422 000 ученици** [\[НСИ, 2023b\]](#).

Професионални и профилирани гимназии

Според съдържанието на подготовката гимназиите в България (8-12 клас) биват **профилирани и професионални** [\[ЗПУО, 2023\]](#).

Профилираните гимназии са училища (8-12 клас), които предлагат **специализирани профили** в различни области. Те са предназначени за ученици, които имат интерес в определена сфера и искат да получат **по-специализирано образование** в нея. Примери за профилирани гимназии са езиковите, математическите, и хуманитарните. В профилираните гимназии в гимназиален етап на обучение **ИТ се изучават от всички ученици** (с неголям хорариум), а при специализираните ИТ профили като “**Софтуерни и хардуерни науки**” броят часове по профилиращите предмети “**Информатика**” и “**Информационни технологии**” е значителен.

Според статистика от НСИ от 2023 г. в гимназиален етап (**8-12 клас**) в **профилирано обучение** в България през учебната 2022/2023 г. учат около **128 000 ученици** [\[НСИ, 2023b\]](#).



Професионалните гимназии са училища (8-12 клас), които акцентират върху предоставяне на **практически и професионални умения** на учениците и ги подготвят за определена **професия** съгласно изискванията на **ЗПОО** (Закон за професионалното образование и обучение). Примери за професионални гимназии са училищата по компютърни и дигитални науки, електротехника, икономика и бизнес, строителство, дървообработване, селско стопанство, туризъм, механика и транспорт, корабоплаване, приложни изкуства, текстил, фризьорство и други. В професионалните гимназии за **ИТ професиите** (например “Приложен програмист” и “Системен програмист”) обучението по **програмиране, софтуерни, компютърни и информационни технологии** са задълбочени и интензивни, с голям брой часове за специализирана подготовка и практически занимания и стажове.

Според статистика от НСИ от 2023 г. в гимназиален етап (**8-12 клас**) в **професионални паралелки** в България през учебната 2022/2023 г. учат около **159 000 ученици** [\[НСИ, 2023d\]](#).

Според статистиката от НСИ от 2023 г. в гимназиален етап (**8-12 клас**) сумарно в **профилирани и професионални паралелки** учат около **287 000 ученици**.

Обучение по програмиране и ИТ в училищата

Обучението по **програмиране и ИТ** в България започва още в основното училище (от 3 клас) и продължава до края на гимназиалния етап (12 клас).

Програмиране и ИТ в началното и основното образование

В **началния етап** на образование в България по учебната програма на МОН в 1 и 2 клас не се изучават предмети, свързани с ИТ. В някои училища във 2 клас се изучава факултативна дисциплина “Информационни технологии”. В **3 и 4 клас** се включва учебния предмет “**Компютърно моделиране**”, където всички български ученици изучават компютърна грамотност и **блоково програмиране** (със Scratch) [\[МОН, 2018e\]](#), [\[МОН, 2019\]](#). По данни на НСИ за учебната 2022/2023 г. в **3 и 4 клас** компютърно моделиране изучават около **116 000 ученици** [\[НСИ, 2023b\]](#).

В **прогимназиален етап** (**5–7 клас**) учениците задължително изучават учебен предмет “**Компютърно моделиране и информационни технологии (КМИТ)**”,



който обхваща мултимедия, използване на софтуерни продукти и програмиране (първоначално на **Scratch**, след това с езика **Python**) [МОН, 2020]. По данни на НСИ за учебната 2022/2023 г. в **5, 6 и 7 клас** на компютърно моделиране и информационни технологии се обучават **189 580 ученици** [НСИ, 2023b].

Програмиране и ИТ в гимназиите: профили и професии

В гимназиалния етап на образование в профилираните гимназии се изучава само **един профил** с интензивно изучаване на информатика и ИТ – профил “**Софтуерни и хардуерни науки**”. В профилите “**Предприемачески**” и “**Математически**” също може да има профилиращи предмети, свързани с програмиране и ИТ [МОН, 2016a], [МОН, 2016b].

Профил “Софтуерни и хардуерни науки” се изучава в **над 150 гимназии** от **около 20 000 ученици** [Наков, 2021].

В професионалните гимназии е наличен избор между **пет професии** с основен фокус върху програмиране и софтуерни технологии: “**Програмист**”, “**Приложен програмист**”, “**Системен програмист**”, “**Програмист на изкуствен интелект**” и “**Програмист на роботи**”. Допълнително програмиране и ИТ се изучават и в други професии, като “**Икономист-информатик**”, “**Организатор на Интернет приложения**”, “**Техник на компютърни системи**” и други.

Според данни на НСИ към 2023 г. в България има **351 професионални гимназии** [НСИ, 2023c], от които професии “**Приложен програмист**”, “**Системен програмист**”, “**Програмист на изкуствен интелект**” и “**Програмист на роботи**” се изучава в **над 70 гимназии** от **около 10 000 ученици** [Наков, 2021].

Колко ученици учат програмиране и ИТ в България?

Няма прецизно събрана и точна статистика за всички, които **изучават програмиране и ИТ в България към 2023 г.**, но можем да дадем следните **ориентировъчни данни**, събрани от вече цитираните по-горе източници:

Клас	Етап	Ученици	Какво изучават
3-4 клас	начален етап	116 000 ученици [НСИ, 2023b]	Компютърно моделиране
5-7 клас	прогимназиален етап	190 000 ученици [НСИ, 2023b]	Компютърно моделиране и информационни технологии



8-10 клас	I гимназиален етап	172 000 ученици [НСИ, 2023b] , [НСИ, 2023d]	Информационни технологии
11-12 клас	II гимназиален етап	30 000 ученици [Наков, 2021]	Програмиране, софтуерни технологии и ИТ
1-12 клас	всички етапи	общо 508 000 ученици	ИТ, компютърно моделиране и програмиране

Както можем да видим от таблицата, обучението по **компютърно моделиране, информационни технологии и програмиране** е **широко застъпено** в българското училищно образование, а това предполага сериозна нужда от специализирано учебно съдържание и подготвени учители.

Учебни планове по програмиране и ИТ в гимназиален етап

В гимназиален етап на образование (8-12 клас) профилираните и професионалните гимназии следват **рамковите учебни планове** на МОН. Те задават **рамки за разпределението на учебните часове** по раздели, предмети и часове за всеки клас в гимназиален етап на образование (8-12). **Рамковите учебни планове** са описани в следните нормативни документи от МОН:

- За **профилираните гимназии**: Наредба № 4 от 30.11.2015 г. за **учебния план** (рамкови учебни планове за общообразователна, профилирана и професионална подготовка) [\[МОН, 2015\]](#).
- За **професионалните гимназии**: Заповед № РД09-1808/17.03.2017 г. за **рамкови програми В** за професионално образование [\[МОН, 2017\]](#).

Всички български гимназии имат право да разкриват паралелки както в **профилирана подготовка**, така и в **професионална подготовка**. Например в профилирана природоматематическа гимназия (ПМГ) може да има паралелки с профил "Софтуерни и хардуерни науки" и същевременно паралелки с професия "Приложен програмист".

Задължителни, избираеми и факултативни дисциплини

Учебният план в гимназиален етап на образование (според наредбата за учебния план) се структурира в три раздела: **раздел "А"**, **раздел "Б"** и **раздел "В"**.



- **Раздел "А"** е общ за всички гимназии и включва всички **задължителни общообразователни дисциплини** – български език и литература, чужди езици, математика, природни науки, физическо възпитание и други. Предметите от раздел "А" и техните хорариуми са определени от МОН за всички български гимназии.
- **Раздел "Б"** предоставя **специализирано образование** в различни области според избрания **профил или професия**. Учебните предмети от раздел "Б" се избират от училището в училищния учебен план (и затова се наричат "избираеми"). След като веднъж са избрани и утвърдени от училищното ръководство, те стават **задължителни** за всички ученици. В профил **"Софтуерни и хардуерни науки"** в раздел "Б" задължително се изучават като профилиращи предмети **информатика** и **ИТ**, а в много училища се изучават и допълнително **математика** и **английски език**.
- **Раздел "В"** включва **допълнителни (факултативни) часове** по предложени от училищното ръководство предмети, които всеки ученик може да запише **по желание**. Това са изборни учебни предмети, наричани още "кръжоци". Те се провеждат най-често след часовете в раздел "А" и "Б".

Законова рамка и наредби за програмиране и ИТ в профилирана подготовка

В профилираната подготовка в българските гимназии се предлага само един профил, ориентиран към програмиране, софтуерни и информационни технологии: **профил "Софтуерни и хардуерни науки"**. В него се изучават като профилиращи предмети задължително **"Информатика"** и **"Информационни технологии"** [[МОН, 2015](#)].

Обучението по **информатика** и **ИТ** в профил **"Софтуерни и хардуерни науки"** следва няколко **нормативни документа**, публикувани от МОН:

- Наредба 4 / 2015 г. за учебния план [[МОН, 2015](#)]
- Наредба 7 / 2016 г. за профилираната подготовка [[МОН, 2016a](#)]
- Наредба за изменение и допълнение на наредба 7 / 2016 г. за профилираната подготовка – учебно-изпитни програми за ДЗИ [[МОН, 2016b](#)]



Според наредбата за профилираната подготовка в профил **“Софтуерни и хардуерни науки”** профилиращите предмети задължително са 3 или 4, от които първите два задължително са **“Информатика”** и **“Информационни технологии”**. Най-често гимназиите избират като допълнителни профилиращи предмети **“Математика”** и **“Английски език”**, но в някои гимназии може да бъдат избрани и други профилиращи предмети.

Законова рамка и наредби за програмиране и ИТ в професионална подготовка

В професионалната подготовка в българските гимназии се изучават няколко професии и специалности, свързани с програмиране и ИТ:

- 4810101 – Програμισ (заповед № РД 09-298/19.02.2009 г.)
- 4810201 – Системен програμισ (заповед № РД 09-298/19.02.2009 г.)
- 4810301 – Приложен програμισ (заповед № РД 09-1984/12.12.2016 г.)
- 4810401 – Програμισ на изкуствен интелект (заповед № РД 09-843/07.04.2021 г.)
- 4810501 – Програμισ на работи (заповед № РД 09-843/07.04.2021 г.)
- 4820101 – Икономическа информатика (заповед № РД 09-413/12.05.2003 г.)
- 4820401 – Електронна търговия (заповед № РД 09-298/19.02.2009 г.)

Професионалните гимназии реализират учебните си планове, следвайки **рамкови програми “Б”** за професионално обучение. Тези рамкови програми регламентират придобиване на втора или трета степен на професионална квалификация за ученици и лица, навършили 16 години в съответствие със:

- Закона за професионалното образование и обучение (ЗПОО)
- Закон за предучилищното и училищното образование (ЗПУО)
- Държавния образователен стандарт (ДОС) за професията

Държавният образователен стандарт (ДОС) в България е **нормативен документ**, който определя **задължителните** знания, умения и компетентности, които учениците трябва да придобият за да покрият изискванията за изучаваната от тях **професия** и специалност от списъка на професиите за професионално образование и обучение (СППОО).



Държавният образователен стандарт (ДОС) определя изискванията към кандидатите, **описанието на професията, единиците резултати от ученето**, изискванията към материалната база и изискванията към обучаващите. Стандартът за всяка професия включва **общата, отрасловата и специфичната професионална подготовка** с необходимите професионални компетентности, които позволяват на обучаемия да упражня професията след успешно завършване на обучението си.

Държавният образователен стандарт (ДОС) замества вече излезлите от употреба **Държавни общообразователни изисквания (ДОИ)**, които са използвани в миналото за описание на професиите от професионалното образование и обучение (ПОО).

Обучението по изброените по-горе професии е обвързано със следните **нормативни документи**, публикувани от МОН:

- Наредба №1 от 15 януари 2018 г. за придобиване на квалификация по професията “Приложен програмист” [\[МОН, 2018c\]](#)
- Наредба № 19 от 9 октомври 2018 г. за придобиване на квалификация по професията “Системен програмист” [\[МОН, 2018d\]](#)
- Наредба № 18 от 23 август 2019 г. за придобиване на квалификация по професията “Програмист” [\[МОН, 2019b\]](#)
- Протокол № 1/23.02.2022 г. – Държавен образователен стандарт (ДОС) за придобиване на квалификация по професия “Програмист на работи” [\[МОН, 2022\]](#)
- Протокол №4/29.09.2021 – Държавен образователен стандарт (ДОС) за придобиване на квалификация по професия “Програмист на изкуствен интелект” [\[МОН, 2021\]](#)

Профил “Софтуерни и хардуерни науки”

Профил “Софтуерни и хардуерни науки” се изучава в **профилираните гимназии** (главно в 11 и 12 клас) с два основни профилиращи предмета: “**Информатика**” и “**Информационни технологии (ИТ)**”.

Според наредбата за профилираната подготовка [\[МОН, 2016a\]](#) в профил “Софтуерни и хардуерни науки” **профилиращите предмети задължително са 3**



или 4, от които първите два задължително са **“Информатика”** и **“Информационни технологии”**. Най-често гимназиите избират като допълнителни профилиращи предмети **“Математика”** и **“Английски език”**, но в някои гимназии може да бъдат избрани и други профилиращи предмети, например **“Физика и астрономия”** или **“Предприемачество”**.

Според дългогодишния преподавателски и учителски опит на д-р Светлин Наков (авторът на настоящата дипломна работа) **най-важен и най-значим** за профила **“Софтуерни и хардуерни науки”** е предметът **“Информатика”**, защото в него се изграждат умения за логическо мислене, решаване на проблеми, програмиране и софтуерни технологии. Той е и **най-труден** и изисква най-много усилия, за да бъде усвоен и затова за него трябва да бъдат отделени **най-голям брой учебни часове**, особено в 8, 9 и 10 клас, тъй като програмирането (както и математиката) се учи с **натрупване** във времето и последователно изграждане и надграждане на умения едно над друго.

Вторият задължителен профилиращ предмет **“Информационни технологии”** е **значително по-лесен от програмирането** и се овладява по-бързо от учениците. На него може да се отделят **по-малко часове**.

Третият и евентуално четвъртият профилиращ предмет са **допълващи** и не са от ключово значение за развитие на уменията по програмиране, софтуерни и информационни технологии (които съставят ядрото на профил **“Софтуерни и хардуерни науки”**).

Рамков учебен план за профил “Софтуерни и хардуерни науки”

Рамковият учебен план за профил **“Софтуерни и хардуерни науки”** в най-често срещания си вариант (с интензивно изучаване на чужд език, според наредбата за профилираната подготовка [[МОН, 2016a](#)]) изглежда ето така:



Рамков учебен план
за профилирано образование с интензивно изучаване на чужд език с профил
"Математически", "Софтуерни и хардуерни науки", "Икономическо развитие" или
"Природни науки"

	Образователна степен: Средна							
	I гимназиален етап				II гимназиален етап			ОБЩО
	Класове			Общо	Класове		Общо	
	VIII	IX	X	VIII - X	XI	XII	XI - XII	
Учебни седмици	36	36	36		36	31		

Раздел А - задължителни учебни часове								
Учебни предмети								
Български език и литература	144	108	108	360	108	93	201	561
Чужд език - ...	648	144	72	864	72	62	134	998
Чужд език - ...		72	72	144	72	62	134	278
Математика	108	108	72	288	72	62	134	422
Информатика	72			72				72
Информационни технологии	36	36	18	90				90
История и цивилизации		144	126	270				270
География и икономика		72	90	162				162
Философия	36	36	72	144				144
Гражданско образование					36	31	67	67
Биология и здравно образование		90	72	162				162
Физика и астрономия		90	72	162				162
Химия и опазване на околната среда		90	72	162				162
Музика	18	18	18	54				54
Изобразително изкуство	18	18	18	54				54
Физическо възпитание и спорт	72	72	72	216	72	62	134	350
Общо за раздел А	1152	1098	954	3204	432	372	804	4008

Раздел Б - избираеми учебни часове								
Общ годишен брой часове за раздел Б		54	198	252	720	620	1340	1592
Общо за раздел А + раздел Б	1152	1152	1152	3456	1152	992	2144	5600



Раздел В - факултативни учебни часове								
Максимален годишен брой часове за раздел В	144	144	144	432	144	124	268	700
Общо за раздел А + раздел Б + раздел В	1296	1296	1296	3888	1296	1116	2412	6300

По плана на МОН специализираната подготовка по програмиране, софтуерни и информационни технологии се провежда най-вече в рамките на часовете от раздел "Б", предимно в 11 и 12 клас (когато за профилиращите предмети са отделени между 18 и 20 часа седмично).

