

Врз основа на член 55 став 1 од Законот за организација и работа на органите на државната управа („Службен весник на Република Северна Македонија“ бр. 58/00, 44/02, 82/08, 167/10, 51/11, 96/2019 и 110/2019) и член 22 став 1 од Законот за средно образование („Службен весник на Република Северна Македонија“ бр. 44/95, 24/96, 34/96, 35/97, 82/99, 29/02, 40/03, 42/03, 67/04, 55/05, 113/05, 35/06, 30/07, 49/07, 81/08, 92/08, 33/10, 116/10, 156/10, 18/11, 42/11, 51/11, 6/12, 100/12, 24/13, 41/14, 116/14, 135/14, 10/15, 98/15, 145/15, 30/16, 127/16, 67/17, 64/18 и 229/20) и член 3 и член 4 став 5 од Законот за математичко-информатичка гимназија („Службен весник на Република Северна Македонија“ бр. 64/18) министерот за образование и наука ја донесе Наставната програма по наставниот предмет *програмирање* за II (втора) година за математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма

ПРОГРАМИРАЊЕ

за II година

Математичко-информатичка гимназија

Скопје, 2021 година

Назив на наставната програма	Програмирање
Тип на наставна програма	Задолжителна
Кредитна вредност на наставната програма	7 (седум) ЕЦВЕТ ¹ кредити (5+2, 2 кредита одговараат на 50 часа активности на ученикот од кои 18 часа за домашна работа, 12 часа за подготовка за писмени работи и 20 часа за самостојно учење)
Ниво на квалификација	IV (четврто) ниво
Година на изучување	II (втора)
Број на часови неделно/годишно за реализација на наставната програма	3/108
Цели на наставната програма	<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> - да ги продлабочи знаењата по информатика во областа на програмирањето и решавањето проблеми и да ги применува стекнатите знаења во секојдневни ситуации, во соодветни предизвици како и во други наставни предмети; - да стекне самодоверба за примена на стекнатите програмерски и алгоритамски вештини за наоѓање, користење и презентирање на програмите; - да ја цени моќта, корисноста и интернационалната димензија на информатиката и програмирањето; - да гради позитивни ставови за примената на програмирањето;

¹Закон за Националната рамка на квалификации.

	<ul style="list-style-type: none"> - активно да учествува до донесување суштински одлуки при соработка со други; - да развива логичко, критичко и креативно алгоритамско размислување.
<p>Теми/подрачја/модуларни единици на наставна програма</p>	<p>ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈАЗИЦИ (4 часа) ОПЕРАЦИИ СО БИНАРНИ БРОЕВИ. ДЕТАЛЕН ПРЕГЛЕД НА ОСНОВНИТЕ ТИПОВИ НА ПОДАТОЦИ (5 часа) ПРОБЛЕМСКИ ЗАДАЧИ КОИ СЕ РЕШАВААТ СО ОСНОВНИ УПРАВУВАЧКИ СТРУКТУРИ (9 часа) ПРОБЛЕМСКИ ЗАДАЧИ КОИ СЕ РЕШАВААТ СО НИЗИ (21 часа) ФУНКЦИСКИ ШАБЛОНИ, РЕКУРЗИЈА И РЕКУРЗИВНИ ФУНКЦИИ (12 часа) ПОКАЖУВАЧИ (7 часа) ПОДАТОЦИ ОД ТИП СТРУКТУРА (STRUCT) (5 часа) ПОДАТОЧНИ СТРУКТУРИ (18 часа) ОПШТИ ТЕХНИКИ ЗА КОНСТРУКЦИЈА НА АЛГОРИТМИ (27 часа)</p>
<p>Материјално-технички и просторни услови</p>	<p>За постигнување на целите на наставата по програмирање неопходно е стручно осмислена и планирана примена на различни наставни средства и задолжително компјутер за секој ученик, со соодветно инсталирани програмски пакети и прилагодени привилегии за корисникот, поврзани на интернет. Наставникот треба да поседува преносен компјутер и опрема за проектирање.</p>
<p>Норматив на наставен кадар</p>	<p>Наставната програма може да ја реализира стручно лице кое има познавање од англиски јазик и вештини за користење компјутер со стекнат:</p> <ul style="list-style-type: none"> -наставник со завршени студии по информатика/наставна или друга насока, VII/1 или VIA според МРК и 240 ЕКТС; <p>Стручно лице кое исполнува најмалку еден од следните услови:</p> <ul style="list-style-type: none"> -да бил ментор на ученик кој бил награден напрестижен меѓународен натпревар од соодветната област; научен степен доктор на информатички науки; -да е запишан на докторски студии на соодветната област; -да има стекнато научен степен на доктор на науки на соодветната област.

ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈАЗИЦИ (4 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на :</p> <ul style="list-style-type: none"> -да знае за историјатот и концептите за програмирање и програмски јазик; -да умее да направи дистинкција помеѓу програмски јазичи на пониско и повисоко ниво (предности и недостатоци); -да се запознае со поделба на програмските јазичи според намената; -да се запознае со поделба на програмските јазичи според начинот на обработка на податоците; -да умее да направи јасна разлика помеѓу јазичи со структурен/процедурален, 	<p>Содржини</p> <ul style="list-style-type: none"> • Програмирање и програмски јазичи • Историјат на програмските јазичи (машински, асембли-симболички, виши програмски јазичи) • Поделба на програмски јазичи: <ul style="list-style-type: none"> - според намена и - според начин на обработка на податоци • Актуелни програмски јазичи 	<p>Активности</p> <p>Наставникот започнува дискусија со учениците за повторување за концептите за програмирање и програмски јазик.</p> <p>Наставникот отвора дискусија за природни и вештачки јазичи, програмирање, програмски јазик.</p> <p>Учениците разгледуваат примери на програмски јазичи.</p> <p>Презентација на историјатот на програмските јазичи.</p>	<p>Ученикот/ученичката може да:</p> <ul style="list-style-type: none"> 1.1: објаснува и презентира за концептите за програмирање и програмски јазик; 1.2: препознава и опишува различни видови примери на програмски јазичи; 1.3: врши процена на популарноста на одредени програмски јазичи; 1.4: прави класификација на програмски јазичи според начин на обработка на податоци; 1.5: прави разлика меѓу структурен/ процедурален, објектно-ориентиран или функциски пристап во програмските јазичи;

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>објектно-ориентиран или функционален пристап; -да умее да ги наброи актуелните програмски јазици и истите да ги класификува според начинот на обработка на податоците.</p>	<p>Поими : Програмирање Програмски јазици Машински јазици Асембли /симболички јазици Виши програмски јазици Императивни јазици Декларативни јазици Процедурални јазици Објектно –ориентирани јазици Функционски јазици Логички јазици FORTRAN Lisp, Prolog, Basic, Pascal Cobol, C, C++, Java C#, Python</p>	<p>Презентација на различните поделби на програмските јазици; Разгледување на готови пример програми во различни програмски јазици со цел утврдување на типот на јазикот. Запознавање со актуелните програмски јазици. Истражување на популарноста на одредени програмски јазици во минатото и сега.</p> <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, решавање проблеми со алгоритам.</p>	<p>1.6: проценува и презентира програмски јазици според начин на обработка на податоци.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------

ОПЕРАЦИИ СО БИНАРНИ БРОЕВИ. ДЕТАЛЕН ПРЕГЛЕД НА ОСНОВНИТЕ ТИПОВИ НА ПОДАТОЦИ (5 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на:</p> <ul style="list-style-type: none"> -да знае да објасни што е бит, бајт, збор; -да знае како се претставуваат целите броеви во компјутерот за концептите за програмирање и програмски јазик; -да знае да изврши операции собирање, одземање и множење на два броја во бинарен броен систем; -да умее да наведе неколку различни типови на податоци; -да умее да направи конверзија на типови на податоци; да користи и употребува функциски библиотеки; -да применува бројни системи во ситуации од реалниот живот и во 	<p>Содржини</p> <p>Повторување за бит, бајт, бинарен броен систем, мерки за количество податоци</p> <p>Операции со бинарни броеви (собирање, одземање, множење, и како специјални: множење и делење со 2)</p> <p>Претставување на целите броеви во компјутер</p> <p>Опсег на броеви кои може да се претстават</p> <p>Нумерички типови (собирање вредности, константи, оператори, функциски библиотеки)</p> <p>Конверзија на типови (имплицитна, експлицитна)</p>	<p>Активности</p> <p>Наставникот започнува дискусија со учениците за запознавање со начините на претставување броеви, знаци, слики и звук во компјутер.</p> <p>Наставникот заедно со учениците развиваат дискусија за бројни системи, бинарен броен систем и презентација за конверзија на броеви од бинарен во декаден броен систем и обратно, како и специфична конверзија на броеви во следните бројни системи:бинарен <-> октален <->хексадекаден.</p>	<p>Ученикот/ученичката може да:</p> <ul style="list-style-type: none"> 1.1: прави разлика меѓу бит, бајт, збор; 1.2: врши конверзија на целите броеви од еден во друг броен систем (бинарен, октален, хексадекаден); 1.3: применува операции со бинарни броеви; 1.4: прави разлика меѓу константи и оператори; 1.5: објасни за претставување на целите броеви во компјутер; 1.6: имплементира конверзија на типови во програмски код; 1.7 решава конкретни проблемски задачи кои вклучуваат знаења од бројни системи.

