

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 и 64/18), и член 3 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 **програмирање** за I (прва) година математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма

ПРОГРАМИРАЊЕ

за I година

Математичко-информатичка гимназија

Скопје, 2019 година

Назив на наставната програма	Програмирање
Тип на наставна програма	Задолжителна
Кредитна вредност на наставната програма	7 (седум) ЕЦВЕТ ¹ кредити (5+2, 2 кредити 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	I (прва)
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 постигне самодоверба во примената на стекнатите програмерски и алгоритамски вештини за наоѓање, користење и презентирање на математичките аргументи; - да ја цени убавината, моќта, корисноста и интернационалната димензија на информатиката и програмирањето и да извлекува задоволство од постигнатите резултати; - да развива логичко, критичко и креативно алгоритамско размислување.
Теми/подрачја/модуларни единици на наставната програма	<ul style="list-style-type: none"> • ОСНОВИ НА ПРОГРАМИРАЊЕ • ОСНОВНИ УПРАВУВАЧКИ СТРУКТУРИ • ОСНОВНА СЛОЖЕНОСТ НА АЛГОРИТМИ

¹Закон за Националната рамка на квалификации.

	<ul style="list-style-type: none"> • ПОТПРОГРАМИ (ФУНКЦИИ) ВО C++ • НИЗИ • ДАТОТЕКИ. ТЕКСТУАЛНИ ДАТОТЕКИ • ПРЕТСТАВУВАЊЕ ПОДАТОЦИ ВО КОМПЈУТЕР. ОПШТИ ТИПОВИ ПОДАТОЦИ • КЛАСИ И ОБЈЕКТИ • СТРУКТУРИ НА ПОДАТОЦИ И СТАНДАРДНА БИБЛИОТЕКА НА ШАБЛОНИ • АЛГОРИТМИ КАЈ НИЗИ И ГРАФОВИ
Материјално-технички и просторни услови	<p>За постигнување на целите на наставата по програмирање неопходна 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>
Норматив за наставен кадар	<p>Наставната програма за I година може да ја реализира:</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 на престижен меѓународен натпревар од соодветната област; - да е запишан на докторски студии соодветната област; - да има стекнато научен степен на доктор на науки на соодветната област.

ОСНОВИ НА ПРОГРАМИРАЊЕ (14 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да ги објаснува поимите алгоритам и програма; - да ги препознае основните концепти на алгоритамското размислување; - да ја осознае примената на програмите во компјутерите и секојдневниот живот. 	<ul style="list-style-type: none"> • Алгоритми и нивно претставување (1 час) • Поим за програма како дел од софтверот и улога на програмите во компјутерот (1 час) <p>Поими: алгоритам, програма</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за алгоритмите и разгледување на реални секојдневни алгоритми. • Учениците креираат едноставни алгоритми со ситуации од секојдневниот живот и од другите наставни предмети: вртење телефонски број, правење торта, одење во училиште и слично. • Учениците изработуваат практична вежба за да се увидат предностите од подредени (сортирани) хартиени документи (според азбучен редослед, на пример) наспроти неподредени. • Наставникот започнува дискусија со учениците за 	<p>Ученикот/ученичката:</p> <p>1.1: може да објаснува за поимот алгоритам и негово претставување;</p> <p>1.2: може да препознава и креира различни видови на алгоритми за ситуации од секојдневниот живот и од другите наставни предмети;</p> <p>1.3: може да објаснува за поимот програма и нејзината примена во компјутерите и секојдневниот живот.</p>