Модули по профилиращ предмет "Информатика"

Профилираната подготовка по "Информатика" за профил "Софтуерни и хардуерни науки" включва следните четири задължителни модула [\[МОН, 2018a\]](#):

- **Модул 1: Обектно-ориентирано проектиране и програмиране** (72 часа, 11 клас) – програмиране, класове, обекти, принципи на ООП
- **Модул 2: Структури от данни и алгоритми** (72 часа, 11 клас) – структури от данни, масиви, линейни структури, текстообработка, работа с файлове, изключения, алгоритми, качество на кода
- **Модул 3: Релационен модел на бази от данни** (72 часа, 12 клас) – бази данни, релационен модел, таблици и връзки, SQL заявки, CRUD операции
- **Модул 4: Програмиране на информационни системи** (52 часа, 12 клас) – информационни системи, обектно-ориентиран модел на база данни, ORM технологии, графични (GUI) приложения с бази данни, CRUD операции, филтриране, навигация

Модули по профилиращ предмет "Информационни технологии"

Профилираната подготовка по "Информационни технологии (ИТ)" за профил "Софтуерни и хардуерни науки" включва следните четири задължителни модула [\[МОН, 2018b\]](#):

- **Модул 1: Обработка и анализ на данни** (72 часа, 11 клас) – анализ на данни с електронни таблици, бази данни, информационни системи
- **Модул 2: Мултимедия** (72 часа, 11 клас) – растерна и векторна графика, звук, видео, заснемане и обработка на видео, създаване на видеоклип



- **Модул 3: Уеб дизайн** (72 часа, 12 клас) – проектиране и разработка на уеб сайтове с HTML и CSS и CMS системи (WordPress)
- **Модул 4: Решаване на проблеми с ИКТ** (52 часа, 12 клас) – управление на проекти, екипна работа, ИКТ решения, инсталация, конфигурация и поддръжка на софтуер, правни аспекти

Защо е некоректно заглавието “Софтуерни и хардуерни науки”?

Според учебния план за профил “Софтуерни и хардуерни науки” на МОН няма **нико един** предмет или модул, в който да се изучава **хардуер**. Всички дисциплини, които се изучават, са изцяло фокусирани върху **програмиране и информационни технологии**. Поради тази причина заглавието на модула може да се приеме за **некоректно** или **подвеждащо**. За да има съответствие между заглавие и съдържание, е по-удачно профилът да се именува например “Програмиране и ИТ”, “Информатика и ИТ”, “Дигитални науки” или само “Софтуерни науки” (без да се поменава “хардуер”).

Програмиране се учи с много практика, години наред!

Според дългогодишния преподавателски опит на д-р Светлин Наков, измежду двата задължителни профилиращи предмета “Информатика” и “Информационни технологии”, **по-трудният и по-важният** за усвояване е “Информатика” и на нея трябва да се отделят повече учебни часове.

Заложените от МОН специфични цели на обучението по профилиращ предмет “Информатика” (описани в Приложение № 4 към чл. 4, ал. 1, т. 4 към Наредба 7 / 2016 г. за профилираната подготовка [\[МОН, 2016a\]](#)) акцентират най-вече върху **умението да програмираш**: да пишеш и разбираш програмен код, да изграждаш **логическо мислене**, да се научиш да решава **алгоритмични проблеми**, да овладееш инструментариума на обектно-ориентираното програмиране (ООП) и да добиеш практически умения за цялостно **разработване на софтуерни проекти**. Фундаментът за постигане на тези цели е **овладяване на програмирането и алгоритмичното мислене в дълбочина**.

Програмиране се учи с много практика, години наред, бавно, последователно, систематично, с надграждане и затова е от ключово значение в профил “Софтуерни и хардуерни науки” да бъдат предвидени **достатъчен брой часове по**



програмиране, равномерно разпределени от 8 до 12 клас! Това е потвърдено и от учителския опит на **д-р Светлин Наков** в Частна професионална гимназия по дигитални науки “**СофтУни Светлина**”. Изключително грешно и неработещо е изучаването на програмирането да започне чак в 11 клас.

Недостигът на часове по програмиране в профил “Софтуерни и хардуерни науки”

Броят часове по програмиране и ИТ в учебния план за профил “**Софтуерни и хардуерни науки**” се изчислява на база на предписанията от Наредба № 4 от 30.11.2015 г. за учебния план [\[МОН, 2015\]](#) и Наредба № 7 от 11.08.2016 г. за профилираната подготовка [\[МОН, 2016a\]](#). Тези наредби описват **колко** и **кои** могат да бъдат **профилиращите предмети** и **минималния и максималния брой часове** за тях съответно в 8, 9, 10, 11 и 12 клас.

Наредбите указват, че **профилиращите предмети** могат да са **3 или 4**, като 2 от тях **задължително** са “**Информатика**” и “**Информационни технологии**”. Часовете в **8, 9 и 10 клас** за профилираните предмети са посочени в наредбата за учебния план (съответно 0, 54 и 198 часа в 8, 9 и 10 клас), а часовете в **11 и 12 клас** са **най-малко 18** и **най-много 20** на седмица. Всеки профилиращ предмет в 11 и 12 клас се изучава най-малко в 4 и най-много в 8 учебни часа седмично.

Следвайки горните ограничения, можем да изчислим, че **минималният брой часове по информатика и ИТ** се получава при 4 профилиращи предмета и минималният допустим хорариум за информатика и ИТ.

Аналогично, **максималният брой часове по информатика и ИТ** се получава при 3 профилиращи предмета и максимален допустим хорариум за информатика и ИТ.

Минимален брой часове за информатика и ИТ

Минималният брой часове по информатика и ИТ в учебния план за профил “**Софтуерни и хардуерни науки**” се формира при максималния брой **4 профилиращи предмета** и когато часовете по профилиращите предмети в раздел Б за **11 клас** и **12 клас** са в минималния допустим размер от **18 часа седмично** (от които за **информатика** и за **ИТ** са заделени **по 4 часа седмично** – минималният допустим брой часове за профилиращ предмет).



Учебният план по-долу е базиран на рамковия учебен план за профила във вариант с **интензивно изучаване на чужд език**, в който има **допълнителни 72 часа Информатика в 8 клас** (за разлика от варианта с разширено изучаване на чужд език). Ето и разбивка на часовете по профилиращите предмети за 8-12 клас при минимален брой часове за Информатика и ИТ:

Учебни предмети	VIII	IX	X	XI	XII	Общо
Информатика (раздел А)	72					72
Информационни технологии (А)	36	36	18			90
Профилиращи предмети (раздел Б)		54	198	648	558	1458
Информатика (раздел Б)		54	108	144	124	430
Информационни технологии (раздел Б)			36	144	124	304
Трети профилиращ предмет (раздел Б)			36	180	155	371
Четвърти профилиращ предмет (раздел Б)			18	180	155	353
Информатика и ИТ (раздел А + Б)	108	90	162	288	248	896
Информатика (раздел А + Б)	72	54	108	144	124	502
Информационни технологии (раздел А + Б)	36	36	54	144	124	394
Общо профилиращи (раздел А + Б)	108	90	216	648	558	1620

Според **рамковия учебен план** на МОН за профил "Софтуерни и хардуерни науки" с **интензивно изучаване на чужд език** имаме:

- **Информатика и ИТ в 8 клас: 108 часа**
 - **72 часа Информатика в 8 клас:** два учебни срока (36 седмици) по 2 часа седмично. Тези часове идват от раздел А и **липсват**, ако се стъпи върху рамковия учебен план с **разширено изучаване** на чужд език (затова той не се препоръчва)



- **36 часа ИТ в 8 клас:** два учебни срока (36 седмици) по 1 час (от раздел А)
- **Информатика и ИТ в 9 клас: 90 часа**
 - **54 часа Информатика в 9 клас:** 18 седмици по 1 час + 18 седмици по 2 часа. Часовете в раздел Б са много малко и се препоръчва всичките 54 часа да се използват за **практическо програмиране** и решаване на задачи за развиване на алгоритмично мислене.
 - Препоръчително е по Информатика да се добавят поне **още 18 часа факултативни** (раздел В), за да се провеждат поне по 2 часа седмично през цялата година. И това е недостатъчно, но все пак ще поддържа нивото по програмиране, за да не се забрави наученото в 8 клас и да се надгражда над него.
 - Ако така или иначе **малкото часове от раздел Б се разделят между всички профилиращи предмети**, за програмирането ще останат твърде малко часове и това ще създаде **огромно препятствие** пред постигане на образователните цели по водещия профилиращ предмет "Информатика".
 - **36 часа ИТ в 9 клас:** два учебни срока (36 седмици) по 1 час (от раздел А)
- **Информатика и ИТ в 10 клас: 162 часа**
 - Тези часове идват от раздел Б (профилиращи предмети) и е най-удачно да се разделят на 108 часа за Информатика (защото програмирането е по-трудно, по-обемно и по-важно) + 36 часа ИТ + 36 часа за трети профилиращ предмет + 18 часа за четвърти профилиращ предмет.
 - **108 часа Информатика в 10 клас:** два учебни срока (36 седмици) по 3 часа седмично
 - **54 часа ИТ в 10 клас:** 18 седмици по 1 час + 18 седмици по 2 часа (общо, от раздел А и раздел Б)
- **Информатика и ИТ в 11 клас: 288 часа**
 - **144 часа Информатика в 11 клас:** Модул 1 ООП (72 часа, първи учебен срок, 18 седмици по 4 часа); Модул 2 Структури от данни (72 часа, втори учебен срок, 18 седмици по 4 часа)



- **144 часа ИТ в 11 клас:** Модул 1 Анализ на данни (72 часа, първи учебен срок, 18 седмици по 4 часа); Модул 2 Мултимедия (72 часа, втори учебен срок, 18 седмици по 4 часа)
- Това са **минималните задължителни часове** по профилиращите предмети Информатика и ИТ и те не могат да бъдат по-малко. Останалите часове от раздел Б могат да се ползват за трети и четвърти профилиращ предмет.
- **Информатика и ИТ в 12 клас: 248 часа**
 - **124 часа Информатика в 12 клас:** Модул 3 Базы данни (72 часа, първи учебен срок, 18 седмици по 4 часа); Модул 4 Информационни системи (52 часа, втори учебен срок, 13 седмици по 4 часа)
 - **124 часа ИТ в 12 клас:** Модул 3 Уеб дизайн (72 часа, първи учебен срок, 18 седмици по 4 часа); Модул 4 Решаване на проблеми с ИКТ (52 часа, втори учебен срок, 13 седмици по 4 часа)
 - Това са **минималните задължителни часове** по профилиращите предмети Информатика и ИТ и те не могат да бъдат по-малко. Останалите часове от раздел Б могат да се ползват за трети и четвърти профилиращ предмет.

Предвидените **учебни часове по Информатика** при минималния вариант **като цяло са недостатъчни** за усвояване на принципите на програмирането, алгоритмичното мислене и съвременните софтуерни технологии, така че се препоръчва училищен учебен план с **повече часове по Информатика** за сметка на третия и четвъртия профилиращ предмет (или само с три профилиращи предмета).

Повече часове по програмиране в 8, 9 и 10 клас

В много гимназии училищният учебен план се конструира така, че в **8, 9 и 10 клас Информатика почти не се изучава**. Това е **огромен проблем** и води до катастрофални резултати, защото **програмиране се учи последователно и с натрупване**. Ако не се започне навреме, още в 8 клас, то когато учениците стигнат 11 клас се оказва невъзможно да се усвои Модул 1 ООП за предвидените за него 72 учебни часа, тъй като той включва огромен обем учебен материал: цялото програмиране и принципите на ООП. Настоятелно препоръчваме на всички директори и училищни ръководства да предвидят **повече часове по програмиране в 8, 9 и 10 клас!**



Максимален брой часове за информатика и ИТ – препоръчителен вариант

Максималният брой часове по информатика и ИТ според учебния план за профил "Софтуерни и хардуерни науки" се формира, когато часовете по профилиращите предмети в раздел Б за 11 клас и 12 клас са в максималния допустим размер от 20 часа седмично, при минималния брой **3 профилиращи предмета** и при максималния брой **8 часа** седмично за "Информатика" и **8 часа** седмично за "ИТ".

Това е **препоръчителният вариант** да се изучава профил "Софтуерни и хардуерни науки", за да се усвоят на добро ниво основите на програмирането, софтуерните и информационните технологии, които съставят ядрото на този профил.

За **трети профилиращ предмет** се препоръчва да се избере "Математика", защото развива логическото мислене, а то е много помага за програмирането и софтуерната разработка. Допълнителни часове по чужд език не са необходими, тъй като той се учи изключително интензивно в 8 клас (с хорариум от 648 часа).

Препоръчителният учебният план за профил "Софтуерни и хардуерни науки" по-долу е базиран на рамковия учебен план за профила във вариант с **интензивно изучаване на чужд език**, в който има **допълнителни 72 часа Информатика в 8 клас** (за разлика от варианта с разширено изучаване на чужд език). Ето и разбивка на часовете по профилиращите предмети за 8-12 клас:

Учебни предмети	VIII	IX	X	XI	XII	Общо
Информатика (раздел А)	72					72
Информационни технологии (А)	36	36	18			90
Профилиращи предмети (раздел Б)		54	198	720	620	1592
Информатика (раздел Б)		54	108	288	248	698
Информационни технологии (раздел Б)			54	288	248	590
Трети профилиращ предмет (раздел Б)			36	144	124	304
Информатика и ИТ (раздел А + Б)	108	90	180	576	496	1450



Учебни предмети	VIII	IX	X	XI	XII	Общо
Информатика (раздел А + Б)	72	54	108	288	248	770
Информационни технологии (раздел А + Б)	36	36	54	288	248	662
Общо профилиращи (раздел А + Б)	108	90	216	720	620	1754

Защо е важен профил “Софтуерни и хардуерни науки”?

Според статистика от 2021 г. [\[Наков, 2021\]](#) профил “Софтуерни и хардуерни науки” се предлага в над 150 училища в България. Приблизително 4200 ученици избират да изучават този профил всяка година (общо около 21 000 ученици за петте години 8-12 клас). Тези статистики илюстрират колко актуален и значим е въпросът с качествено учебно съдържание по Информатика и ИТ за профил “Софтуерни и хардуерни науки”.

Модул “Обектно-ориентирано проектиране и програмиране”

Модул 1. “Обектно-ориентирано проектиране и програмиране (ООП)” е първият задължителен модул в профилираната подготовка по най-важния за профила учебен предмет “Информатика”. Модулът въвежда учениците в основите на програмирането (програмен код, среда за разработка, променливи и данни, условни конструкции, цикли, методи, графичен потребителски интерфейс, GUI приложения) и ги запознава с принципите на обектно-ориентираното програмиране и моделиране (ООП, класове, обекти, инстанциране, полета, конструктори, свойства, методи, статични членове, принципи на ООП, капсулация, абстракция, интерфейси, наследяване, полиморфизъм, връзки между класове, обектно-ориентирано моделиране, събития).

Официална учебна програма за Модул 1 “ООП”

Според официалната учебна програма на МОН за Модул 1 “ООП” в рамките на минимум 72 учебни часа са предвидени следните теми [\[МОН, 2018a\]](#):

1.1. Езици и среди за програмиране: обектно-ориентирани, функционални, ...



- 1.2. Обектно-ориентиран подход: класове, състояние, поведение
- 2.1. Интегрирана среда за обектно-ориентирано програмиране (IDE)
- 2.2. Основни елементи на език за ООП: клас, обект, инстанция, поле, метод
- 2.3. Обекти: създаване на обекти, конструктори, референция към обект
- 3.1. Член-променливи (полета и свойства) на клас: деклариране, достъп
- 3.2. Методи на клас: параметри, аргументи, връщан резултат, извикване
- 3.3. Стандартен метод на клас: ToString() метод
- 4.1. Средства на интегрирана среда за създаване на графичен интерфейс
- 4.2. Създаване на елементарен графичен интерфейс: GUI, хващане на събития
- 5.1. Аритметични и логически оператори, релации за сравнение, оператор за присвояване и конкатенация: данни, изрази и пресмятания с данни
- 5.2. Оценяване на логически изрази: логически оператори и изрази
- 6.1. Алгоритми: линеен, разклонен и цикличен изчислителен процес
- 6.2. Линейни изчислителни процеси: последователност от команди
- 6.3. Разклонени изчислителни процеси: условни конструкции за управление
- 6.4. Циклични изчислителни процеси: for-цикли, while-цикли, вложени цикли
- 7.1. Капсулиране на данни и методи: public, private, видове конструктори
- 7.2. Видимост на променливите: област на видимост, живот на променлива
- 7.3. Статични членове на клас: константи, изброими типове, статични данни
- 8.1. Проектиране на графичен интерфейс: форми, UI контроли, свойства
- 9.1. Връзки между класове: наследяване, асоциация, агрегация, композиция
- 9.2. Обектите като членове на класове: релации между класовете (А има В)
- 9.3. Модулна организация на класове: вложени, локални и анонимни класове
- 10.1. Наследяване на класове: базов клас, клас-наследник, класови йерархии
- 10.2. Капсулиране на данни при наследяване: достъп до базов клас, protected
- 10.3. Полиморфизъм: смяна на тип, upcasting, downcasting
- 10.4. Абстрактен клас: абстрактни и виртуални методи, подмяна (override)
- 10.5. Интерфейси: чист абстрактен клас, декларация, имплементация
- 11.1. Обработка на събития: обработка на събитие, обект на събитие

Целта на модула е доста амбициозна: да въведе учениците в **основите на програмирането** (аритметични и логически оператори, условни конструкции, цикли) и принципите на **обектно-ориентираното програмиране** (класове, обекти, капсулация, наследяване, полиморфизъм) само за 72 учебни часа.