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>решавање на специфични проблемски задачи.</p>	<p>Знаковни типови Специфични проблемски задачи кои вклучуваат знаења од бројни системи</p> <p>Поими: Бинарен броен систем, декаден броен систем, октален броен систем, хексадекаден броен систем, бајт, бит, конверзија, тип на податок</p>	<p>Учениците изработуваат вежби за совладување на операција собирање, одземање и множење на бинарни броеви. Повторување за бит и бајт; Презентација на мерки за количество податоци. Дискусија за примена на бинарниот броен систем и запознавање со еден начин на претставување на целите броеви во компјутер. Презентација и објаснување за функционални библиотеки. Претставување на некои општи типови на податоци.</p> <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми.</p>	
--	--------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

ПРОБЛЕМСКИ ЗАДАЧИ КОИ СЕ РЕШАВААТ СО ОСНОВНИ УПРАВУВАЧКИ СТРУКТУРИ (9 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на :</p> <p>-да решава специфични проблемски задачи кои се решаваат со примена на основните управувачки структури, а во себе вклучуваат алгоритми како:</p> <p>одредување на максимум и минимум на дадена серија од броеви;</p> <p>манипулација со датуми;</p> <p>припадност на точка во круг и во правоаголник;</p> <p>манипулирање со серија податоци (внесување, печатење) и креирање на статистички податоци;</p> <p>наоѓање просек, збир, и производ на дадена серија од броеви;</p>	<p>Содржини</p> <p>Основни управувачки структури</p> <p>Претставување формат на управувачки структури;</p> <p>Примери за користење на управувачки структури;</p> <p>Решавање на задачи со целобројна аритметика;</p> <p>Работа со збир на броеви и просечен успех;</p> <p>Припадност на точка во круг и правоаголник;</p> <p>Одредување на цифри на даден број;</p> <p>Работа со датуми и нивно прикажување.</p> <p>Поими:</p>	<p>Активности</p> <p>Наставникот започнува дискусија со учениците за основни управувачки структури.</p> <p>Наставникот објаснува конкретни елементарни алгоритми.</p> <p>Учениците изработуваат имплементација на изучените конкретни алгоритми.</p> <p>Учениците практично демонстрираат работа со серија на податоци, наоѓање на цифри за даден број и слично, како и имплементација на истите во задачи.</p> <p>Учениците изработуваат програми кои решаваат</p>	<p>Ученикот/ученичката може да:</p> <p>1.1: утврди потреба од основни управувачки структури кај одредени алгоритми;</p> <p>1.2: објасни и примени алгоритамски програмски решенија со некои основни алгоритми (сума на броеви, максимум/минимум, аритметичка средина, издвојување на цифри и сл.) во манипулација со серија на податоци;</p> <p>1.3: изработува програми со примена на изучените алгоритми;</p> <p>1.4: применува управувачки структури за наоѓање на решение на зададени операции на даден број или серија од броеви;</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>-издвојување на цифрите на даден број; -проверка дали даден број е прост; одредување НЗД, НЗС .</p>	<p>основни управувачки структури, серија на податоци и нивна обработка</p>	<p>текстуални проблемски задачи со примена на соодветните концепти од темата.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со имплементирање програмски решенија.</p>	<p>1.5: дизајнира структури за проблем на припадност на точка во круг и во правоаголник; 1.6: имплементира алгоритми за пронаоѓање прости броеви, делење на броеви на едноставни целини, пресметување на НЗД и НЗС; 1.7 решава текстуални проблеми со примена на изучените алгоритми.</p>
--	----------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ПРОБЛЕМСКИ ЗАДАЧИ КОИ СЕ РЕШАВААТ СО НИЗИ (21 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на: -да препознае проблемски ситуации во кои треба да се примени низа; -да имплементира алгоритми кои прават елементарни манипулации со елементите од низа;</p>	<p>Содржини Изминување на низа и наоѓање на сума, просек Минимален/максимален елемент, нивна замена Палиндром во низа Периодичност на елементи Работа со стрингови во низа Трансформација на елементи</p>	<p>Активности Наставникот започнува дискусија за: - концептот на низи; - потребата за примена на низи; - итерација и изминување на елементите во дадена низа;</p>	<p>Ученикот/ученичката може да: 1.1: објасни потреба од користење на низа; 1.2: оперира со елементи во низа и креира програмски решенија; 1.3: манипулира со елементи во низа;</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>-да решава специфични проблемски задачи за кои е неопходно користење на структура од елементи – едноразмерна низа, а во себе вклучуваат алгоритми како:</p> <ul style="list-style-type: none"> сума на елементи, просек, одредување на мин/макс елемент во низа; проверка за периодичност на низа; утврдување дали даден стринг (низа од знаци) е подмножество на друг стринг; сортирање на елементите во низата; <p>-да дефинира и употребува матрица;</p> <p>да решава специфични проблемски задачи за кои е неопходно користење на структура од елементи - матрица, а во себе вклучуваат алгоритми како:</p>	<p>Додавање/бришење на елементи</p> <p>Операции над елементи во низа</p> <p>Примена на теорија на броеви (Ератостеновото сито, Паскалов триаголник);</p> <p>Решавање специфични проблемски задачи за кои е неопходно користење на структура од елементи – едноразмерна низа</p> <p>Дефинирање на матрица</p> <p>Елементи во матрица</p> <p>Итерација во матрица</p> <p>Операции над елементи во матрица</p> <p>Главна и споредна дијагонала во матрица</p> <p>Додавање/бришење редици и колони во матрица</p> <p>Промена на позиција на одредени елементи (пример минимум и максимум)</p> <p>Сортирање на елементи во матрица</p>	<p>Презентирање на примери од секојдневието и програми кои користат низи.</p> <p>Наставникот објаснува алгоритми за манипулација со елементи од низа.</p> <p>Учениците изработуваат задачи со трансформација на низи и наоѓање на мин/макс елемент во низа, палиндром, замена на вредности.</p> <p>Наставникот започнува дискусија за:</p> <ul style="list-style-type: none"> - концептот на матрици; - потребата за примена на матрица; - итерација и изминување на елементите во дадена матрица. <p>Презентирање на примери од секојдневието и на</p>	<p>1.4. презентира карактеристики и предности на низа;</p> <p>1.5: решава проблеми со примена на низи;</p> <p>1.6: креира низа со различни операции;</p> <p>1.7: решава алгоритми со примена на теорија на броеви (Ератостеновото сито, Паскалов триаголник);</p> <p>1.8: модифицира и повторно организира елементи во низа;</p> <p>1.9: структурира задача со наоѓање на палиндром и најдолг палиндром;</p> <p>1.10: креира алгоритам за сортирање елементи на низа;</p> <p>1.11: генерира алгоритми и функции за работа со стрингови;</p> <p>1.12: опишува матрица и нејзина структура;</p> <p>1.13: оперира и манипулира со елементи во матрица;</p> <p>1.14: креира програмски решенија со матрици;</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>одредување минимален/максимален елемент во матрица; наоѓање на просек по редици и колони; креирање спирална матрица; трансформација на матрица (додавање/бришење редици и колони, бришење на дупликати, симетрични поместувања и сортирање по редици и колони).</p>	<p>Примена на стрингови кај матрица Наоѓање на својства на елементи и примена во програмски решенија Решавање специфични проблемски задачи за кои е неопходно користење на структура од елементи - матрица</p> <p>Поими: структура на податоци, низа, позиција на елементи, итерација, замена на елементи, периодичност на елементи, палиндром, стринг, матрица, ред/колона, трансформација на матрица</p>	<p>пример програми кои користат матрици. Наставникот објаснува алгоритми за манипулација со елементи од матрица. Учениците изработуваат задачи со трансформација на матрици и наоѓање на мин/макс елемент во матрица, палиндром, замена на вредности. Учениците изработуваат програми кои решаваат текстуални проблемски задачи со примена на соодветните концепти од темата.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми со низи и матрици</p>	<p>1.15: решава задачи со користење на повеќе од една низа / матрица; 1.16: имплементира додавање елементи од низа во матрица; 1.17: решава проблеми со користење на матрици; 1.18: имплементира алгоритми за испитување својства на елементи во матрица; 1.19: користи итерација на елементи во матрица по дијагонала; 1.20: модифицира матрица според одредени критериуми.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ФУНКЦИСКИ ШАБЛОНИ, РЕКУРЗИЈА И РЕКУРЗИВНИ ФУНКЦИИ (12 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на:</p> <ul style="list-style-type: none"> -да го објасни поимот функциски шаблон; -да ги користи предностите од повикување на функции и за препознавање на проблемски ситуации во кои треба да се примени функциски шаблон; -да решава проблеми со примена на функции; за правилно пренесување на вредности во функции преку параметар и преку референца; -да препознава и реши проблем со обременување на функции; -да решава проблеми со примена на функции без 	<p>Содржини</p> <ul style="list-style-type: none"> Потсетување за примена на функции, функции од математичка библиотека, функции со различен број на параметри Потсетување за пренесување на вредност преку вредносен и референцен параметар Проблемски задачи со примена на функции Обременување на функции Функциски шаблони Рекурзија и рекурзивни функции Примери за користење на рекурзија 	<p>Активности</p> <ul style="list-style-type: none"> Наставникот започнува дискусија за потсетување на претходно изучените концепти за функции преку изработка на соодветни програми. Дискусија на концептот за функциски шаблони и нивна поврзаност со обременети функции. Дискусија за случаите во кои можат да се применуваат функциски шаблони. Демонстрација на примери за функциски шаблон (пример: функциски шаблон за максимум). Дискусија за дефинирање на функциски шаблони со посебен осврт на 	<p>Ученикот/ученичката може да:</p> <ul style="list-style-type: none"> 1.1: опишува, дефинира и декларира функциски шаблон; 1.2: повикува функциски шаблон во главната функција; 1.3: изработува програми со функциски шаблони; 1.4: користи функциски шаблони за решавање на претходно работени задачи и ги применува во решенија со функција; 1.5: опишува, дефинира и декларира рекурзивна функција; 1.6: повикува рекурзивна функција во главната функција;