			<p>програмите како неопходен дел од компјутерот.</p> <p>Методи: дискусија, дијалог, демонстрација, учење преку откривање-истражување, решавање проблеми со алгоритам.</p>	
2	<ul style="list-style-type: none"> - да го сфати концептот на програмирање; - да објаснува за различните типови на програмски јазици; - да објаснува за процесот на пишување и извршување на една програма; - ги познава и правилно да ги именува и користи елементите од интегрираната околина за програмирање (едитор, компајлер, дебагер (покажувач на грешки и помош)); - да извршува готов точен програмски код; - да ја осознае функцијата на дебагерот при извршување на неточен програмски код; - да ги препознава и да ги разликува датотеките кои се креираат при извршување и зачувување на програмата. 	<ul style="list-style-type: none"> • Програмирање и програмски јазици (1 час) • Процес на изработка на една програма (од изворна до извршна верзија) (1 час) • Запознавање со основните елементи на интегрирана околина за програмирање (1 час) • Изглед на готови пример програмски кодови и нивно извршување и датотеки кои притоа се продуцираат, дебагирање (1 час) <p>Поими: програмирање, програмски јазик, преведувач, програмер, интегрирана околина за програмирање,</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за природни и вештачки јазици, програмирање, програмски јазик, примери на програмски јазици. • Наставникот објаснува за процесот на преведување на програма, дискусија за преведувач, примери програмски јазици, програмер. • Наставникот објаснува и започнува дискусија за основните елементи на интегрирана околина за програмирање. • Наставникот објаснува за елементите на една програма напишана во програмскиот 	<p>2.1: Може да објаснува за поимите: програмски јазик, типови на програмски јазици, преведувач, изворна и извршна програма;</p> <p>2.2: може да именува и користи елементи од интегрираната околина за програмирање;</p> <p>2.3: може да извршува готов точен програмски код, да ги препознава и да ги разликува датотеките кои се креираат при извршување и зачувување на програмата;</p> <p>2.4: може да објаснува за функцијата на дебагерот при извршување на неточен програмски код.</p>

		изворна програма, извршна програма, дебагирање	<p>јазик (ПЈ), преку пример програма.</p> <ul style="list-style-type: none"> • Учениците практично на компјутер извршуваат готови пример програми. • Учениците практично на компјутер извршуваат пример програми кои содржат грешка со прикажување на функцијата на дебагерот. • Наставникот започнува дискусија и објаснува за синтакса и семантика. • Учениците ги разгледуваат креираните датотеки по извршување на програма. <p>Методи: дискусија, дијалог, демонстрација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми.</p>	
3	<ul style="list-style-type: none"> - Да ги објаснува основните елементи од програмскиот јазик (ПЈ); - да ја користи техниката за приказ на екран; - да ја користи техниката на редоследно извршување на 	<ul style="list-style-type: none"> • Основни елементи на програмскиот јазик (ПЈ) (азбука на јазикот, градбени делови, коментари, величини, идентификатори, искази, клучни зборови, 	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за основните елементи на програмскиот јазик - само начелно споменување на посложените. 	<p>3.1: Може да објаснува за основните елементи на програмскиот јазик;</p> <p>3.2: може да ја применува техниката за приказ на екран;</p> <p>3.3: може да изработува едноставни програми со редоследна структура.</p>

	<p>секвенца од искази (напластени искази, напластување);</p> <p>- да изработува едноставна програма со редоследна структура.</p>	<p>потпрограми), исказ за приказ на екран (1 час)</p> <ul style="list-style-type: none"> • Техника на редоследно извршување и изработка на програми (1 час) <p>Поими:</p> <p>коментар, величина, идентификатор, исказ (наредба), клучен збор, оператор, потпрограма, приказ на екран, секвенца од искази, редоследно извршување</p>	<ul style="list-style-type: none"> • Наставникот објаснува за поимот исказ со пример преку исказ за приказ на екран. • Наставникот презентира неколку програми со прикази на екран. • Учениците изработуваат програми со прикази на екран. • Наставникот започнува дискусија со учениците за техниката за секвенца од изрази и редоследно извршување. • Наставникот презентира пример програми, а учениците изработуваат програми каде повеќе искази за приказ на екран се напластени (се извршуваат редоследно). <p>Методи: дискусија, дијалог, демонстрација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми.</p>	
--	--	---	---	--