Несъвършенства на учебната програма от МОН

Учебната програма на МОН за Модул 1 “ООП” има редица **несъвършенства**, които могат да бъдат изгладени от по-опитните учители:

- Подредбата на темите, включени в учебната програма на МОН, е **неправилна**. Според програмата учениците трябва **първо да изучават ООП**, да пишат класове, обекти, да спазват основните принципи на обектно-ориентираното програмиране, а **чак след това** да учат за основни аритметични и логически оператори, условни конструкции и цикли.
 - **Естествената подредба** при изучаването на посочените теми започва от писане на **проста програмна логика** и продължава след това към **класове и обекти** и накрая към **ООП принципите**.
- Предвидените 72 часа от МОН за цялото това учебно съдържание са **крайно недостатъчни**, ако учениците не са се занимавали преди това с програмиране.
 - **Не е възможно** в рамките на само 72 часа да се изучат на добро ниво **толкова много учебни теми** и концепции, ако ученикът започне да програмира за първи път от нулата.
 - За да се реши този въпрос, или трябва да се **увеличи броят часове**, или трябва да се **съкрати учебното съдържание**, или трябва да се разчита на **предишни знания и умения** (от 8, 9 и 10 клас).
 - Учебният план за профил “Софтуерни и хардуерни науки” позволява в 11 клас **до 288 учебни часа** за модулите от профилиращ предмет “Информатика”, така че има достатъчно възможности за повече часове за този модул.

Въпреки несъвършенствата, **учебната програма на МОН като цяло е разумно структурирана** и обхваща най-важното от основите на програмирането и ООП.

Недостигът на учебно съдържание по програмиране и ИТ

Всеки учител по програмиране и ИТ е разбира напълно, че **за да преподаваш качествено в клас**, е необходима **подготовка**, а качеството на преподаване пряко зависи от наличието на **старателно разработени учебни материали за работа в клас**: презентации, задания за упражнения, изпитни материали и други.



Добрата подготовка: ключ към качествено преподаване

По мнение на автора на настоящата дипломна работа д-р Светлин Наков съотношението **време за подготовка към време в клас** за един преподавател е средно **5:1** при по-лесни теми, и до 10:1, дори 15:1 при по-трудни теми, където се изисква по-голяма доза проучване и изучаване на непознати технологии и инструменти. Това означава, че ако нямаш разработени учебни материали, за да влезеш в клас и **да преподаваш 1.5 часа** (2 учебни часа), са необходими **7-8 часа подготовка** на урока и на учебните материали за него. След първи път, когато учителят вече разработени материали и преподава даден урок за втори или трети път, времето му за подготовка може да е минимално.

Без **добра подготовка** не можеш да преподаваш качествено! **Подготовката** за преподаване на програмиране и ИТ в гимназиален етап включва два компонента:

- **Изучаване на учебния материал на добро ниво**, тъй като софтуерните технологии бързо се променят и се изисква непрестанно обновяване на знанията и уменията на учителите, за да са в крак с новостите. Трябва да познаваш добре езика, технологията, техниката или инструмента, който ще преподаваш и то в най-съвременната му версия. В противен случай учиш заедно с учениците и качеството на преподаване пада много.
- **Подготовка на учебни материали** за работа в клас, материали за домашна работа, изпитни материали и проекти. **Учебните материали** могат да бъдат **старателно разработени** във вид на **презентации** с живи **примери**, задания за **упражнения** и практически **проекти**, изпитни **тестове** и **изпитни задания**, инструменти за проверка на знанията и други. С тяхна помощ учителят работи ефективно в клас и реализиран високо качество на преподаване.

Добрата подготовката за преподаване в клас включва и **двата компонента** и от тях зависи дали учителят ще преподава компетентно и добре организирано материята и практическите занимания по нея или ще излезе пред учениците неуверен и неподготвен, без добре структуриран урок и подготвени практически занимания към урока и без задълбочени знания по темата на урока.



От какво имат нужда учителите?

Според проведени редица неформални проучвания и последващо **формално допитване до 373 български ИТ учителя** от Фондация "Софтуерен университет" през 11/2022 г. [\[СофтУни фондация, 2022\]](#), се оказва, че по-важният от горните два компонента за българския ИТ учител са **учебните материали**. Създаването им е **трудоемко** и **времеотнемащо** и изисква специфични умения и опит и затова малцина учители успяват да подготвят качествени учебни материали за своите учебни занятия и преподавателска работа.

Учебните материали, когато са старателно разработени и с високо качество, служат и като **инструмент за самообучение** и самоподготовка по учебния материал, който даден учител трябва да преподава, така че те са отправната точка при подготовката за преподаване на програмиране и ИТ в училище.

Недостиг на учебни материали и знания

Традиционно българските ИТ учители изпитват **остър недостиг на учебни материали и знания по преподаваните учебни предмети**. Причините за това са много:

- **Учебният материал често се променя** и учителите не могат да ползват едни и същи материали години наред (както може примерно учителят по математика и по български език или история).
 - **Учителите трябва да учат обемен нов материал** в много кратки срокове, тъй като нови учебни предмети и промени изникват често.
 - **Програмиране и ИТ** са различни от математика, езици и история. В програмирането на всеки 2-3 години нещо фундаментално се променя и **съществуващите знания и умения трябва да се актуализират**.
- ИТ учителите са **претоварени** (с над 20 часа седмично) и **не им остава време** да създават учебно съдържание. В много училища, особено в по-малките населени места, има **недостиг на ИТ учители** и се налага съществуващите да поемат повече учебни часове от препоръчителното за добрата им работа.
- ИТ учителите нямат **уменията** да създават качествено учебно съдържание. За това се изискват определени компетенции и опит.



- ИТ учителите не се чувстват достатъчно **компетентни** по изучаваната материя, за да създават учебни материали за нея. Често пъти те са заставени **да преподават материал, с който нямат опит** и нямат време да го научат в дълбочина.
- Поради бързата смяна на технологиите с нови и нови през годините, **няма достатъчно натрупване на учебно съдържание през годините** (както е например по математика). Има учебници за някои ИТ предмети, а за други няма. Липсват актуални и добре разработени сборници със задачи и проекти, учебни помагала, методически указания и помощни инструменти за организиране на учебния процес и оценяването на знанията.

Недостигът на учебни материали е едно от най-големите страдания на българския ИТ учител [[СофтУни фондация, 2022](#)], защото заради бързата промяна на технологиите, няма достатъчно натрупвания на качествено съдържание, достъпно за учителската общност.

Учебници по програмиране и ИТ

Учебници по програмиране и ИТ има, но само за някои от учебните предмети и те не покриват цялостно нуждите на учителите. Някои от учебниците са добре разработени и актуални, други са остарели или не са на добро ниво.

- За профилираната подготовка по “Софтуерни и хардуерни науки” има **учебници** за всички задължителни модули по “Информатика” и “Информационни технологии”, но това изобщо не е достатъчно. Необходими са презентации за преподаване в клас, сборници със задачи за упражнения, изпитни теми, проекти и системи за оценяване.
- За **професионалната подготовка** (примерно за професия “Програмист на изкуствен интелект”) **учебници по повечето предмети липсват** (никога не са били създавани) и учителите се оправят както успеят.

Учебниците на заместват учебните материали

ИТ учителите споделят, че **наличието на учебници не замества острата нужда от добре разработени учебни материали** за обучение по даден учебен предмет [[СофтУни фондация, 2022](#)]. Наличните учебници по информатика и ИТ за ученици в началния, основния и гимназиалния етап на образование са предимно



теоретични и не достатъчно изчерпателни. Те са **добра отправна точка**, защото структурират уроците логически един след друг и дават представа какво трябва да съдържа всеки урок, но не са инструмент, с който учителят да работи в клас.

- Липсват **презентации и примери на живо**, които са подходящи за показване в клас на интерактивните дъски при преподаването на нов урок. Времето на писане по дъската мина и съвременните учители имат нужда от визуализация на учебния материал. Най-удачен формат за това са добре направените **презентации** (с малко текст, много примери, картинки и визуализации), придружени с **примери на живо** (писане на код на живо, решаване на задача на живо, създаване на ИТ проект на живо и подобни).
- Липсват и достатъчно **практически задачи**, чрез които учениците да затвърдят наученото в клас. Програмиране и ИТ са предимно **практика**, а за изграждането на умения са необходими **упражнения**, **практически казуси** и **проекти** от реалния живот. Това липсва в повечето учебници.
- Липсват **изпитни задания**, които да насочат учителя и учениците за препоръчителния форма на оценяването на натрупаните знания и умения.
- Липсват **системи за автоматично оценяване на код и проекти** (online judge systems) за предложените упражнения, които помагат много на учителите, особено когато работят с по-големи групи ученици.
- Липсват **задания и постъпкови ръководства** за изграждане на цялостни **ИТ и софтуерни проекти**. Програмиране и ИТ се учи с практика и **работа по проекти**, но за да се реализира добре известният проектно-базиран подход на обучение, са необходими отново старателно подготвени **учебни материали**, отвъд стандартните учебници.

Проект “Обучение за ИТ кариера”

Националната програма “Обучение за ИТ кариера” предлага факултативен формат за **обучение по програмиране** за гимназиални ученици от цялата страна (от всички профили и професии) за придобиване на професионална квалификация по професия “Приложен програмист”. Програмата е организирана от **МОН** с подкрепата на **БАСКОМ** (Българка асоциация на софтуерните компании) [[СофтУни фондация, 2017](#)]. Участието е **безплатно** за учениците. Учебните занятия се провеждат на живо в 5 регионални центъра:



- В ТУЕС и СофтУни – за София-град
- В ПГ по КТС, гр. Правец – за западна България
- В ОМГ “Акад. К. Попов”, гр. Пловдив – за Пловдив и централна България
- В ПГЕЕ Апостол Арnaudов, Русе – за региона североизточна България
- В ПГЕЕ “Константин Фотинов”, гр. Бургас – за югоизточна България

В рамките на програмата селектираните ученици изучават следните курсове по програмиране, софтуерна разработка и софтуерни технологии:

- Увод в програмирането
- Програмиране
- Увод в обектно-ориентираното програмиране (ООП)
- Увод в алгоритмите и структурите от данни
- Математически основи на програмирането
- Обектно-ориентирано програмиране
- Базис данни
- Разработка на софтуер
- Операционни системи
- Програмиране на вградени системи
- Конкурентно програмиране
- Интернет програмиране
- Софтуерно инженерство
- Функционално програмиране
- Алгоритми и структури от данни

Националната програма “Обучение за ИТ кариера” стартира за първи път през учебната 2017/2018 г. с обучения по първите няколко курса по програмиране. За успешното ѝ провеждане професионалисти от екипите на **БАСКОМ** и **БАИТ**, заедно с **д-р Светлин Наков** и **Ангел Георгиев** от Фондация “Софтуерен университет” разработват **учебно съдържание** за гореизброените курсове, като подготвят презентации, упражнения и изпитни материали. Тези курсове обаче **не се припокриват** напълно с модулите от профил “Софтуерни и хардуерни науки”.

Разработеното учебно съдържание за обученията в програмата “Обучение за ИТ кариера” е с **различно качество в различните курсове**. В началните курсове съдържанието е подробно, с детайлно разписани упражнения и автоматизирана



система за оценяване (online judge system). В някои от следващите курсове материалите не са подробни и не са добре форматирани, а упражненията не са детайлно разписани в разбираем за учениците и учителите вид. Като цяло качеството на материалите за някои от курсовете е спорно.

В програмата “Обучение за ИТ кариера” участват само **селектирани** с приеман изпит ученици от профилирани и професионални гимназии, а Министерството на образованието **не споделя публично в Интернет** създадените от екипа ресурси. Те се публикуват в портал с ограничен достъп, въпреки отворения им лиценз.

Проект “Отворено учебно съдържание” на СофтУни Фондацията

През октомври 2020 г. фондация “Софтуерен университет” с подкрепата на **БАСКОМ** стартира проект “Отворено учебно съдържание по програмиране и ИТ за българските училища” (BG-IT-Edu) с цел разработка на качествени учебни ресурси за обучение по информатика и ИТ в училище. Създадените от СофтУни и съмишленици учебни ресурси за обучение по програмата “Обучение за ИТ кариера” са подредени, осъвременени, надградени и разширени и са публикувани публично под **свободен лиценз** в GitHub [\[BG-IT-Edu, 2020\]](https://github.com/BG-IT-Edu/School-Programming):

- <https://github.com/BG-IT-Edu/School-Programming>

През 2022 година екипът, ръководен от **д-р Светлин Наков**, се разраства и в изготвянето на ресурси се включват **преподаватели и специалисти по програмиране** като **Александрина Механджийска** (софтуерен инженер от Благоевград), **Ангел Георгиев** (ИТ преподавател от София), **Петър Петров** (ИТ учител от Бургас), **д-р Мухарем Моллов** (ИТ учител от региона на Пловдив), **Стефан Куюмджиев** (ИТ специалист от Велико Търново), **Димитър Манолов** (ИТ учител от Лом) и други учители и съмишленици [\[СофтУни фондация, 2022a\]](#).

В периода 11/2022 – 06/2023 екипът на Фондацията работи усърдно и задълбочено по изготвяне на **качествено учебно съдържание** за обучение в **Модул 1 “ООП”** и **Модул 2 “Структури от данни и алгоритми”** за гимназиален профил “Софтуерни и хардуерни науки”. За двата модула са изготвени **учебни материали с много високо качество**, които включват: **учебни програми** в няколко варианта (базов, разширен и експерт), учебни **презентации** за всеки урок, задачи за **упражнения** в клас с насоки и напътствия за учениците, система за



автоматизирано оценяване на решенията в помощ на учениците (online judge system), **изпитни теми** за учителите, **въпроси за ДЗИ** с отговори и обяснения, **решения** на всички задачи, въпроси и изпити (за учителите), задания за **цялостни практически проекти** с постъпкови ръководства за разработката им.

Екипът на Фондацията **продължава работа** по изграждане на качествено учебно съдържание за обучение в **следващите модули по Информатика** (бази данни и информационни системи) за гимназиален профил “Софтуерни и хардуерни науки”, както и за **Компютърно моделиране и ИТ за 5 и 6 клас**.

Недостиг на учебни материали за профил “Софтуерни и хардуерни науки”

При първоначалното въвеждане на профил “Софтуерни и хардуерни наука” в българското гимназиално образование, с **първия випуск** ученици приети за обучение през учебната 2017/2018 г. се появява **несигурност сред ИТ учителите**: какво да преподават, по какви учебни планове и програми, по какви учебници, с какви учебни материали, на кой език за програмиране, с какви инструменти и други. Сред **директорите** новият по това време ИТ профил е непознат и те нямат опит как да структурират учебния план. Има остър недостиг подготвени учители с опит с новия профил, учебни ресурси, знания и умения по учебния материал.

В помощ на учителите в този несигурен за тях момент СофтУни организира през септември 2021 г. **обучителен курс за подготовка на учители** [\[СофтУни, 2021\]](#) за преподаване на четирите задължителни модула по “Информатика” от профил “Софтуерни и хардуерни науки”. Целта на курса е да предаде **знания и умения** по съдържанието на четирите модула, както и да предостави на учителите **учебни ресурси за преподаване в клас** със свободен лиценз за:

- Модул 1: [Обектно-ориентирано програмиране](#) (72 часа, 11 клас)
- Модул 2: [Структури от данни и алгоритми](#) (72 часа, 11 клас)
- Модул 3: [Релационен модел на бази от данни](#) (72 часа, 12 клас)
- Модул 4: [Програмиране на информационни системи](#) (52 часа, 12 клас)

Учебните ресурси за четирите модула са качени в GitHub в хранилището на Фондацията за публичен свободен достъп: <https://github.com/BG-IT-Edu/School->



[Programming/tree/main/Courses/Software-Sciences](https://programming/tree/main/Courses/Software-Sciences). Проектът е доста успешен и указва сериозна подкрепа на ИТ учителите, но не решава всички казуси, защото:

- Предоставените в този курс учебни материали **не следват точно учебните програми на МОН**. Те са извадка от съществуващи курсове в СофтУни и не са разработвани специално за нуждите на ИТ учителите.
 - **Не всички учебни теми** от програмите на МОН са обхванати.
- Курсът за Модул 4 “**Информационни системи**” от СофтУни е базиран на уеб технологиите и **ASP.NET MVC**, а по концепцията на МОН трябва да бъде базиран на технология за разработка на **desktop GUI приложения**.
- Учебните материали са на **английски език** и това е сериозен проблем за много учители и ученици.