	<p>параметри и функции со повеќе параметри; да го објасни поимот за рекурзивна функција; -да ја опише функцијата на основниот чекор во рекурзијата; -да ја објасни функцијата на рекурзивниот чекор во рекурзијата; -да ги идентификува условите кои треба да важат за да терминира рекурзивна функција; правилно да користи рекурзивни функции; -да решава проблеми со рекурзивни функции; -да ги наведе сличностите и разликите меѓу итерација и рекурзија; -да ја демонстрира мемориската сложеност на рекурзивните функции.</p>	<p>Рекурзија наспроти итерација</p> <p>Поими: Шаблон (темплејт) Шаблонска параметарска листа Формален типизиран параметар Рекурзија Рекурзивна функција Основен чекор или терминален чекор Рекурзивен повик или рекурзивен чекор</p>	<p>специфицирањето на податочните типови во шаблонот. Дискусија за инстанците во фамилијата на функции кои се дефинираат со даден функциски шаблон преку примери (на пример да се разгледаат сите функции кои се дефинирани со шаблон за максимум ако податочните типови можат да бидат int, double или float). Објаснување на концептот на рекурзија. Дефинирање на рекурзивни функции. Учеиците разгледуваат и решаваат некои едноставни рекурзивни функции (пресметка на факториел, степен, фибоначиев број...).</p> <p>Дискусија за основниот случај во рекурзивна функција.</p>	<p>1.7: изработува програми со користење на рекурзивни повици; 1.8: користи рекурзивни повици за решавање на претходно работени задачи и ги применува во решенија со рекурзија; 1.9: анализира рекурзивни чекори и нивна функција; 1.10: споредува итерација и рекурзија и дава примери; 1.11: изработува програми со употреба и имплементација на функциски шаблон и/или рекурзија; 1.12: креира програми кои користат рекурзивна функција и функции со и без параметри.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