4	<ul style="list-style-type: none"> - Да го применува концептот за аритметички операции и изрази; - да го објаснува поимот за променлива во програмски јазик; - да ја применува техниката за доделување на вредност; - да користи променливи и искази за доделување во програмскиот јазик; - да препознава различни типови на променливи; - да изработува програми со изучените техники. 	<ul style="list-style-type: none"> • Аритметички операции, изрази, величини, константи и променливи во ПЈ (1 час) • Доделување вредност на променлива. Оператор за доделување и изработка на програми (1 час) <p>Поими: аритметички операции, изрази, константа, променлива, тип на променлива, оператор за доделување</p>	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за аритметичките операции, изрази, величини во ПЈ (константи и променливи). • Демонстрација на наставникот за доделување константни вредности на променлива, доделување вредност на аритметички израз и доделување вредност на друга променлива. • Наставникот започнува дискусија со учениците за тип на променлива преку наведување на два најчесто користени типови на променлива (на пр. целобројна и реална променлива). • Учениците изработуваат програми со вклучување на доделувања, аритметички пресметки и прикази на екран (напластени искази). 	<p>4.1: Може да објаснува за променливи и типови на променливи во програмскиот јазик;</p> <p>4.2: може да применува константи, променливи и изрази со аритметички операции при изработка на програма;</p> <p>4.3: може да изработува програми со изучените техники;</p>
---	---	---	--	---

			<p>Методи: дискусија, дијалог, демонстрација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p>	
5	<ul style="list-style-type: none"> - Да ги применува техниките за внесување податоци во програмата; - да ја применува техниката за објаснувања за податоците кои се очекуваат од корисникот; - да користи дополнителни специфики на јазикот. 	<ul style="list-style-type: none"> • Искази (техники) за внесување податоци во програмата (1 час) • Техника за објаснувања за податоците кои се очекуваат од корисникот (1 час) • Дополнителни специфики од ПЈ (1 час) • Изработка на поголем број програми со досега изучените техники (1 час) <p>Поими: внесување податоци</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за исказите за внесување податоци во програмата и техника за објаснувања за податоците преку искази за приказ на екран пред секој исказ за внесување податоци. • Наставникот презентира пример на програми. • Учениците изработуваат програми со досега изучените техники. • Наставникот објаснува за конкретни специфики за C++ преку дискусии и примери, а учениците изработуваат програми кои нив ги вклучуваат (пр. знаковен тип на променлива, форматирано печатење...). 	<p>5.1: Може да применува техники за внесување на податоци во програмата;</p> <p>5.2: може да применува техника за објаснувања за податоците кои се очекуваат од корисникот;</p> <p>5.3: може да изработува програми со изучените техники.</p>

			<p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p>	
--	--	--	--	--

ОСНОВНИ УПРАВУВАЧКИ СТРУКТУРИ (16 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да го распознава концептот на споредбен израз ($a > b$, $a \geq b, \dots$); - да конструира едноставни споредбени изрази (услови); - да конструира логички изрази (сложени услови) преку конјункција, дисјункција и негација на споредбени изрази (услови); - да ја користи точно синтаксата и семантиката на структурата за избор од две можности; 	<ul style="list-style-type: none"> • Споредбени и логички изрази (1 час) • Структура (исказ) за избор од две можности (1 час) • Блок од искази (наредби) и техника на вгнездување на искази (1 час) • Изработка на програми (1 час) <p>Поими: споредбен израз (услов), логички израз,</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за споредбени изрази (услови). Вредност на споредбен израз – точно/неточно, т.е. условот е исполнет/не е исполнет. • Наставникот започнува дискусија со учениците за логички изрази (сложени услови) добиени преку конјункција, дисјункција и 	<p>Ученикот/ученичката:</p> <p>1.1: може да конструира едноставни споредбени изрази (услови). 1.2: може да конструира логички изрази (сложени услови) преку конјункција, дисјункција и негација на споредбени изрази (услови). 1.3: може да применува структура за избор од две можности во конструкција на програми</p>