Недостиг на учебно съдържание за Модул 1 “ООП”

Въпреки, че свободните учебни материали за Модул 1 “ООП”, които СофтУни Фондацията публикува през 2021 г. са добра основа за преподаване на “**Модул 1. Обектно-ориентирано проектиране и програмиране**” в профил “Софтуерни и хардуерни науки” в профилираните гимназии, тези материали не отговарят напълно на нуждите на учителите, защото:

- не следват точно учебната програма на МОН
- предоставени са на **английски език**
- няма заложена гъвкавост за преподаване с различен хорариум
- не съдържат достатъчно много задачи за упражнения

Главната цел на разработените **нови учебни материали** за обучение в Модул 1 “ООП” в периода 11/2022 – 06/2023 от СофтУни Фондацията под ръководството на д-р Светлин Наков и описани детайлно в настоящата дипломна работа, е да поправи описаните по-горе проблеми и да предостави за бесплатно ползване **висококачествено учебно съдържание за преподаване на Модул 1 “ООП”**, което следва указанията на МОН, но в същото време е универсално и **може да се адаптира за нуждите на всички български ИТ учители**, които преподават Информатика в училище.



Какво е качествено учебно съдържание?

Учебното съдържание включва **материалите** и **знанията**, които се предават и изучават в рамките на образователния процес, **систематизирани** във вид на старателно разработени и подредени учебни ресурси за учители и ученици.

Качественото учебно съдържание е подробно, добре систематизирано и с **практическа насоченост** (т.нар. hands-on learning). То се характеризира с **разнообразие** от учебни ресурси (презентации, задачи, проекти, изпити и т.н.), които осигуряват изчерпателни обяснения на ключови за съответната дисциплина концепции, съчетани с много практика и интерактивни занимания.

Според дългогодишния учителски опит и стандартите за качество на д-р Светлин Наков и СофтУни Фондацията **качественото учебно съдържание** за преподаване по програмиране и ИТ се състои от следните активи:

- **Учебна програма**, която описва обхванатите в учебния курс теми (задължителни и допълнителни)
- **Слайдове** за презентиране в клас + **живи примери** (live demos) за демонстриране на живо на новия материал (за учителите и учениците)
- Задачи за **упражнения** (за учениците) с насоки за решаване и **автоматизирана проверка** (чрез online judge система)
- **Практически проекти** (задания + насоки и постъпкови ръководства за имплементация, за учениците)
- **Изпитни задания** (примерни изпитни задания за контролна и класна работа в помощ на учителите)
- **Авторски решения на задачите** от упражненията, изпитите и проектите (сорс код, за учителите)
- **Тестове за ДЗИ** (Държавен зрелостен изпит по информатика, за тренировка за изпита, за ученици и учители)
- **Кратки методически указания** и напътствия за учителите за целия учебен курс и за всяко учебно занятие.

Според проведеното от Фондация “Софтуерен университет” през 11/2022 г. **допитване до стотици български ИТ учители** [\[СофтУни фондация, 2022\]](#), техните



разбирания за качествено учебно съдържание са много близки до описаните по-горе активи, от които трябва да се състои един добре подготвен учебен курс.

Учителите споделят, че **техните нужди от учебни материали се покриват на 100%** от учебните материали, разработени от СофтУни Фондацията, във формата описан по-горе. Учителите имат нужда за всеки учебен курс, който преподават, да се налични старателно разработени описаните по-горе активи за всяка учебна тема. Това им спестява време за подготовка и **вдига значително качеството** на учебния процес и резултатите на учениците.

Защо е необходимо качествено учебно съдържание?

Качественото учебно съдържание е от ключово значение за учителите по информатика и ИТ. Наличието на **систематизирани ресурси**, изготвени от професионалисти в сферата, **улеснява** и **ускорява** процеса на подготовка на преподавателите и на техните ученици, като предоставя възможност и за **практическо приложение** на изучаваните концепции. Качественото учебно съдържание дава следните **предимства**:

- **Спестява време** на учителите: вместо учителите да си подготвят сами учебни ресурси, те ползват готови учебни материали, с **по-високо качество, по-добре структурирани** и **по-актуални** спрямо съвременния свят, за който подготвят учениците.
- **Служи като обучение за учителите**: когато имаме детайлно разписано учебно съдържание, с внимателно подготвени примери, със задачи за упражнения и проекти и решения към тях, учителят много бързо минава през тях и възприема новата материя, която трябва да преподава и бързо и качествено се подготвя да влезе в клас.
- **Отговаря на въпроса “какво да преподавам”**: качественото учебно съдържание дава конкретика – какво точно да се включи във всяка една учебна тема, заложена в учебните планове и програми, какви примери да се дадат във всяка тема, какви упражнения да се дадат на учениците.
- **Отговаря на въпроса “как да преподавам”**: какво да включи в теорията, какви упражнения да дам съответно на по-бързо и на по-бавно напредващите ученици, как да си подготвя изпитите, какви проекти да дам



за практическа работа, кои теми са задължителни и кои мога да пропусна, ако не ми достига времето по някаква причина и т.н.

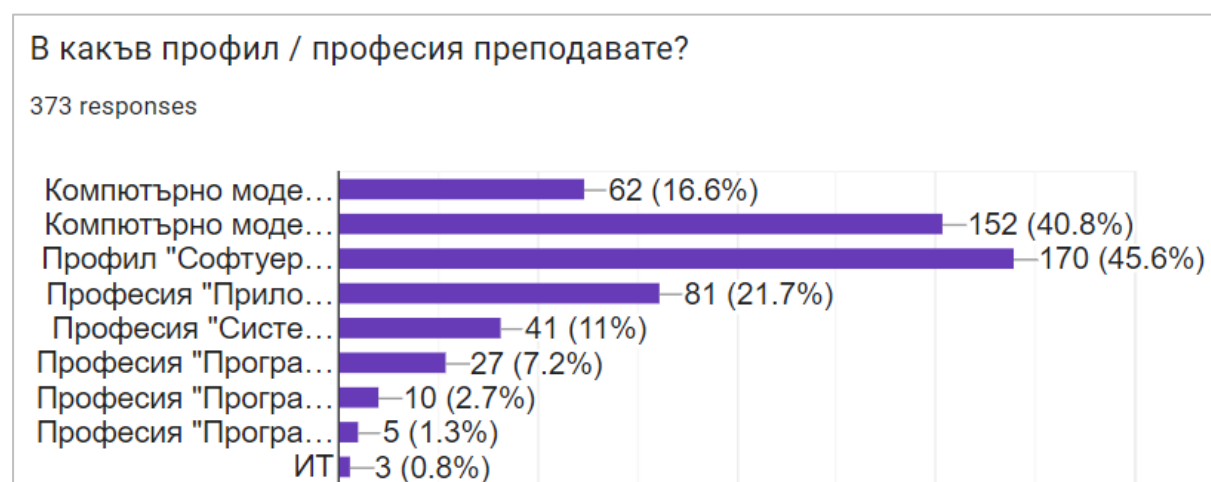
- **Прави обучението "по-интересно"**: когато учебното съдържание е съвременно, ориентирано към практиката, с много примери от реалния живот, с упражнения и проекти близки до живота на учениците, то е по-интересно за учениците и шанса да се запалят да учат е по-висок.

Качественото учебно съдържание в комбинация с **мотивирани и добре подготвени учители** е ключът към качественото училищно образование.

Проучване на нуждите на ИТ учителите (ноември 2022)

През ноември 2022 г. д-р Светлин Наков от името на фондация "Софтуерен Университет", проведе мащабно проучване сред учителите по информатика и ИТ в България [[СофтУни фондация, 2022](#)]. Главен фокус на анализ бяха **проблемите**, с които се сблъскват учителите в работния процес, както и **ресурсите, от които се нуждаят**, за да оптимизират работата си.

В проучването участваха **373 ИТ учителя** от цяла България: от малки населени места и големи градове, от малки и големи училища, с най-разнообразен учителски опит, от различен образователен етап и с различна специализация:

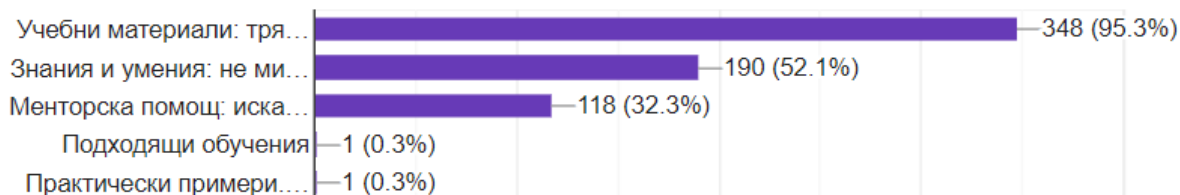


На база резултатите от проучването установихме, че **95.3%** от анкетираните преподаватели смятат, че **наличните към момента учебни материали по програмиране и ИТ са недостатъчни** и има нужда от разработка на **нови** такива:



От какво имате нужда?

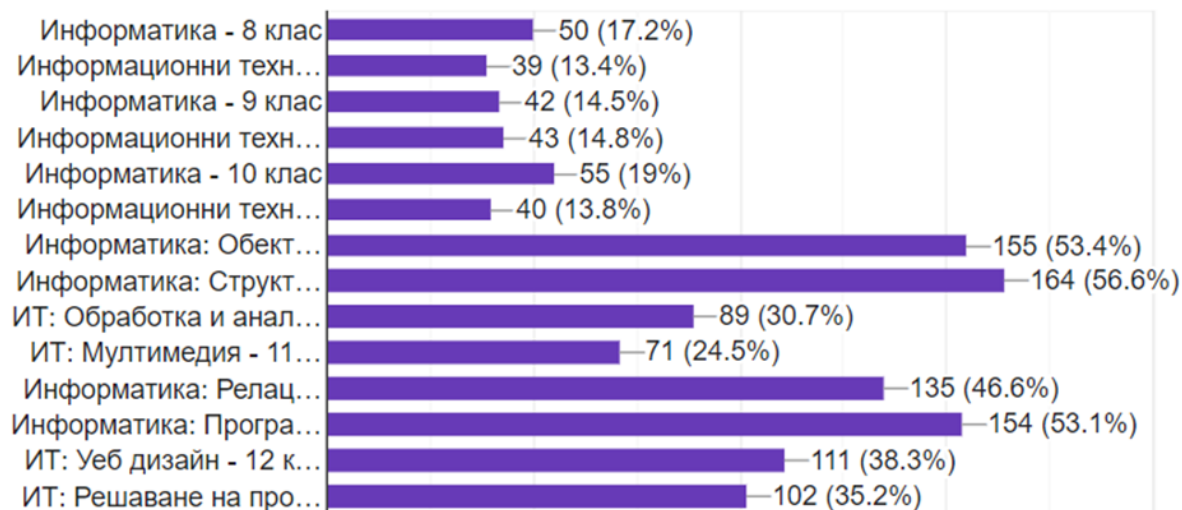
365 responses



След анализ на учебните предмети и модули, които учителите считат за най-проблемни, достигнахме до извода, че най-остра нужда от качествено учебно съдържание към 2022 г. има за модулите от профил **"Софтуерни и хардуерни науки"**: "Модул 1: ООП", "Модул 2: Структури от данни и алгоритми", "Модул 3: Релационни бази данни" и "Модул 4: Програмиране на информационни системи":

За кои учебни предмети имате най-голяма нужда от учебни материали? (профил "софтуерни и хардуерни науки")

290 responses

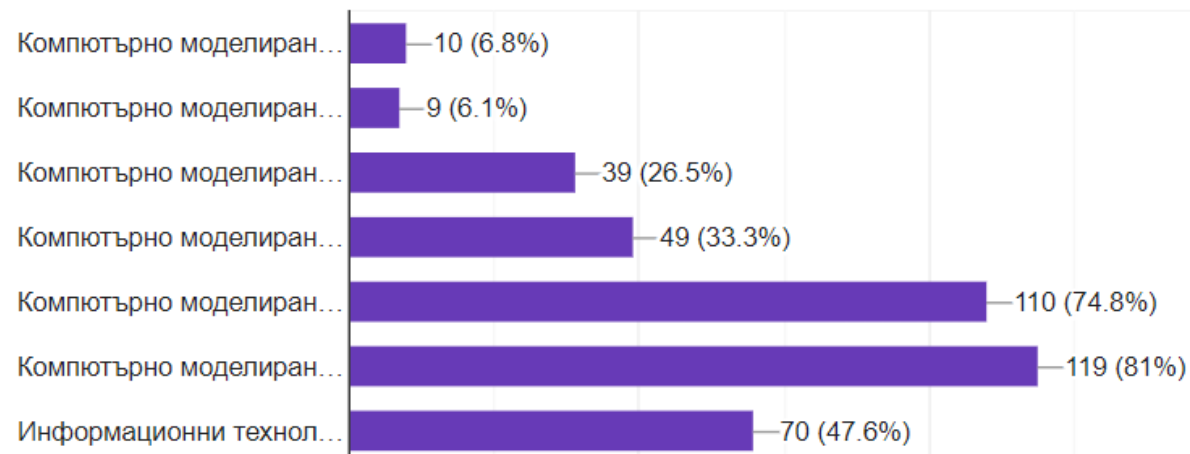


На второ място по нужда от учебни материали се оказаха **учителите от 5 и 6 клас**, които преподават **Компютърно моделиране и Информационни технологии**:



За кои учебни предмети имате най-голяма нужда от учебни материали? (начално и основно образование)

147 responses

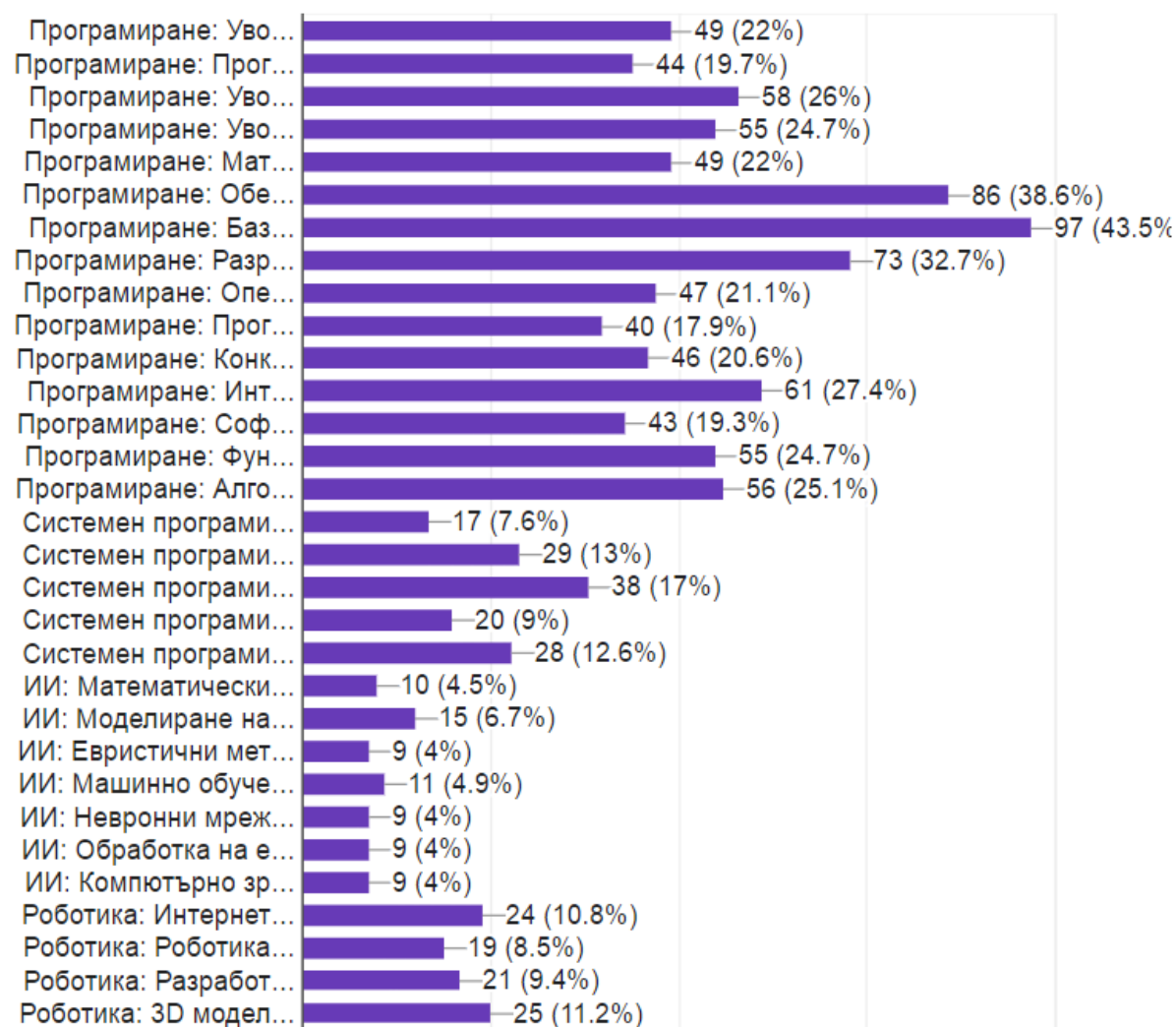


На трето място се оказаха **ИТ учителите от професионалните гимназии**, които преподават професиите "Приложен програмист" и "Системен програмист":



За кои учебни предмети имате най-голяма нужда от учебни материали?
(професионални паралелки: програмист, приложен програмист, системен програмист, програмист на ИИ, програмист на роботи)

223 responses



На базата на извършеното проучване можем да направим заключение, че **недостигът на качествено учебно съдържание за преподаване на програмиране и ИТ в училище е сериозен и повсеместен**. Разбира се, учебни материали има най-различни, но повечето са набързо събрани от някои и качеството им не е достатъчно добро, според нуждите на учителите. Те най-често споделят, че **имат нужда от по-съвременно, по-добре структурирано съдържание, в по-удобен за ползване формат** (презентации + примери + упражнения), имат нужда от много и разнообразни **упражнения** за различно напредналите ученици и **практически проекти**, с които да разпалят желанието на учениците да програмират.