			<p>Дискусија за разделување на проблемот во два концептуални делови - дел кој програмата знае да го реши и дел кој не знае како да го реши.</p> <p>Дискусија за правилата кои треба да важат за да терминира една рекурзивна функција.</p> <p>Учениците изработуваат рекурзивни функции.</p> <p>Учениците практично претставуваат програми кои искористуваат рекурзивни функции.</p> <p>Дискусија за сличностите и разликите на итерацијата и рекурзијата.</p> <p>Дискусија за мемориската сложеност на рекурзивните функции.</p> <p>Учениците изработуваат програми кои решаваат текстуални проблемски задачи со примена на</p>	
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

			соодветните концепти од темата. Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми преку потпрограми.	
--	--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

ПОКАЖУВАЧИ (7 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	Ученикот/ученичката ќе биде способен/на: -да го објаснува концептот на покажувачите и потребата од нивно користење; -да декларира и иницијализира покажувачка променлива од различен тип и да „оживее“ покажувач со new;	Содржини Декларација и иницијализација на покажувачка променлива од одреден тип Резервирање на меморија со помош на покажувач (new) Оператори за покажувачи	Активности Дискусија за сместување во меморијата на статичките и динамичките променливи и потребата од постоење покажувачи. Објаснување за директно и индиректно референцирање променлива преку	Ученикот/ученичката може да: 1.1: дефинира и опишува покажувач; 1.2: прави разлика помеѓу адресен оператор & и оператор за дереференцирање*; 1.3: дефинира и селектира покажувачи во програмски решенија;

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>-да користи оператори за покажувачи; -да користи референца преку покажувач кај функции; -да ја објасни поврзаноста помеѓу покажувачите и низите; -да објаснува аритметички операции кои може да се изведуваат со покажувачи; -да ја објасни разликата помеѓу стандардната аритметика и аритметиката со покажувачи; -да објасни дека покажувачите кои покажуваат кон низи може -да се индексираат исто како и имињата на низите; -да дефинира низи од покажувачки променливи; за практична примена на стекнатите знаења за покажувачи преку изработка на програми.</p>	<p>Креирање низа во динамичка меморија со помош на покажувач Поврзаност меѓу покажувачи и низи Адресна аритметика Пример програма со низа од покажувачи</p> <p>Поими: Покажувач (покажувачка променлива) Директно и индиректно референцирање променлива Нулти покажувач Оператори за покажувачи Адресен оператор Оператор за индирекција (оператор за дереференцирање) Операторот new</p>	<p>наведување примери и графички презентации. Презентација на едноставни пример програми кои користат покажувачи. Објаснување на оператори за покажувачи преку графички презентации и едноставни пример програми. Објаснување за чести грешки при програмирање кога се користи дереференцирање на покажувач (дереференцирање променлива која не е покажувач, дереференцирање на неиницијализиран покажувач, дереференцирање на нулти покажувач). Дискусија, споредба и истакнување разлики помеѓу трите начина за</p>	<p>1.4: креира едноставни програми користејќи покажувачи и оператори за покажувачи; 1.5 креира низа во динамичка меморија со помош на покажувач; 1.6: изработува програми со правилно користење на адресна аритметика; 1.7: споредува и анализира стандардна аритметика и аритметика со покажувачи; 1.8: изработува програми со правилно користење на стандардна и адресна аритметика; 1.9: решава задачи со поврзаност помеѓу покажувач и низа; 1.10: комбинира покажувачи и креира програми со низи; 1.11: планира и организира програми со покажувачи за инкрементирање, декрементирање и</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>Инкрементирање, декрементирање на покажувач, додавање и одземање целобројна вредност, додавање и одземање еден покажувач од друг</p>	<p>предавање аргументи на функција во C++ - предавање по вредност, по референца со аргументи – референци, по референца со аргументи – покажувачи. Дискусија за креирање низа во динамичка меморија со помош на покажувач и изработка на програми кои го вклучуваат тоа. Дискусија за тесната поврзаност меѓу покажувачите и низите; Објаснување преку споредба на аритметички операции во стандардната аритметика и аритметиката со покажувачи. Дискусија и анализа на пример програми кои содржат аритметика со покажувачи: (инкрементирање (++), декрементирање на покажувач (--), додавање (+ или +=) и одземање (- или -</p>	<p>аритметички операции со покажувачи.</p>
--	--	-----------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------