	<ul style="list-style-type: none"> - да ја применува структурата за избор од две можности во конструкција на програми; - да го применува концептот за блок од искази (наредби); - да ја осознае техниката на вгнездување на искази и резултатите од истото при извршувањето на програмите; - да ја применува техниката на вгнездување на искази во конструкција на програми. 	<p>структура за избор од две можности, блок од искази (блок, сложен исказ), техника на вгнездување</p>	<p>негација на споредбени изрази.</p> <ul style="list-style-type: none"> • Учениците составуваат споредбени изрази и логички изрази врз база на дадени текстуални барања. • Наставникот објаснува за синтаксата и семантиката на структура (исказ) за избор од две можности. Презентација на пример програми. Презентација на примери од секојдневието. • Учениците изработуваат програми со структура за избор од две можности. • Дискусија за блок од искази (блок, сложен исказ). Блок како дел од структура за избор од две можности. • Дискусија за структура за избор од две можности како дел од блок. • Наставникот објаснува за примената на техниката на вгнездување искази; • Учениците изработуваат низа од програми со техниката за вгнездување (со 	<p>1.4: може да применува техника на вгнездување на искази во конструкција на програми.</p>
--	--	--	---	---

			<p>вклучување на сите од досега изучените искази).</p> <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p>	
2	<ul style="list-style-type: none"> - Да ја примени точно синтаксата и семантиката на структурата за избор од повеќе можности; - да ја применува структурата за избор од повеќе можности во конструкција на програми; - да изработува програми со досега изучените техники. 	<ul style="list-style-type: none"> • Структура за избор од повеќе можности (1 час) • Изработка на програми со сите претходно изучени техники (1 час) <p>Поими: структура за избор од повеќе можности</p>	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија за синтаксата и семантиката на структура (исказ) за избор од повеќе можности со презентација на пример програми од секојдневието. • Учениците изработуваат програми: <ul style="list-style-type: none"> - со структура за избор од повеќе можности; - вклучување на останатите изучени искази во програмите преку техниките на напластување и вгнездување. <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување,</p>	<p>2.1: Може да применува структура за избор од повеќе можности во конструкција на програми;</p> <p>2.2: може да изработува програми со досега изучените техники</p>

			практична работа на компјутер, решавање проблеми преку програми	
3	<ul style="list-style-type: none"> - Да ја примени точно синтаксата и семантиката на структура за повторување на циклус до исполнување на услов; - да ја примени структурата за повторување на циклус до исполнување на услов во конструкција на програми; - да изработува програми со досега изучените техники. 	<ul style="list-style-type: none"> • Структура за повторување на циклус до исполнување на услов (споредбен израз) (1 час) • Изработка на програми (3 часа) <p>Поими: циклус, структура за повторување на циклус до исполнување на услов</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за: <ul style="list-style-type: none"> - концептот на циклус (јамка) во една програма; - синтаксата и семантиката на структура (исказ) за повторување на циклус до исполнување на услов. • Презентација на пример програми, дел од нив со визуелизација и презентација на примери од секојдневието. • Учениците изработуваат: <ul style="list-style-type: none"> - програми со структура за повторување на циклус до исполнување на услов; - програми со вклучување на останатите изучени искази преку техниките на напластување и вгнездување 	<p>3.1: Може да применува структура за повторување на циклус до исполнување на услов во конструкција на програми;</p> <p>3.2: може да изработува програми со досега изучените техники на напластување и вгнездување.</p>

			<p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p>	
4	<ul style="list-style-type: none"> - Да ја примени точно синтаксата и семантиката на структурата за повторување на циклус со броење на циклусите; - да ја примени структурата за повторување на циклус со броење на циклусите во конструкција на програмите; - да ја примени точно синтаксата и семантиката на друга структура за повторување; - да изработува програми со досега изучените техники. 	<ul style="list-style-type: none"> • Структура за повторување на циклус со броење на циклусите (1 час) • Дополнителни структури за повторување (1 час) • Изработка на програми (1 час) <p>Поими: структура за повторување на циклус со броење на циклусите</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за: <ul style="list-style-type: none"> - синтаксата и семантиката на структура (исказ) за повторување на циклус со броење на циклусите. • Презентација на пример програми и презентација на примери од секојдневието. • Учениците изработуваат: <ul style="list-style-type: none"> - програми со структура за повторување на циклус со броење на циклусите; - програми со вклучување на останатите изучени искази преку техниките на напластување и вгнездување. • Презентација на други структури за повторување на циклуси. 	<p>4.1: Може да применува структура за повторување на циклус со броење на циклусите во конструкција на програми; 4.2: може да објаснува друга структура за повторување; 4.3: може да изработува програми со досега изучените техники на напластување и вгнездување.</p>