Глава 2. Разработка на учебното съдържание

Основната цел на настоящата дипломна работа е **разработка и публикуване** за свободно ползване на качествено **учебно съдържание** за преподаване в първия и най-важен модул от гимназиален профил "Софтуерни и хардуерни науки", а именно **"Модул 1. Обектно-ориентирано проектиране и програмиране (ООП)"**.

За да реализираме тази цел, вече дефинирахме какво е нашето **разбиране** за **"качествено учебно съдържание"**. Сега следва да дефинираме **учебните програми** и **структурата на учебното съдържание**, така че да отговаря на нуждите на учителите и на наредбите на МОН, след което да разработим и самото учебно съдържание за всяка от темите от учебната програма.

Учебна програма

Предложената от нас **учебна програма** за обучение в Модул 1 "ООП" е разработена в **три варианта** (нива): **базов, разширен и експертен**, предназначени за различните групи според нивото на напредък на учениците и според броя на заделените за модула учебни часове.

Модул 1. "Обектно-ориентирано проектиране и програмиране" от профилиращ предмет "Информатика" е предвиден за преподаване в рамките на минимум **72 учебни часа** в **XI клас** за профил "Софтуерни и хардуерни науки". По мнението на учители, с които разговаряхме, е невъзможно само за **72 учебни часа** да се изучат на добро ниво всички учебни теми и концепции, които са включени в програмата на МОН за този модул. За да се реши този въпрос или трябва да се **увеличи броят часове**, или трябва да се **съкрати учебното съдържание**, или учениците трябва да дойдат с **предварителна подготовка** по програмиране, придобита още в 8, 9 и 10 клас. Най-добрият вариант е да се предвидят в учебния план **повече от 72 часа**, за да има време за усвояване на материята и за повече упражнения.

Тъй като предвидените **72 часа** за планираните от МОН учебни теми от Модул 1 "ООП" са **недостатъчни** за ученици без предишна подготовка по програмиране, предлагаме няколко варианта на изучаване на модула, съответно с **хорариум 72 часа** (базов), **108 часа** (разширен) и **144 часа** (експерт).



Базов, разширен и експертен вариант на учебната програма

Ето как изглеждат разработените **три варианта на учебна програма** за курса по ООП, с различен брой учебни теми, според броя часове:

Вариант	Описание	Необходими часове
Базов (теми без звездичка)	Обхваща най-важните теми от курса като покрива изискванията на МОН на съвсем базово ниво	72 часа (4 часа * 18 седмици), I учебен срок
Разширен (теми със звездичка)	Покрива напълно изискванията на МОН	108 часа (6 часа * 18 седмици), I учебен срок
Експерт (теми с 2 звездички)	Покрива изискванията на МОН + по-сложни теми за напреднали	144 часа (8 часа * 18 седмици), I учебен срок

Целта на трите варианта на учебна програма е да се даде гъвкавост на учителите и училищните ръководства да изучават Модул 1 "ООП" в по-малко или в повече часове, като съответно включват допълнителни теми, повече упражнения и проекти. Всяка от темите от учебното съдържание е отбелязана **без звездичка** (задължителна тема), с **една звездичка** (разширена тема) или с **две звездички** (тема за експерти) според варианта на учебната програма.

Според препоръките на СофтУни Фондацията и д-р Светлин Наков Модул 1 "ООП" трябва да се изучава с **повече от минималните 72 часа** (например 108 или 144 часа), за да може да се проведат повече упражнения и да се обхванат по-задълбочено планираните теми и някои незадължителни теми.

Друг вариант е в **8, 9 и 10 клас** да се усвоят **основите на програмирането** (работа с данни, програмна логика, условни конструкции, цикли, масиви и списъци), а в **11 клас** в Модул 1 "ООП" да се изучава **обектно-ориентирано програмиране** (ООП). В сегашния си вид модулът предполага усвояване и на двете: основи на програмирането и след тях основи на ООП. За двете няма как да стигне времето и това е огромен проблем, докладван от много ИТ учители, които преподават в профил "Софтуерни и хардуерни науки".



Разпределения на часовете в учебната програма

Предлагаме следните **три варианта** на учебна програма за Модул 1 "ООП". Те включват разпределение на учебни теми и посочват препоръчителен брой часове за теория и упражнения в трите предложени варианта:

Тема	Наименование на урока	Часове теория	Часове практика	Часове общо
01	Introduction to Programming	2	2	4
02	GUI Apps with Windows Forms	2	2	4
03.1	Conditional Statements - Basics	2	4	6
03.2	Conditional Statements - Advanced *	1	2	3
03.3	Conditionals - More Exercises **	0	3	3
04.1	Loops - Basics	2	4	6
04.2	Loops - Advanced *	1	2	3
04.3	Loops - More Exercises **	0	3	3
05.1	Methods - Basics	2	2	4
05.2	Methods - Advanced *	2	3	5
06	Mid Exam: Coding Basics	0	2	2
07.1	Classes & Objects - Basics	2	4	6
07.2	Classes, Objects, Lists - Advanced *	1	2	3
07.3	Classes & Objects - More Exercises **	0	3	3
08.1	Encapsulation - Basic	2	2	4
08.2	Encapsulation - Advanced *	1	1	2
09.1	Inheritance - Basic	2	2	4
09.2	Inheritance - Advanced *	1	1	2
10.1	Abstract Classes and Interfaces - Basic	2	2	4



10.2	Interfaces - Advanced *	0	2	2
11.1	Polymorphism - Basic	2	2	4
11.2	Polymorphism - Advanced *	0	2	2
12.1	Delegates and Events - Basic	1	1	2
12.2	Delegates and Events - Advanced *	2	2	4
13.1	Final Exam: Object-Oriented Programming	0	2	2
13.2	Final Quiz: Programming and OOP *	0	1	1
14	Design Patterns **	2	2	4
15	SOLID Principles **	2	2	4
16.1	Practical Projects: Rock-Paper-Scissors	0	4	4
16.2	Practical Projects: Guess a Number	0	4	4
16.3	Practical Projects: Quick Draw *	0	4	4
16.4	Practical Projects: Random Sentences **	0	6	6
	Задължителни часове (базов вариант)	21	39	60
	Резерв часове (базов вариант)	5	7	12
	Общ брой часове (базов вариант)	26	46	72
	Задължителни часове (разширен вариант)	30	61	91
	Резерв часове (разширен вариант)	6	11	17
	Общ брой часове (разширен вариант)	36	72	108
	Задължителни часове (вариант експерт)	34	80	114
	Резерв часове (вариант експерт)	13	17	30
	Общ брой часове (вариант експерт)	47	97	144



В таблицата по-горе всички теми **без звездичка** формират **базовия вариант** на програмата (за 72 учебни часа). Темите **без звездичка**, заедно с темите с **една звездичка** формират **разширения вариант** на програмата (за 108 учебни часа). Всички теми заедно (без, с една и с две звездички) формират експертния вариант на учебната програма (за 144 учебни часа).

Разпределенията в трите варианта са внимателно съобразени, така че:

- Броят часове в **базовата програма** да е **кратен на 2**, с предвиждането модулът да се изучава 18 седмици по 2 пъти седмично по 2 часа.
- Броят часове в **разширената програма** да е **кратен на 3**, с предвиждането модулът да се изучава 18 седмици по 2 пъти седмично по 3 часа.
- Броят часове в **разширената програма** да е **кратен на 4**, с предвиждането модулът да се изучава 18 седмици по 2 пъти седмично по 4 часа или 4 пъти седмично по 2 часа.

В учебната програма е предвиден стандартен **резерв в размер на 15-20% от часовете** за празници, извънредни ваканции, затвърждаване на знания, допълнителни изпитвания, работа по допълнителни проекти и други.

Учебната програма е **съобразена с изискванията на МОН** и съответства на официалната учебна програма за Модул 1 “ООП” [[МОН, 2018a](#)], но подрежда темите по по-логичен начин, така че първо да се изучават основите на програмирането, след това работата с класове и обекти и накрая принципите на ООП.

Разпределенията по-горе са примерни и служат като отправна точка за учителите. Всеки учител е свободен да прави промени според своето разпределение на часовете в рамките на седмицата, според нивото и темпото на класа, според непредвидени ситуации в учебния график, както и да работи с различна скорост с по-напредналите и с по-бавните ученици.

Структура на учебните ресурси

За всеки вариант на учебната програма сме разработили следните **учебни ресурси**, които смятаме, че са необходими и полезни на ИТ учителите:

- **Слайдове** за презентиране в клас + **живи примери** (live demos) – за учителя



- Задачи за **упражнения** с насоки за решаване и **автоматизирана проверка** (чрез online judge система) – за учителя и за учениците
 - Детайлно разписани условия на задачи, заедно с **насоки за решаване** на **първите** няколко задачи – за учениците
 - Задачите са **подредени по трудност**: от най-лесната към най-трудната. По този начин най-бавно напредващите ученици решават едва първите няколко задачи, а най-бързо напредващите решават всички задачи. Целта е **да обхванем всички ученици** и никой да не се откаже още в началото, защото не е успял да реши някоя задача.
 - Първите няколко задачи са разписани в големи детайли, за следващите са дадени подсказки и насоки, а за последните няколко задачи са дадени само условия, примерен вход и изход. Целта е трудността да се **покачва плавно**, заедно с плавното изграждане на практическите умения на учениците
- **Изпитни задания**: примерни изпитни задания за контролна и класна работа в помощ на учителите (заедно с авторски решения). Целта е да насочим учителя към **формата и съдържанието на изпитите**, които той може да организира, за да оцени знанията и уменията на учениците. За съжаление няма как предложените изпитни задачи да се използват от учителите без промяна, защото след като са публично достъпни или са давани веднъж на контролно изпитване, те вече се знаят от учениците, а решенията са публично известни, така че не могат да се използват отново без промяна.
- **Тестове за ДЗИ**: примерни въпроси и отговори за Държавен зрелостен изпит по информатика. Те служат за тренировка на учениците за ДЗИ. Въпросите са много сходни на задачите по основи на програмирането и ООП, които вече са давани на ДЗИ по информатика в последните няколко години. Могат да се ползват от учителите и като насока при разработване на изпитни тестове.
- **Авторски решения**: сорс код на решенията на всички задачи от упражненията, практическите проекти и изпитите, достъпен са само за учителите, в **отделно непублично хранилище**.
 - Достъп до хранилището за учители се получава с контакт към екипа на СофтУни Фондацията.



- **Методически указания** за учителите за всяка изучавана тема. Съдържат наставления и насоки за целта на дадена тема, какво обхваща тя, нейната важност (дали е задължителна или опционална), на какво да се обърнат внимание учителите и други инструкции.

Хранилище на проекта

Разработените учебни ресурси за обучение по Модул 1 "ООП" са **публично достъпни** в хранилището на проекта в GitHub: <https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Software-Sciences/Module-1-OOP>.

Хранилището е предназначено за **учители и ученици** и е организирано по професии и профили и по учебни курсове. За всеки учебен курс е предоставено **описание, учебна програма**, а за всяка учебна тема има отделна **папка с учебни ресурси** и описание на темата. Ето как изглежда **описанието на курса** в GitHub:

Модул 1. "Обектно-ориентирано проектиране и програмиране"

Този модул е предвиден за преподаване в рамките на минимум **72 учебни часа** в XI клас от профил "Софтуерни и хардуерни науки".

Внимание: предвидените **72 часа** от МОН за предвиденото от МОН учебно съдържание са **крайно недостатъчни** и е невъзможно в толкова малко часове да се изучат на добро ниво толкова много учебни теми и концепции. За да се реши този въпрос или трябва да се **увеличи броят часове**, или трябва да се **съкрати учебното съдържание**. В помощ на учителите сме разработили 3 варианта на курса (с различен брой учебни теми) според броя часове:

Вариант	Описание	Необходими часове
Базов (без звездичка)	Обхваща най-важните теми от курса като покрива изискванията на МОН на съвсем базово ниво	72 часа (2 часа * 36 седмици)
Разширен (със звездичка)	Покрива напълно изискванията на МОН	108 часа (3 часа * 36 седмици)
Експерт (с 2 звездички)	Покрива изискванията на МОН + по-сложни теми за напреднали	144 часа (4 часа * 36 седмици)

Ето как са структурирани **папките с ресурси** в хранилището на курса в GitHub:



← → ↻	github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Software-Sciences/Module-1-OOP	🔍 📄 ⚙️	👤
📁	main ▾ School-Programming / Courses / Software-Sciences / Module-1-OOP /	↑ Top	
📁	..		
📁	01-Intro-to-Programming	Minor fixes for intro to programming exercises	2 months ago
📁	02-GUI-Apps	Renamed folders: Module-1-OOP (new / old)	2 months ago
📁	03.1-Conditional-Statements-Basics	Fixed animations for conditional statements basics	2 weeks ago
📁	03.2-Conditional-Statements-Advanced	Updated README.md for conditional statements advanced	last month
📁	03.3-Conditional-Statements-More-Exercises	Updated README.md for conditional statements more exer...	last month
📁	04.1-Loops-Basics	Minor fixes on loops basics	2 months ago
📁	04.2-Loops-Advanced	Updated README.md for loops advanced	last month
📁	04.3-Loops-More-Exercises	Updated README.md for loops more exercises	last month
📁	05.1-Methods-Basics	Updated README.md for methods basics	last month
📁	05.2-Methods-Advanced	Minor fixes on methods advanced exercises	2 months ago
📁	06-Mid-Exam	Minor fixes on mid exam	2 months ago
📁	07.1-Classes-and-Objects-Basics	Updated classes and objects basics exercises	2 months ago
📁	07.2-Classes-and-Objects-Advanced	Minor fixes on classes and objects advanced exercises	2 months ago

Ето какво съдържат папките за всяка една учебна тема от курса:

← → ↻	github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Software-Sciences/Module-1-OOP/04.1-Loops-Basics	🔍 📄 ⚙️	👤
📁	main ▾ School-Programming / Courses / Software-Sciences / Module-1-OOP / 04.1-Loops-Basics /	↑ Top	
📁	..		
📄	04.1-Loops-Basics-Exercises.docx	Minor fixes on loops basics	2 months ago
📄	04.1-Loops-Basics.pptx	Minor fixes on loops basics	2 months ago
📄	README.md	Renamed folders: Module-1-OOP (new / old)	2 months ago

README.md

Цикли (базов вариант)

Урокът "Цикли" (базов вариант) има за цел да запознае учениците с циклите **for** и **while** в езика C#:

- Конструкция и използване на **for**-цикли
- Работа с текст
- Конструкция и използване на **while**-цикли
- Вложени цикли

Методически указания

- Този вариант на темата е **задължителен** за всички, защото знанията за циклични конструкции ще бъдат **необходими** във всички следващи теми.



За всяка учебна тема сме създали кратко **описание**, **методически указания** за учителите, **презентация** и **упражнения** (в повечето случаи). За някои теми може да има **живи примери**, ресурси към упражненията и други файлове и документи.