			<p>=)целобројна вредност на/од покажувач, додавање и одземање еден покажувач на/од друг).</p> <p>Дискусија околу чести грешки при програмирање кога се користи аритметиката со покажувачи (логички грешки кои се појавуваат при: користење на аритметика со покажувачи на покажувач кој не покажува кон низа, одземање или споредба на два покажувачи кои не се однесуваат на елементи од иста низа, користење на аритметика со покажувачи за да се помести покажувач надвор од границите на некоја низа).</p> <p>Изработка на едноставни програми кои вклучуваат адресна аритметика.</p> <p>Дискусија и објаснување на примери преку кои треба да</p>	
--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

			<p>се согледа дека сите изрази со индексирани низи може да се запишат со покажувач и поместување, користејќи го или името на низата како покажувач или користејќи посебен покажувач кој покажува кон низата.</p> <p>Објаснување и дискусија за нотацијата покажувач/поместување;</p> <p>Објаснување како покажувачите се користат за пристап до елементи на низа (нотација со индексирани низа, нотација покажувач/поместување со името на низата како покажувач, нотација со индексирани покажувач, нотација покажувач/поместување со покажувач).</p> <p>Изработка на едноставни програми (пр. печатење на пет елементи од целобројна низа со користење на</p>	
--	--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

			<p>четирите нотации за пристап до елементите на низа. Изработка на програми со користење покажувачи.</p> <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми со користење на покажувачи и оператори за покажувачи.</p>	
--	--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

ПОДАТОЦИ ОД ТИП СТРУКТУРА (STRUCT) (5 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	Ученикот/ученичката ќе биде способен/на: -да дефинира, декларира и иницијализира податочен тип структура;	Содржини Дефинирање на податоци од тип структура	Активности Наставникот започнува дискусија со учениците и објаснува за структурирани типови податоци (низи,	Ученикот/ученичката може да: 1.1: опишува работа со податочен тип структура;

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>-да ја објасни потребата од користење на податочниот тип структура; -да декларира покажувач кон податок од тип структура; -да користи податоци од тип структура (само со податочни членки) во програми.</p>	<p>Декларирање и иницијализација на променливи од даден тип структура Покажувачи кон податоци од тип структура Функции-членки на структурата Изработка на програми</p> <p>Поими: Структура како тип на податок, struct Членови на структура Анонимна структура Променливи од соодветен тип структура Оператор за дофат</p>	<p>стрингови) и поврзување на претходните знаења со новиот податочен тип структура. Дискусија и давање примери за примената на податочниот тип структура. Дефинирање на новиот податочен тип – struct. Дискусија и објаснување за потребата за уникатност на имињата на членовите од иста структура, но не и од различни структури. Дефинирање на променливи од даден тип структура. Дискусија и објаснување за пристап до членовите на структурата преку презентирање примери. Учениците решаваат примери за структура и воочување на честа грешка при програмирање (синтаксичка) – заборавање на точка запирка на крајот</p>	<p>1.2: презентира карактеристики на податочен тип структура; 1.3: употребува низа при изработка на задачи со податочен тип структура; 1.4: објаснува што е функција-членка на структурата; 1.5: креира нов податочен тип структура; 1.6: селектира и организира податоци од тип структура за решавање проблемски задачи; 1.7: изработува задачи со пример структура како тип на податок.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

			<p>од дефиницијата на структурата. Објаснување пристап до членови на структура преку покажувач. Учениците изработуваат едноставни програми со примена на стекнатите знаења за податочен тип структура.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер, решавање на едноставни програми со пример податочен тип структура.</p>	
--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

ПОДАТОЧНИ СТРУКТУРИ (18 часа)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
	Ученикот/ученичката ќе биде способен/на:	Содржини	Активности	Ученикот/ученичката може да:

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

<p>1.</p>	<p>-да го објасни поимот динамичка структура на податоци; -да дефинира самореференцирачки тип структура; -да го објасни поимот линеарна поврзана листа и поимот еднострано поврзана линеарна листа; -да препознае јазол (елемент на еднострана листа) и да ја објасни неговата структура; -да креира подалгоритми за основните операции со еднострано поврзана линеарна листа; -да го објасни поимот куп; -да опише подалгоритми за основните операции со куп; -да го објасни поимот ред и -да направи разлика од куп; -да го објасни поимот дрво како нелинеарна листа; -да креира програми со примена на еднострано</p>	<p>структури на податоци Самореференцирачки тип структура Динамичка алокација на меморија и податочни структури Линеарни поврзани листи Еднострани поврзани линеарни листи (креирање и манипулација со елементите) Куп (анг. stack) и основни операции кај куп (креирање и манипулација со елементите) Запознавање со ред (анг. queue) и основни операции кај ред наспроти куп Нелинеарни наспроти линеарни податочни структури (пример: дрво)</p>	<p>Наставникот започнува дискусија со учениците за поврзување на претходно стекнатите знаења за типови податоци со новите поими. Објаснување и дискусија за статички и динамички податочни структури. Објаснување преку примери за самореференцирачки тип структура. Дискусија за динамичка алокација на меморија и податочни структури. Објаснување за јазол и за структура на еднострано поврзана линеарна листа преку графички прикази. Креирање и анализа на подалгоритми за операции со еднострано поврзани линеарни листи: креирање празна еднострано поврзана линеарна листа, додавање јазол на</p>	<p>1.1: опишува податочни структури и образложува за потребата од вклучување во дадени програмски решенија; 1.2: користи статички и динамички структури на податоци во програми; 1.3: користи самореференцирачки тип структура за креирање листи; 1.4: креира подалгоритми за основните операции со еднострано поврзана линеарна листа; 1.5: објаснува за куп и ред како податочни структури; 1.6: решава задачи со имплементација на куп и ред како податочни структури; 1.7: креира подалгоритми за основните операции кај куп и ред;</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>поврзана листа и куп како типови на податочни структури; -да ги познава и користи имплементациите од стандардната библиотека на шаблони (STL) на C++ за куп и ред при решавање на текстуални проблемски задачи</p>	<p>Изработка на креирање на програми со примена на погоренаведените типови на податочни структури Запознавање со имплементацијата на изучените структури во стандардната библиотека на шаблони (STL) на C++ Решавање на проблемски задачи кои користат податочни структури реализирани во стандардната библиотека на шаблони (6-8 часа)</p> <p>Поими: Куп Врв на куп Избивање елемент од куп Набивање елемент во куп</p>	<p>почетокот/крајот на еднострано поврзана линеарна листа, додавање јазол помеѓу два јазли во еднострано поврзана линеарна листа, бришење јазол од еднострано поврзана линеарна листа. Објаснување и дискусија за куп преку графички приказ на куп со неколку елементи. Креирање и анализа на подалгоритми за основните операции (иницијализација на празен куп, проверка дали купот е полн, ставање податок во куп, проверка дали купот е празен и земање податок од куп) и за имплементација на куп. Објаснување и дискусија за ред преку графички приказ на ред со неколку елементи. Презентација на подалгоритми за основните операции (иницијализација на празен ред, проверка дали</p>	<p>1.8: креира пример програми во кои се применуваат погоренаведените типови на податочни структури; 1.9: имплементира програмски решенија со изучените структури и алгоритми.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>Ред Опашка и глава на редот Покажувач на почетокот/крајот на редот Кружен ред Дрво</p>	<p>редот е полн, ставање податок на крајот од редот, проверка дали редот е празен и земање податок од почетокот на редот) и за имплементација на ред. Објаснување и дискусија преку шематски приказ за елемент (јазол) на дрво. Учениците изработуваат програми кои решаваат текстуални проблемски задачи со примена на соодветните концепти од темата.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање на програми со примена на податочни структури за примери од реалноста</p>	
--	--	-----------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

ОПШТИ ТЕХНИКИ ЗА КОНСТРУКЦИЈА НА АЛГОРИТМИ (27 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1.	<p>Ученикот/ученичката ќе биде способен/на:</p> <ul style="list-style-type: none"> -да ја примени техниката рекурзија при решавање на задачи со наоѓање на сума, производ, степен, факториел; проблем на Ханојски кули, манипулирање со елементи од низа; -да решава задачи со користење на низи при рекурзивна имплементација за даден програмски код: наоѓање на збир на елементи, мин/макс, печатење на избрани елементи...; -да опише пребарување со поврат (backtracking); 	<p>Содржини</p> <ul style="list-style-type: none"> Рекурзија и рекурзивни задачи Сума, производ, степен и факториел со рекурзивна постапка Ханојски кули со рекурзивна постапка Збир, мин/макс, пребарување и сортирање на низа со рекурзивна постапка Рекурзија и Евклидов алгоритам Фибоначи имплементација со низа Пребарување со поврат (backtracking) Алгоритми за груба сила со Судоку 	<p>Активности</p> <ul style="list-style-type: none"> Наставникот започнува дискусија со учениците и прави потсетување за претходни техники како рекурзија и рекурзивни повици. Учениците практично решаваат примери и задачи со рекурзија. Наставникот објаснува за имплементација на Ханојски кули со рекурзивна постапка со особен осврт на сложеноста на алгоритмот. Демонстрација и решение на задачи за наоѓање на збир, мин/макс, пребарување и сортирање на низа со рекурзивна постапка. 	<p>Ученикот/ученичката може да:</p> <ul style="list-style-type: none"> 1.1: објасни примена на рекурзија и рекурзивни повици во задачи; 1.2: изработи решенија за сума и производ на први N броеви, степен и факториел со рекурзивна постапка; 1.3: применува рекурзија во алгоритми за систематско броење и пребарување; 1.4: реализира алгоритми со рекурзивна имплементација; 1.5: објасни и имплементира алгоритми за сортирање со спојување и брзо сортирање; 1.6: примени пребарување со поврат во програми;