Свободен лиценз

Разработените учебни ресурси за обучение по Модул 1 “ООП” са **публично достъпни** в хранилището на проекта в GitHub и се разпространяват под **свободен лиценз** (подобен на CC-BY-NC-SA).

Разработка на учебното съдържание

В периода **януари – юни 2023 г.** екип на Фондация “Софтуерен университет” в състав: д-р Светлин Наков, Александрина Механджийска и Стефан Куюмджиев разработи гореспоменатите учебни ресурси за “**Модул 1. Обектно-ориентирано проектиране и програмиране**”. Те са базирани на учебни материали на СофтУни, които са първоначално създадени от д-р Светлин Наков и във времето са подобрявани и разширявани от различни преподаватели и експерти от СофтУни, без ясна следа кой кога с какво точно е допринесъл.

Може да се твърди, че **обучителната концепция** за разработеното учебно съдържание е измислена от **д-р Светлин Наков**, а имплементацията на обучителните материали (слайдове, упражнения, проекти и други) е извършена от **широк екип от експерти** под ръководството и насоките на д-р Наков.

Теми от учебното съдържание

Завършеното учебно съдържание за **Модул 1 “ООП”** се състои от следните **теми** (уроци, упражнения и проекти):

Тема	Заглавие и коментар
Тема 1	Въведение в програмирането (базов урок, задължителен): запознаване с програмирането, среда за разработка, работа с данни и прости пресмятания с езика C# и средата Visual Studio Въведение в програмирането – упражнения : серия практически задачи за запознаване с програмирането (прости пресмятания с въвеждане и извеждане на данни).



Тема 2	GUI приложения с Windows Forms (базов урок, задължителен): запознаване с GUI приложенията и Windows Forms, работа с форми и UI контроли (Form, Button, Label, TextBox), обработка на събития GUI приложения – упражнения: изграждане на няколко прости GUI приложения с Windows Forms (визуализация на ден и час, суматор на числа, конвертори за валути)
Тема 3.1	Условни конструкции (базов вариант, задължителен): логически изрази, проверки, конструкция if и if-else, серии от проверки, конструкция switch-case, разклонена програмна логика Условни конструкции – упражнения: поредица от задачи върху писане на прости условни конструкции (четене на вход, сравнения и изчисления с условие), GUI конвертор на мерни единици
Тема 3.2	Условни конструкции – продължение (разширен вариант, опционален): вложени условни конструкции, логически оператори AND, OR, NOT, обхват и живот на променливите Условни конструкции – продължение – упражнения: задачи върху условни конструкции с по-сложни условия и програмна логика с вложени проверки, GUI калкулатор за цени на продукти
Тема 3.3	Условни конструкции – още упражнения (вариант експерт, опционален): по-предизвикателни задачи с условни конструкции, задача с GUI интерфейс (време + 15 минути)
Тема 4.1	Цикли (базов вариант, задължителен): повторение на блок от код, конструиране на for-цикли, използване на while-цикли, концепция за вложени цикли, примери и задачи Цикли – упражнения: прости задачи с въртене на цикъл от 1 до N, обхождане на елементите на символен низ, по-сложни задачи с цикъл със стъпка, прости задачи с вложени цикли
Тема 4.2	Цикли – продължение (разширен вариант, опционален): по-сложни for-цикли, цикъл с обратна стъпка, по-сложни while-цикли, по-сложни вложени цикли, прекъсване на вложен цикъл с двоен break Цикли – продължение – упражнения: по-сложни задачи с цикли със стъпка, вложени цикли на няколко нива, задачи с цикли и по-сложна логика, GUI app за визуализация на сграда с етажи и стаи
Тема 4.3	Цикли – още упражнения (вариант експерт, опционален): по-сложни задачи за цикли и нетривиална програмна логика с цикли, проверки и пресмятания, GUI визуализация на пирамида с числа
Тема 5.1	Методи (базов вариант, задължителен): концепция за подпрограма (метод), деклариране на метод, извикване на метод, подаване на аргументи и връщане на резултат от метод, добро именуване



	Методи – упражнения: задачи върху дефиниране и извикване на методи, приемане на параметри и връщане на стойност, GUI app за изчисления с числа
Тема 5.2	Методи – продължение (разширен вариант, опционален): съкратен синтаксис за дефиниране на метод, незадължителни параметри, варианти на методи, стойностни и референтни типове данни Методи – продължение – упражнения: по-сложни задачи, които комбинират методи, цикли и условни конструкции, извикване на методи един от друг
Тема 6	Контролна работа по основни на програмирането (препоръчителна): задача 1 – пресмятания с числа; задача 2 – пресмятания с условни конструкции; задача 3 – цикли (и условни конструкции)
Тема 7.1	Класове и обекти (базов вариант, задължителен): използване на класове и обекти, дефиниране на прости класове, полета, свойства, методи, конструктори, инстанциране на клас Класове и обекти – упражнения: задачи за упражнения на начално ниво върху дефиниране на прости класове и работа с обекти (инстанциране, достъп до свойство, извикване на метод)
Тема 7.2	Класове, обекти и списъци (разширен вариант, опционален): вериги от конструктори и инстанциране, списъци от обекти (създаване, добавяне, изтриване, обхождане, отпечатване, сортиране) Класове, обекти и списъци – упражнения: задачи върху класове, обекти и списъци от обекти, използване на списък от обекти като свойство в клас (агрегация 1 към много)
Тема 7.3	Класове и обекти – още упражнения (вариант експерт, опционален): по-сложни задачи с класове, обекти и списъци от обекти, моделиране на връзки между класовете, асоциация и композиция
Тема 8.1	Капсулация (базов вариант, задължителен): ООП принцип за "капсулация на данни", модификатори за достъп (public, private), хвърляне на изключение, валидация на данни Капсулация – упражнения: задачи за дефиниране на класове с private полета и public свойства, с конструктори и валидация на вътрешното състояние на обекта
Тема 8.2	Капсулация – продължение (разширен вариант, опционален): вериги от изключения и повторно хвърляне на изключение, работа с променими и непроменими обекти Капсулация – продължение – упражнения: по-сложни задачи върху дефиниране на класове, моделиране на реалния свят с класове, капсулация на данни и валидация



Тема 9.1	Наследяване (базов вариант, задължителен): концепция за наследяване на класове в ООП, дефиниране на йерархии от класове, достъп до членовете от базовия клас, преизползване на код Наследяване – упражнения: базови задачи върху наследяване на класове в ООП, дефиниране на базов клас и клас наследник, изграждане на йерархия чрез наследяване на няколко нива
Тема 9.2	Наследяване – продължение (разширен вариант, опционален): виртуални методи и преизползване на код, ОО моделиране, разширяване на клас, асоциация, композиция, делегация Наследяване – продължение – упражнения: по-сложни задачи за дефиниране на класови йерархии с преизползване на код чрез виртуални методи и обектно-ориентирано моделиране
Тема 10.1	Абстрактни класове и интерфейси (базов вариант, задължителен): принцип на абстракцията в ООП, дефиниране и имплементиране на интерфейси, абстрактни методи и абстрактни класове Абстрактни класове и интерфейси – упражнения: задачи върху интерфейси и абстрактни класове с виртуални и неvirtуални методи, моделиране на абстракции и конкретни имплементации
Тема 10.2	Абстрактни класове и интерфейси – допълнителни упражнения (разширен вариант, опционален): допълнителни практически задачи върху изграждане на йерархии от интерфейси и класове
Тема 11.1	Полиморфизъм (базов вариант, задължителен): полиморфизъм в ООП, достъп до обект през базовите му интерфейси и типове, virtual и override, оператори is и as в C#, overload на методи Полиморфизъм – упражнения: задачи върху полиморфизъм, дефиниране на йерархии от класове, предефиниране (override) на виртуални методи в клас-наследник, извикване през базов интерфейс
Тема 11.2	Полиморфизъм – допълнителни упражнения (разширен вариант, опционален): допълнителни задачи за затвърждаване на знанията за полиморфизъм и припокриването на виртуални методи
Тема 12.1	Делегати и събития (базов вариант, задължителен): концепция за делегати (референция към метод), записване на метод в делегат, извикване на делегат, предикати, събития и обработчици Делегати и събития – упражнения: практически задачи върху използване на делегати (обработка на колекции с делегат или ламбда функция), дефиниране на метод с параметър делегат
Тема 12.2	Делегати и събития – продължение (разширен вариант, опционален): дефиниране на собствени делегати, функции от по-висок ред, дефиниране на собствени събития и обработчици



	Делегати и събития – продължение – упражнения: упражнения върху дефиниране и използване на делегати, функции от по-висок ред и дефиниране и използване на собствени събития
Тема 13.1	Контролна работа по ООП (препоръчителна): задача 1 – несложен клас; задача 2 – базов клас с два наследника; задача 3 – ООП йерархия с интерфейс + базов клас + наследници + полиморфизъм
Тема 13.2	Теоретичен тест по модела на ДЗИ (опционален): задачи за подготовка за тест във формата на ДЗИ по Информатика (въпроси с по няколко реда код и списък от няколко възможни отговора)
Тема 14	Шаблони за дизайн (разширен вариант, опционален): преглед на най-разпространените шаблони за дизайн (design patterns) в ООП – шаблони за създаване, структурни и поведенчески шаблони Шаблони за дизайн – упражнения: постъпкова имплементация на класически design patterns: Singleton, Façade, Builder, Command, Prototype, Composite, Template
Тема 15	SOLID принципи (вариант експерт, опционален): единствена отговорност, отворен / затворен, заместване на Лисков, разделяне на интерфейсите, обръщане на зависимостите SOLID принципи – упражнения: преработка на съществуващи проекти със C# код, така че да спарват SOLID принципите за качествен код
Тема 16.1	Практически проект "Камък, ножица, хартия" (базов вариант, препоръчителен): постъпково ръководство за изграждане на много проста конзолна игра + качване в GitHub + публикуване онлайн
Тема 16.2	Практически проект "Познай числото" (базов вариант, препоръчителен): постъпково ръководство за изграждане на конзолен проект + качване в GitHub + публикуване онлайн
Тема 16.3	Практически проект "Бързо действие" (разширен вариант, опционален): постъпково ръководство за изграждане на конзолна игра + качване в GitHub + публикуване онлайн
Тема 16.4	Практически проект "Генератор на случайни изречения" (вариант експерт, опционален): постъпково ръководство за изграждане на конзолен проект + качване в GitHub + публикуване онлайн

Учебни презентации

За всяка учебна тема, в която е предвидено да се усвоява **нов учебен материал**, е разработена **презентация**, която поднася информацията по добре структуриран начин, последователно, с надграждане. **Принципите на изграждане на учебните**



презентации, зададени от д-р Светлин Наков, и в които вярва целия екип, участвал в проекта, са следните:

- **Учене чрез примери:** по-малко текст и обяснения, повече примери
 - В програмирането **най-важен е кодът** и програмистите рядко четат текста, когато кодът е четлив и разбираем.
 - По-лесно е да обясниш дадена концепция или задача с **добри примери**, отколкото с описателни разкази и дълги текстове.
- Принцип "**учене чрез правене**": преподавателят трябва да пише код на живо и да демонстрира как се ползва дадена концепция с "**жив пример**".
 - След обясненията на учителя следват упражненията, където учениците **пробват на живо с ръцете си** примерите от слайдовете.
- **Слайдовете** от презентацията **служат и на учителя и на ученика**.
 - Слайдовете заместват учебника в ръцете на **учителя**, но му позволяват **да ги презентира и обяснява** на дъската в клас.
 - Слайдовете заместват учебника в ръцете на **ученика**, но не са излишно подробни и са идеални за "**бърз преговор на материала**".
- **Малко текст, много примери**, картинки и илюстрации
 - **Без дълги изречения:** ползваме само фрази, които запазват смисъла на изречението, но в по-малко думи.
- **Картинките подсилват посланието:**
 - **Картинките** и илюстрациите трябва да подсилват посланието в слайда или да илюстрират визуално обясняваната концепция.
- **Анимирано появяване** на отделните части от слайда
 - Съдържанието в слайда се появява **анимирано, стъпка по стъпка**, защото ако голямо количество информация се покаже наведнъж, учениците се зачитат и не слушат учителя.
 - В идеалния случай трябва да се появят 1-2 изречения + картинка, да се даде време на учителя да ги обясни и след тях следващите.
- **Размер: 20-30 слайда**
 - Целта е една типична презентация да се събере в 1 учебен час.

Типичната **структура на презентациите** изглежда ето така:

- **Начална секция**



- **Заглавен слайд:** заглавие на урока, подзаглавие, картинка, автор
- **Съдържание на урока:** основни обхванати теми (секции)
- **Секция със съдържание**
 - **Заглавен слайд** на секция: заглавие, подзаглавие, картинка
 - **Теория:** концепции, обяснения, картинки
 - **Примери:** сорс код и резултат от изпълнението му
 - **Практическа задача:** условие + решение (ключов фрагмент от кода)
- **Следваща секция със съдържание ...**
- **Заклучителна секция**
 - **Обобщение:** кратко резюме на обхванатия в урока материал
 - **Въпроси:** слайд за задаване на въпроси

Ето няколко примера как изглеждат презентациите на практика:

Наследяване на класове

Базов клас, клас-наследник, йерархия от класове, достъп до базовия клас, преизползване на код



SoftUni Team
Technical Trainers
Software University

SoftUni

Software University
<https://about.softuni.bg/>

Съдържание Software University

1. **Наследяване**
 - Клас-наследник разширява базов клас
 - Наследява данни и действия и добавя нови
2. **Йерархия от класове**
 - Дърво от наследявания в ООП
3. **Достъп до базови членове на класа: base**
4. **Преизползване на код (code reuse)**
 - Общите данни методи отиват в базовия клас

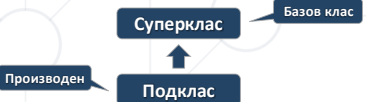
Наследяване на класове



Клас-наследник разширява базов клас

Наследяване Software University

- **Суперклас (superclass)** – родителски (parent) class, базов (base) class
 - Класът предава своите членове на класа дете (child)
- **Подклас (subclass)** – child class (дете/дъщерен клас), derived class (производен)
 - Класът взема членовете си от базовия клас





Наследяване в C#

- В езика C# наследяването се дефинира с оператора ":"

```
class Person { ... }  
class Student : Person { ... }  
class Employee : Person { ... }
```

8

Използване на наследени членове

- Можете да достъпите наследените членове както обикновено:

```
class Person { public void Sleep() { ... } }  
class Student : Person { ... }  
class Employee : Person { ... }
```

```
Student student = new Student();  
student.Sleep();  
Employee employee = new Employee();  
employee.Sleep();
```

10

Задача: Верижно наследяване

- Създайте класовете **Animal**, **Dog** и **Puppy**:
- Dog** трябва да наследява **Animal**
- Puppy** трябва да наследява **Dog**

```
Puppy puppy = new Puppy();  
puppy.Eat();  
puppy.Bark();  
puppy.Weep();
```

Проверете решението си тук: <https://judge.softuni.org/Contests/Practice/Index/4064#1>

18

Виртуални методи

- virtual** – дефинира метод, който **може да бъде презаписан**

```
public class Animal  
{  
    public virtual void Eat() { ... }  
}  
  
public class Dog : Animal  
{  
    public override void Eat() {}  
}
```

22

Обобщение

- Наследяването ни позволява да **преизползваме код**
 - Общият код отива в базовия клас
- Наследяването води до **йерархии**
- Подкласа **наследява** членовете от **суперкласа** и може да **презаписва** (подменя) методи

28

Въпроси?

© SoftUni – <https://softuni.org>. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Задания за упражнения

Задачите за упражнения са не по-малко важни от презентациите. Чрез тях учениците придобиват **умения**, които в програмирането са по-важни от теорията. За да са ефективни задачите за упражнения в един урок по програмиране, те трябва да отговарят на следните изисквания:

- **Да съответстват на учебния материал от презентацията:** обикновено за всяка секция от презентацията се правят по 1-2 упражнения.
- **Нарастваща сложност:** в началото трябва да стартираме с по-лесни упражнения, за да мотивираме и по-бавно напредващите ученици. След това **постепенно вдигаме сложността** докато достигнем до трудните задачи накрая на упражнението. Те обикновено са по силите само на по-силните ученици. Така **всеки ученик решава по няколко задачи**, според нивото си на придобити умения.



- **Повече обяснения в първите задачи, по-малко на следващите**
 - Практиката показва, че **първите задачи служат за пример** на учениците и е добре да са детайлно обяснени и **решението им да е разписано постъпково**. Това дава самочувствие на учениците, че могат да решат поне няколко задачи и ги мотивира да вървят напред, към по-сложните задачи.
 - За **следващите задачи** е препоръчително да бъдат дадени **напътствия**, но да се остави ученикът да си ги реши сам.
 - За **последните задачи** обикновено се дава **единствено условието** и примерен вход и изход. Те целят ученикът да поработи самостоятелно, за да развива **алгоритмичното си мислене** и уменията си за решаване на проблеми.
- **Задължително се дават примери:** най-добрият начин да обясниш задача по програмиране е като дадеш **старателно подготвени примери** с обяснения върху тях (примерен вход и съответен примерен изход за конзолните задачи). Този подход е стандартен и се спазва от десетилетия по олимпиади, състезания и изпити по програмиране.
- **Автоматизирано тестване в online judge система:** за всички задачи, за които е възможно, предоставяме линк към онлайн judge система за **моментална автоматизирана проверка**.
- **Картинки и илюстрации:** за задачите, за които има смисъл, се дават картинки, диаграми, или илюстрации. Отличен пример са GUI приложенията, за които се показва приблизително как може да изглежда крайният резултат, когато приложението е готово.

Всичките тези изисквания са базирани на **дългогодишния преподавателски опит** на **д-р Светлин Наков**, който е обучавал десетки хиляди студенти в продължение на десетилетия. Същите изисквания са **потвърдени и от анкетирани ИТ учители**, които използват задачи за упражнения в описания по-горе формат [[СофтУни фондация, 2022a](#)], [[СофтУни фондация, 2023](#)]. Този формат е **утвърдена практика** в СофтУни и в СофтУни Фондацията и е доказано ефективен.

Ето няколко **примера** как изглеждат **заданията за упражнения** на практика:



Упражнения: Повторения (цикли)

Тествайте решенията си в Judge системата: <https://judge.softuni.org/Contests/Practice/Index/3898>

1. Числа от 1 до 100

Напишете програма, която отпечатва числата от 1 до 100, всяко на нов ред.

Примери

Вход	Изход
(няма)	1 2 3 ... 98 99 100

Насоки

- Създайте нов проект с име "Numbers1To100".
- Създайте метода `Main(String[] args)`

```
static void Main(string[] args)
{
    ...
}
```
- Създайте **for-цикъл**, който да започва от 1 и да завършва на 10

```
for (int i = 1; i <= 100; i++)
{
    ...
}
```
- За всяка итерация на **for-цикъл**а отпечатайте **текущото число**

```
for (int i = 1; i <= 100; i++)
{
    Console.WriteLine(i);
}
```

11. Най-голямо число

Напишете програма, която до получаване на командата "Stop" чете **цели числа**, въведени от потребителя, и намира **най-голямото** измежду тях. Въвеждат се по едно число на ред.

Примери

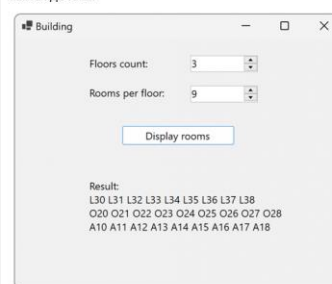
Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход
100	100	-10	20	45	99	999	999	-1	-1
99		20		-20		Stop		-2	
80		-30		7		99		Stop	
70		Stop		Stop					

Насоки

- Създайте променлива, която да съхранява **най-голямото число** и има начална стойност `int.MinValue`
- Прочетете **входните данни**
- Създайте **while-цикъл**, с който да итерирате до получаване на командата "Stop"
- Проверете дали прочетеното число е **по-голямо** от текущото **най-голямо** число
 - Ако е **по-голямо**, запазете това число като **максимално**
- Преминете към **следващите входни данни**

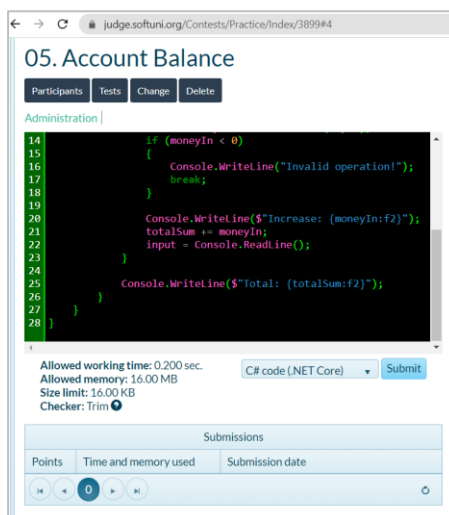
10. GUI App – Сграда

Реализирайте предходната задача като **Windows Forms GUI app**. Потребителският интерфейс може да изглежда така:



Система за автоматична проверка

Автоматизираната проверка на решения на задачи от упражнения и изпити, написани от учениците, **спестява много време и усилия** на учителите. Повечето задачи, които сме подготвили в учебните ресурси, идват заедно с линк към **online judge система**, където след пускане на кода за съответната задача, се получава **автоматична оценка** (на базата на поредица от тестове):



Submissions		
Points	Time and memory	Submission date
✓✓✓✓✓ 100 / 100	Memory: 0.00 MB Time: 0.160 s	22:40:39 02.07.2023 Details
Compile time error	---	22:40:35 02.07.2023 Details
✗✓✓✗✗ 40 / 100	Memory: 0.00 MB Time: 0.140 s	22:40:27 02.07.2023 Details
✓✗✗✓✓ 60 / 100	Memory: 0.00 MB Time: 0.130 s	22:40:02 02.07.2023 Details

Автоматизираната проверка има своите **плюсове и минуси**:



- Ученикът получава **незабавна обратна връзка** дали решението му е **коректно, грешно или частично коректно**, независимо дали работи в клас или в домашни условия.
- Ученикът получава от judge системата обратна връзка единствено за **коректността на решението**, но не може да получи съвети от учителя относно подхода към решаване на задачата, качеството на кода и други насоки. Все пак **учителският поглед си остава полезен**.
- Понякога при коректно условие judge системата дава **нулев резултат**, заради неспазване на формата на изхода или друга техническа подробност.

Задачи за практически изпити

Като част от учебното съдържание за обучение в Модул 1 “ООП” са разработени задачи за **контролни изпитвания в клас** (изпити). Те са **ориентировъчни** и дават на учителя примерна **структура на изпита, типовете задачи**, които може да се подготвят и **примерни задачи** за ориентир. Ето и някои препоръки:

- **Не се препоръчва да се използват директно примерните изпити** за реално оценяване на придобитите умения, тъй като учениците са много добри в търсене и споделяне на решения в Интернет и по между си и това може да компрометира оценяването.
- Всички задачи от примерните изпити могат да бъдат решени изцяло със знанията, придобити от **базовите варианти на темите**. Подходящи са за всички ученици, независимо от варианта на програмата, която се следва (базова, разширена или експертна).

Ето **пример** как изглеждат **указанията за контролни изпитвания** на практика:



Контролна работа (основи на програмирането)

Първата контролна работа за срока има за цел да провери знанията на учениците по темите, изучавани до този момент:

- Въведение в програмирането
- Условни конструкции
- Цикли

Задачи

Препоръчителен обхват на задачите:

- Задача 1: прости пресмятания с числа
- Задача 2: пресмятания с условни конструкции
- Задача 3: цикли (и условни конструкции)

Изпитни тестове във формат ДЗИ

Като част от учебното съдържание за обучение в Модул 1 "ООП" са разработени и **примерни изпитни тестове**. Те съдържат въпроси за подготовка за **теоретичен тест по модела на ДЗИ** (Държавен Зрелостен Изпит по "Информатика" за профил "Софтуерни и хардуерни науки").

- Въпросите са в типичния за **ДЗИ** формат и обхващат целия учебен материал от настоящия модул (основи на програмирането и ООП).
- Всеки въпрос съдържа **описание, няколко реда код и въпрос** с няколко възможни отговора, измежду които трябва да се избере правилният.

Въпросите от тестовете може да се ползват в **допълнение към задачите за контролна работа** или като **алтернатива на контролно изпитване** с решаване на задача (писане на код) - по преценка на учителя.

Ето **пример** как изглеждат **въпросите от теста за ДЗИ** на практика:



3. Каква ще бъде изходната стойност на следния C# код?

```
int x = 5;  
int y = 10;  
int z = (x > y) ? x : y;  
Console.WriteLine(z);
```

- A) 5
- Б) 10
- В) True
- Г) Грешка при компилация

9. Колко пъти ще се изпише "OK" на конзолата след изпълнението на следния код?

```
int i = 0;  
while (i == 0)  
{  
    Console.WriteLine("OK");  
    if (i == 1)  
        break;  
}
```

- A) 0
- Б) 1
- В) Кодът не е коректен, защото има допусната синтактична грешка.
- Г) Безброй много пъти.

Практически проекти

Практическите проекти имат за цел да предоставят на учениците възможност да затвърдят знанията и уменията си чрез създаването на **цялостно приложение на C#**, което решава някаква **интересна задача** (най-често реализира игра). Опитът ни показва, че много ученици се запалват и мотивират да работят по проектите, защото те са им **по-интересни** от задачите в упражненията.

Проектите, които сме предложили за учениците, са детайлно разписани и се състоят от **детайлно задание** (описание, визуализации и снимки от екрана), **постъпкови инструкции** за реализацията и изграждат и допълнителни умения отвъд писането на код: работа със **сорс контрол** хранилище в GitHub, качване на проекта за стартиране на живо в Интернет (**live demo**) и други полезни трикове.

Проектите могат да се ползват в следните **сценарии**:

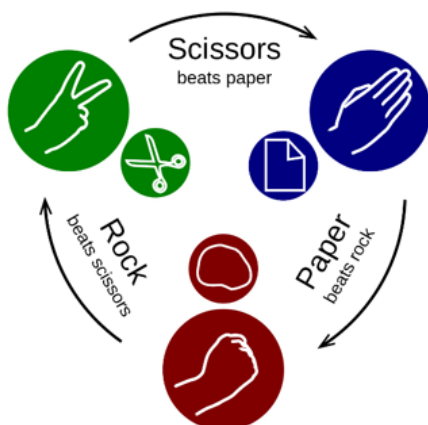
- За **по-напредналите ученици**, които бързо решават всички задачи от упражненията и след това случаят в клас. Те могат да работят по проекти, за да им е интересно докато другите решават задачите с по-ниска скорост.
- За ученици, на които им е **скучно да решават задачи**, защото не могат да намерят смисъла от тях. За тях проектите често пъти са по интересни.
- За работа в **извънкласни занимания** по програмиране.
- За **домашна работа**, като част от критериите за завършване на модула.

Ето **пример** как може да изглежда един **практически проект** на практика:



Практически проект: Камък, ножица, хартия

Днес ще разработим конзолна игра "Камък, ножица, хартия":



Камък, ножица, хартия е игра за двама, където вие и вашият опонент (компютъра) едновременно избирате една от следните три опции: **камък (rock)**, **ножица (scissors)** или **хартия (paper)**. Правилата са следните:

- Камъкът побеждава ножицата (ножицата се чупи от камъка)
- Ножицата побеждава хартията (хартията се реже от ножицата)
- Хартията побеждава камъка (камъкът се покрива от хартията)

Победител е потребителят, чийто избор побеждава избора на неговия опонент. Ако и двамата потребители изберат една и съща опция (например "хартия"), резултатът от играта е **"draw"** (равен).

```
Microsoft Visual Studio ...  
Choose [r]ock, [p]aper or [s]cissors: r  
The computer chose Rock.  
This game was a draw.
```

```
Microsoft Visual Studio ...  
Choose [r]ock, [p]aper or [s]cissors: r  
The computer chose Scissors.  
You win.
```

Статистика за свършеното

Следва статистика на свършената работа по изграждане на свободно учебно съдържание за българските ИТ учители за преподаване в Модул 1 "ООП" в гимназиален профил "Софтуерни и хардуерни науки":

- Изготвени са **20 презентации**, които съдържат **над 650 слайда** (с обучително съдържание за преподаване в клас), общо **110 страници** (ако се отпечата по 6 слайда на страница).
- Създадени са **25 задания за упражнения**, които съдържат общо **224 задачи**, разписани на общо **235 страници** (условия + напътствия).
- Разписани са **4 проекта**, съдържащи общо **97 страници** с инструкции за изграждане на проекти.
- Подготвени са общо **2 примерни изпита**, които съдържат общо **6 изпитни задачи** и се състоят от общо **7 страници**.



- Изготвени са общо **3 теста** с въпроси по Информатика за ДЗИ, които съдържат общо **45 въпроса**, разписани на общо **16 страници**.
- **Сумарно**, ако цялото учебно съдържание за курса се отпечата на хартия, ще са необходими **465 страници**.
- Към юни 2023 г. проектът има близо **100 звездички в GitHub**: <https://github.com/BG-IT-Edu/School-Programming/stargazers>. Това е измерител на популярността на open source проектите в GitHub.

Глава 3. Внедряване и оценка на въздействието

След завършване основната работа по разработката на учебните материали по “Модул 1: Обектно-ориентирано проектиране и програмиране” екипът на фондация “Софтуерен университет” сподели публично всички разработени ресурси – презентации, задачи, решения, примерни изпити, ДЗИ тестове и проекти и събра **обратна връзка от учители**, които ги използват.

Анкета от учители по ООП – май 2023 г.

На **17 май 2023 г.** д-р Светлин Наков и Александрина Механджийска проведоха **семинар с ИТ учители** от гимназии в цялата страна с цел дискусия на постигнатия прогрес и получаване на **обратна връзка** относно разработените учебни материали за **Модул “ООП”** от профилиращ предмет “Информатика” за гимназиален профил “Софтуерни и хардуерни науки” [СофтУни фондация, 2023]. По време на семинара беше събрана **обратна връзка от участниците** под формата на **анкета** с няколко въпроса. Анкетата беше попълнена от **ИТ учители от цялата страна**, повечето от които използват учебните материали на СофтУни Фондацията в ежедневната си учителска работа, включително разработените учебни ресурси за Модул “ООП”.

Участници в анкетата

Анкетата за обратна връзка за разработените учебни материали за преподаване в Модул 1 “ООП” попълниха **45 ИТ учителя**:



Име на учител

45 responses

Турхан Сюлейманов	Ивелина Антимова	Живка Попдончева	Таня Димитрова	Диляна Димитрова	Йордан Николов	Данка Михайлова	Калинка Ангелова
Наташа Ангелова	Зинаида Чернева	Павлина Дончева	Веселина Стоянова	Донка Георгиева	Антоанета Вълкова	Валентина Тодорова	Красимира Бозева
Сезгин Исмаил	Цветяна Йовева	Жулиета Петрова	Мартин Катев	Ивелина Димова Вълчева	Росица Траянова	Елена Гълбачева	Георги Стоянов
Беркант Сейдолски	Драгомира Стоянова	Делка Куртакова	Мая Караджова	Сашка Петрова	Стефка Петкова	Фикри Хърлов	Анна Николова Георгиева
Марина Георгиева	Златина Иванова	Лидия Атанасова	Виолета Микова	Нели Гунова	Елена Пенчева-Колева	Мария Бурла	Петя Драгоева
Тихомира Тенчева	Стелиана Чиликова	Петко	Антоанета Иванова	Павлина Сербезова			

Училище и град

39 responses

ППМГ "Акад. Н. Обрешков" - гр. Разград	ОУ "Св. св. Кирил и Методий" Силистра	СУ "Железник" Стара Загора	31 СУЧЕМ "Иван Вазов" гр. София	Град Раднево	33 ОУ САНКТ ПЕТЕРБУРГ, СОФИЯ	гр. Враца, ППМГ "Академик Иван Ценов"	ППМГ "Добри Чинтулов" град Сливен
Добрич	ПМГ "Акад. С. Корольов" Благоевград	ППМГ "Акад. Никола Обрешков" Бургас	ПГПЧЕ "Димитър Димов" - Плевен	гр. Пловдив СУ "Свети Константин-Кирил Философ"	Момчилград	ПМГ "Проф. Ем. Иванов"; град Кюстендил	СУ "Иван Вазов" - Стара Загора
ППМГ - Разград	МГ "Д-р Петър Берон" - Варна	Математическа Гимназия - Варна	СУ "Отец Паисий", гр. Кърджали	29. СУ "Кузман Шапкарев", София	СУ "Христо Ботев" - гр. Глоджево	ПМГ "Иван Вазов" - Димитровград	55 СУ "Петко Каравелов" гр. София
НПМГ София	Нова Загора СУ "Христо Ботев"	ППМГ "Добри Чинтулов" - Сливен	СУ "Христо Ботев" гр. Тутракан	СУ "Христо Ботев", гр. Тутракан	СУ "П.К. Яворов" гр. Чирпан	ПМГ "К. Величков" Пазарджик	СУ "Евлоги Георгиев" град Тръстеник

7. СУ "Свети Седмочисленици", гр. София	СУ "П. Кр. Яворов" гр. Пловдив	ПМГ "Иван Вазов" - Димитровград	СУ "П. Яворов" гр. Пловдив	27. СУ "Акад. Г. Караславов"	СУ "Св. Климент Охридски" гр. Аксаково	СУ "Александър Иванов-Чапай" гр. Белово
---	--------------------------------	---------------------------------	----------------------------	------------------------------	--	---

Резултати от анкетата

Резултатите бяха **впечатляващи**. Учителите споделиха, че екипът на Фондацията е осигурил ценни за тях ресурси и е успял да предостави решение на най-големите проблеми, с които са се сблъскали досега: **липса на учебно съдържание, недостиг на часове, твърде голям обем на материала**.