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>-да користи стратегија „Раздели, па владеј!“ при имплементирање на програмски решенија;</p> <p>-да имплементира алгоритми за сортирање со спојување и брзо сортирање;</p> <p>-да прави разлика меѓу алчни алгоритми и пристап на динамичко програмирање;</p> <p>-да реши задачи со изминување на графови по широчина и длабочина;</p> <p>-да реши задачи чии решенија вклучуваат изминување на графови.</p>	<p>Алгоритми за груба сила со магичен и латински квадрат</p> <p>Алгоритам за сортирање со спојување</p> <p>Алгоритам за брзо сортирање</p> <p>Вовед во алчни алгоритми и динамичко програмирање</p> <p>Потсетување за графови и нивни презентации</p> <p>Задачи со изминување на графови по широчина и длабочина</p> <p>Поими:</p> <p>Рекурзија и алгоритми</p> <p>Пребарување со поврат</p> <p>Алгоритми со груба сила</p> <p>Раздели па владеј</p> <p>Алгоритам за сортирање со спојување (Merge Sort)</p>	<p>Имплементација на Евклидов алгоритам со рекурзивна постапка.</p> <p>Креирање алгоритам за Фибоначи со рекурзија</p> <p>Дискусија и решавање на задачи со пребарување со поврат (backtracking).</p> <p>Практична демонстрација на алгоритми за груба сила со магичен и латински квадрат</p> <p>Примери за алгоритам за груба сила со низа</p> <p>Дискусија за пример со стратегијата „Раздели, па владеј!“.</p> <p>Учениците практично имплементираат алгоритми за сортирање со спојување и брзо сортирање.</p> <p>Наставникот ги изложува концептите на алчни алгоритми и динамичко програмирање и изложува елементарни примери.</p> <p>Учениците практично имплементираат алгоритми</p>	<p>1.7: идентификува и анализира чекори за алгоритми за груба сила;</p> <p>1.8: објасни за алчни алгоритми</p> <p>1.9: дизајнира и креира решенија со примена на динамичко програмирање;</p> <p>1.10: имплементира програмски решенија со изучените алгоритми.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>Алгоритам за брзо сортирање (Quick Sort) Алчни алгоритми Динамичко програмирање Графови и дрва</p>	<p>кои имаат алчен пристап и алгоритми со динамичко програмирање. Учениците имплементираат задачи со изминување на графови по широчина и длабочина. Учениците изработуваат програми кои решаваат текстуални проблемски задачи со примена на соодветните концепти од темата.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање на програми</p>	
--	--	-------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

<p>Оценување на постигањата на учениците</p>	<p>Во текот на наставата по програмирање се препорачува формативно следење кое вклучува изработка и водење портфолио на учениците што опфаќа:</p> <ul style="list-style-type: none"> - собирање показатели (ученички изработки на компјутер) за секој ученик поединечно; - тековни (формативни) однапред подготвени евалвациони листи за секој ученик посебно кои се изработуваат по конкретната негова активност. Исто така, се следи и интересот на ученикот кон работата, соработката со останатие ученици и истрајноста во извршувањето на задачите. <p>На крајот на секое тримесечје, врз основа на сознанијата од формативното оценување, се реализира микросумативното оценување.</p> <p>Согласно природата на програмата по предметот програмирање оценувањето може да се реализира усно, практично, со презентација и слично.</p> <p>Ученикот се оценува со бројчана оценка.</p> <p>Наставникот, според своето согледување, може да го проверува знаењето со усни одговори на учениците, со тестови според модуларните единици, домашни задачи и друго.</p>
<p>Литература и други извори</p>	<ul style="list-style-type: none"> - Учебник по прогмирање избран на ниво на училиштето и одобрен од страна на министерот за образование и наука; - Збирки задачи одобрени од страна на министерот за образование и наука; - Интернет, образовни софтвери и ресурси; - интегрирана околина за програмирање и демо програми; - аудио-визуелни средства.
<p>Почеток на имплементација на наставната програма</p>	<p>Учебна 2021/2022 година</p>

Институција/ носител на програмата	Биро за развој на образованието (БРО)
Потпис и датум на донесување на наставната програма	бр. _____ _____ година <p style="text-align: right;">МИНИСТЕРКА, Мила Царовска</p> <hr style="width: 20%; margin-left: auto; margin-right: 0;"/>
Датум на ревизија	