Обратна връзка за **разпределението на часовете**

35 responses

Добре е структурирано	ok
да	da
разпределението е добро	отлично
Добре е разпределението, ние сме на 4 часа и пак не сме експерти.	Нужно е да сме проактивни
Добре е	подходящо е
Много полезно	Разпределението е добре
Не достигат	В моето училище имаме часове в 8 и 9 клас, така че до 10-ти имат добра основа за ООП, структури и алгоритми.

Обратна връзка за учебните **презентации**

41 responses

добре структурирани	отлично	много добри
Ползвам ги от няколко месеца, чрез мултимедия.	приложими са, доста полезни	Адмирации за огромният ви труд
полезни	Много полезно	Чудесно са направени
Супер са!	Супер са	От това което видях в срещата са супер :)
Отличен 6.00 :)	чудесни - все едно аз съм си ги правила като логика	Доста подробни презентации, което е хубаво. Аз си ги преработвам, за да ми стигнат 40 минути за теория и да има време за 1 или 2 задачи, за още 40 минути. За повече няма време, материала е много, задъхваме се в Модул 1.

Презентациите ми харесват. Те са изчерпателни и с анимации, които са най-добро средство за преподаване и усвояване на знанията и уменията по определена тема	Вашите презентации са идеално методически издържани. Визуални, с анимация, която позволява фокусът да се задържа на конкретната част от преподаването. Догодина ще ползвам за пръв път на практика повечето от Вашите разработки. Сега в 8. и 9. клас ми влизат в употреба - 3. и 4. тема.	Презентациите са направени по подробен начин, така че и учениците да могат да се подготвят и самостоятелно с тях и да ги разберат. Могат да се редактират и от нас, което също е голям плюс.
Изключително полезни има по- подробно за елементи на графичния интерфейс, което се забравя от 8 до 11 клас.	Мисля, че са полезни.	?
Отлични!	учебните презентации са ми много полезни	



Изключително полезни - синтезирано и атрактивно представят учебния материал. Ползвам ги постоянно.	Идеални!	Много добър е материала, има примери
Много добре подготвени като съдържание, много лесно се работи в часовете. Ползвам ги благодарение на курса за учители, който организира СофтУни.	Удобни презентации, отлично представена информация	Добри
Страхотни са ! Благодаря!	Перфектно разработени и полезни	Допадат ми

Обратна връзка за упражненията

40 responses

Много добре подбрани задачи.	Някои от задачите са прекалено усложнени и с дълъг текст	ok
достатъчни са	да	обхващат материала и има достатъчно задачи
отлично	много добри	Харесват ми
полезни са	Много полезни	Има доста задачи, и ми харесват
интересни	Много полезно	Практически са много полезни
Супер са!	Ще ги разгледам по-късно по-подробно.	Ще ги огледам в детайли лятото.
Отличен 6.00 :)	напълно достатъчни с добра грацията	Достатъчно много задачи сте включили, имаме избор според нивото на учениците.

Неизменна част от работата ми - както за упражнение в клас, така и за домашна работа	Идеални!	Подходящи и различен тип задачи
Да	Много добри	Достатъчно упражнения, обхващащи всичко по дадените теми
Добри	Страхотни са ! Благодаря!	Перфектно разработени и полезни
Много добре	Упражненията са много добре структурирани и степенувани по сложност	За упражненията мога отново да повторя - ПРЕКРАСНА методика. А това, че могат да си тестват кода в judge е безценно.
Упражненията обхващат както най-важния материал, така и с по-голяма сложност. Което е подходящо за всеки тип ученици. Такива които се справят със затруднения и такива, които са напреднали.	полезни	Много полезни!



Обратна връзка за **изпитните материали**

36 responses

Добре са	ok	не мога да преценя
да	има нужда от още тестове за матури	може и по-сложни
много добри	все още е рано да кажа, трябва да ги изпробвам...	Ще ги пробвам догодина
Няма как да преценя без да съм ги приложила на практика, просто вече сме в модул2	полезни	Много полезно
Много полезни	Супер са!	Ще ги разгледам по-късно по-подробно.
-	Ок са	Удовлетворителни! Може още някой друг тест и вариант да се разработи за ДЗИ, че и там има дефицит на материали
Да, трябва да са съобразени с ДЗИ	Идеални!	Да

Обратна връзка за **практическите проекти**

36 responses

Не съм ги виждал още	ok	не мога да преценя
харесват ми	отлични	много добри
....	Аз лично имах огромна нужда от помощ за проекти	Някои от тях съм давала на учениците си, и дори по-слабите проявят интерес да правят игри
полезни	Много полезно	Практически приложими са и стойности
Супер са!	Ще ги разгледам по-късно по-подробно.	-
Може да се добавят и няколко такива с GUI.	Интересни	Много добре и подробно са описани и дават възможност учениците да да сглобят отделните парченца, които са учили, за да видят по-цялостна картина.
Идеални!	Да	Мисля, че ще са интересни за учениците.

Изводи от анкетата

Ето и **изводите** от изпратените коментари в анкетата:

- **Учебните материали** са структурирани добре и са изключително полезни на учителите. Десетки анкетираните учители ги ползват постоянно.
- Разделянето на темите на **три варианта** с по-малък и по-голям хорариум предоставя **гъвкавост** спрямо наличните учебни часове и нивото на учениците. Разработените подробни **учебни разписания** за отделните варианти на програмата осигуряват ясен и систематизиран път за преподаване на учебния материал.



- **Презентациите** са оценени много високо и форматът им много се харесва на учителите. Подходът с малко текст, много примери и анимирано показване на всяка следваща част от слайда се харесва на учителите.
- **Упражненията** като цяло се одобряват. Учителите харесват, че има много задачи, измежду които да си избират. Ползват ги и за упражнения в клас и за домашни. Харесват методиката и логиката на упражненията. Харесват възможността за автоматична проверка в онлайн judge системата.
- Няма разработени въпроси за подготовка за ДЗИ. Към момента на провеждане на анкетата действително нямаше разработени **изпитни въпроси за ДЗИ**. Взехме предвид обратната връзка по този въпрос и разработихме 45 изпитни въпроса за подготовка за ДЗИ по Информатика.
- Наличието на **практически проекти** дава възможност на учениците да използват наученото за разработка на приложения и игри, като по този начин им дава **мотивация** да продължат да развиват уменията си в областта на програмирането.
- Има училища, в които се очаква от следващата учебна година да се направи **преход от Java към C#**, за да може да се използват предоставените учебни материали от СофтУни Фондацията.

На базата на **проведената анкета и мненията на десетки учители по ООП**, повечето от които ползват разработените учебни материали за Модул 1 “ООП” в своята ежедневна работа, можем да сме доволни, че **сме свършили добра работа**, която е наистина **приложима и полезна** на ИТ учителите в България.

Целите на дипломната работа са изпълнени повече от успешно.

Заклучение

В периода януари–юни 2023 г. екип на Фондация “Софтуерен университет” в състав: д-р Светлин Наков, Александрина Механджийска и Стефан Куюмджиев разработи описаните в настоящата дипломна работа безплатни учебни ресурси за учители за **Модул 1. “Обектно-ориентирано проектиране и програмиране”** от профилираната подготовка по информатика за гимназиален профил “Софтуерни и хардуерни науки”.



Разработените учебни материали са **много високо оценени от българските учители по информатика и ИТ** и се ползват масово от тях. Проектът е **изключително успешен и полезен за българското гимназиално ИТ образование** и има значим позитивен ефект върху цялостната технологична екосистема в България (образование / иновации / предприемачество / бизнес).

Планове за развитие

Към юни 2023 г. екипът на Фондацията и е в напреднал стадий на разработка на учебни ресурси и за **Модул 2. “Структури от данни и алгоритми”** от профил “Софтуерни и хардуерни науки”, в същия формат и дълбочина.

До края на 2023 година екипът на Фондацията, в разширен състав, планира разработка на качествени учебни ресурси за обучение по следните учебни модули в помощ на българските учители:

- **Модул 2. “Структури от данни и алгоритми”** за 11 клас от профилираната подготовка по информатика
- **Модул 3. “Релационен модел на бази от данни”** за 12 клас от профилираната подготовка по информатика
- **Модул 4. “Програмиране на информационни системи”** за 12 клас от профилираната подготовка по информатика
- **Компютърно моделиране и информационни технологии** за 5 клас от общообразователната по ИТ
- **Компютърно моделиране и информационни технологии** за 6 клас от общообразователната по ИТ

За да се изпълнят планираните разработки, към екипа се присъединяват **д-р Мухарем Моллов** (учител по програмиране и ИТ с дългогодишен опит, преподавател по програмата “Обучение за ИТ кариера”) и още един **нов колега**.

Разработката за проекта се финансира от **Фондация “Софтуерен университет”** чрез дарения от основателите на Фондацията и външни съмишленици.



Благодарности

Като автор на настоящата дипломна работа и като създател и ръководител на проекта за **свободно учебно съдържание за българските ИТ учители**, като учител, родител и гражданин, мога да споделя, че съм **горд, че съм в основата на този мащабен и значим за българското училищно образование проект**, който помага на **стотици учители** да преподават качествено програмиране и на **хиляди ученици** от цяла България да учат програмиране с удоволствие и лекота.

Благодарности към целия екип на СофтУни фондацията, към всички директни участници в проекта (Александрина Механджийска, Стефан Куюмджиев, Мухарем Моллов, Ангел Георгиев), към всички учители, които споделиха свои учебни ресурси, към учителите, които участваха в проучванията и анкетите, към колегите от БАСКОМ, към колегите от СофтУни и към цялата ИТ академична общност в България, към съпругата ми (която ме изтърпя докато работя и пиша по нощите), научния ми ръководител гл. ас. д-р Симеон Монов (който ме подкрепяше и помагаше през цялото време), на рецензента ми проф. Коста Гъров (за градивната критика и мащабният му принос към ИТ образованието в България). Благодаря ви на всички, че ме подкрепихте в тази кауза!



Информационни източници

Използваните литературни източници са подредени **по азбучен ред** според фамилиното име на първия автор и по втори критерий **година на публикуване**, като първо са цитирани източниците на латиница, а след тях тези на кирилица:

1. [BG-IT-Edu, 2020] Проект “Отворено учебно съдържание по програмиране и ИТ за българските училища”, Фондация “Софтуерен университет”, 2020-2023 г, <https://github.com/BG-IT-Edu/School-Programming>
2. [ЗПУО, 2023] Министерство на образованието и науката, “Закон за предучилищното и училищното образование (ЗПУО)”, 2023 г., https://web.mon.bg/upload/34577/Zkn_PedUchObrazovanie-izm07022023.pdf
3. [МОН, 2015] Министерство на образованието и науката, “Наредба № 4 от 30.11.2015 г. за учебния план (рамкови учебни планове за общообразователна, профилирана и професионална подготовка)”, https://web.mon.bg/upload/32936/nrdb4-2015_izm2022-ucheben-plan-31082022.pdf
4. [МОН, 2016a] Министерство на образованието и науката, “Наредба № 7 от 11.07.2016 г. за профилираната подготовка”, https://cioo.mon.bg/wp-content/uploads/2014/07/naredba_7_11.08.2016_profilirana_podgotovka.pdf
5. [МОН, 2016b] Министерство на образованието и науката, “Наредба за изменение и допълнение на Наредба № 7 от 2016 г. за профилираната подготовка – учебно-изпитни програми за държавните зрелостните изпити (ДЗИ)”, <https://www.mon.bg/upload/23985/izmNdrb-PP-24092020.pdf>
6. [МОН, 2017] Министерство на образованието и науката, “Заповед РД09-1808/17.03.2017 г. за рамкови програми В за професионално образование”, https://web.mon.bg/upload/3835/3_Programa_V_2017.pdf
7. [МОН, 2018a] Министерство на образованието и науката, “Заповед РД09-4022/21.12.2018 г. за учебните програми в профил Информатика”, <https://web.mon.bg/upload/18364/profil-Informatika.pdf>
8. [МОН, 2018b] Министерство на образованието и науката, “Заповед РД09-4022/21.12.2018 г. за учебните програми в профил Информационни технологии”, <https://web.mon.bg/upload/18365/profil-IT.pdf>



9. [МОН, 2018с] Министерство на образованието и науката, “Наредба № 1 от 15.01.2018 г. за придобиване на квалификация по професията Приложен програмист”, <https://navet.government.bg/bg/media/DOS-Prilozhen-programist.pdf>
10. [МОН, 2018d] Министерство на образованието и науката, “Наредба № 19 от 9.10.2018 г. за придобиване на квалификация по професията Системен програмист”, https://navet.government.bg/bg/media/sisitemen_programist.pdf
11. [МОН, 2018е] Министерство на образованието и науката, “Учебни програми за III клас в сила от учебната 2018-2019 година”, <https://web.mon.bg/bg/1689>
12. [МОН, 2019а] Министерство на образованието и науката, “Учебни програми за IV клас в сила от учебната 2019/2020 г.”, <https://web.mon.bg/bg/2190>
13. [МОН, 2019b] Министерство на образованието и науката, “Наредба № 18 от 23.08.2019 г. за придобиване на квалификация по професията Програмист”, https://navet.government.bg/bg/media/DOS_PROGRAMIST.pdf
14. [МОН, 2020] Министерство на образованието и науката, “Заповед РД09-2819/09.10.2020 г. за учебните програми по компютърно моделиране в 5, 6 и 7 клас”, <https://web.mon.bg/upload/24189/zap2819-PC-modelirane14102020.pdf>
15. [МОН, 2021] Министерство на образованието и науката, “Протокол №4/29.09.2021 г. – Държавен образователен стандарт за придобиване на квалификация по професия Програмист на изкуствен интелект”, https://www.navet.government.bg/bg/media/dos_eru_programist_ai.pdf
16. [МОН, 2022] Министерство на образованието и науката, “Протокол № 1/23.02.2022 г. – Държавен образователен стандарт за придобиване на квалификация по професия Програмист на роботи”, https://www.navet.government.bg/bg/media/dos_eru_programist_roboti.pdf
17. [Наков, 2021] Наков С., “Свободно ИТ учебно съдържание за учители по програмиране и ИТ”, София, OpenFest, 2021 г., <https://nakov.com/blog/2021/08/14/open-learning-resources-for-it-and-programming-teachers>



18. [НСИ, 2023a] Национален статистически институт, “Общообразователни училища по вид”, 2023 г.,
<https://www.nsi.bg/bg/content/3477/общообразователни-училища-по-вид>
19. [НСИ, 2023b] Национален статистически институт, “Учащи по групи класове и пол в общообразователните училища по форма на собственост”, 2023 г.,
<https://nsi.bg/bg/content/3440/учащи-по-групи-класове-и-пол-в-общообразователните-училища-по-форма-на-собственост>
20. [НСИ, 2023c] Национален статистически институт, “Професионални училища по вид”, 2023 г., <https://nsi.bg/bg/content/3481/професионални-училища-по-вид>
21. [НСИ, 2023d] Национален статистически институт, “Учащи в професионалните училища”, 2023 г., <https://nsi.bg/bg/content/3444/учащи-в-професионалните-училища>
22. [СофтУни фондация, 2017] Проект “Обучение за ИТ кариера” (09/2017),
<https://softuni.foundation/projects/training-for-it-career>
23. [СофтУни фондация, 2022a] Фондация “Софтуерен университет”, Анкета към българските ИТ учители: от какво имате нужда? (11/2022),
<https://softuni.foundation/news/2022/11/19/survey-to-bulgarian-it-teachers-what-do-you-need>
24. [СофтУни фондация, 2022b] Постоянен екип “учебно съдържание по програмиране и ИТ за учители” в СофтУни фондацията (12/2022),
<https://softuni.foundation/news/2022/12/21/team-educational-content-on-programming-and-it-for-teachers>
25. [СофтУни фондация, 2023] Анкета от учители по ООП – май 2023 г. – за безплатния ООП курс на Фондацията: обратна връзка за "Модул 1: Обектно-ориентирано проектиране и програмиране" (05/2023),
<https://softuni.foundation/news/2023/05/20/survey-oop-teachers-may-2023>
26. [СофтУни, 2021] “Обучение за учители по Информатика за профилирана подготовка - септември 2021”, <https://softuni.bg/trainings/3523/training-for-teachers-in-informatics-for-profiled-education-september-2021>