			<p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p>	
5	<ul style="list-style-type: none"> - Да применува алгоритми за решавање дадени проблеми; - да ги програмира изучените алгоритми; - да изработува програми со сите изучени техники. 	<ul style="list-style-type: none"> • Познати алгоритми за решавање на некои проблеми (1 час) • Изработка на програми (2 часа) 	<p>Активности:</p> <ul style="list-style-type: none"> • Презентација на познати алгоритми во зависност од сите претходно изучени техники (на пример, алгоритам за НЗД, пресметување на n-ти Фибоначиев број). • Учениците изработуваат предизвикувачки програми (проект-програми) кои ги вклучуваат сите изучени техники. <p>Методи: дискусија, дијалог, презентација, практична работа на компјутер, решавање проблеми преку програми</p>	<p>5.1: Може да применува одредени познати алгоритми при решавање на проблеми;</p> <p>5.2: може да изработува проект-програма со вклучување на сите изучени техники.</p>

ОСНОВНА СЛОЖЕНОСТ НА АЛГОРИТМИ (4 часа)

Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да го примени точно концептот за временска и просторна сложеност на алгоритмите; - да применува некои алгоритми за решавање на дадени проблеми (сума на броеви, минимум/максимум од множество броеви, аритметичка средина на броеви, издвојување на цифри на број и нивна манипулација); - да програмира со изучените алгоритми, и за нивно препознавање во зададени текстуални проблеми. 	<ul style="list-style-type: none"> • Поим за временска и просторна сложеност на алгоритмите (1 час) • Претставување примери за алгоритми со различна временска сложеност (1 час) • Текстуални задачи и утврдување на временски и просторни ограничувања (1 час) • Изработка на програми кои решаваат текстуални проблеми (1 час) <p>Поими: сложеност на алгоритам, временска сложеност, просторна сложеност</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за алгоритмите и за концептот на временска и просторна сложеност на алгоритам • Наставникот објаснува за концептите на временска и просторна сложеност на алгоритмите преку примери за алгоритми со различна сложеност кои решаваат исти проблем. • Учениците изработуваат програми кои решаваат текстуални проблеми. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична</p>	<p>Ученикот/ученичката:</p> <p>1.1: може да ја објасни и утврдува временската и просторната сложеност на одредени алгоритми;</p> <p>1.2: може да објасни и примени некои основни алгоритми (сума на броеви, максимум/минимум, аритметичка средина, издвојување на цифри и сл.) во решавање на текстуални проблеми;</p> <p>1.3: може да изработува програми со примена на изучените алгоритми.</p>

			работа на компјутер со решавање текстуални проблеми преку програми	
--	--	--	--	--

ПОТПРОГРАМИ (ФУНКЦИИ) ВО C++ (6 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да го објаснува поимот потпрограма; - да осознае дека и главната програма е потпрограма во C++; - да ги разликува функциите на потпрограма и на главна програма; - да ја осознае потребата и предностите од користење (повикување) на потпрограми во главната програма; - да препознава проблемски ситуации во кои треба да се примени потпрограма; - да ги дефинира и применува основните математички 	<ul style="list-style-type: none"> • Модули (1 час) • Концепт „Раздели па владеј“ при решавање на посложен проблем, пристап „од врвот-надолу“ (1 час) • Потпрограми (т.е. функции во C++), имплементација и примена на функции (1 час) • Примери на готови функции од <code>cmath</code> библиотеката во C++ (1 час) • Предавање на вредности во функција преку параметри (1 час) • Изработка на функции и нивно искористување во програми (1 час) 	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија за: <ul style="list-style-type: none"> - концептот за потпрограма. - потребата за примена на потпрограми; - концептот на разделување на проблемот на помали и поедноставни, полесно решливи проблеми. • Презентирање примери од секојдневието и пример програми кои користат потпрограми. • Наставникот започнува дискусија за: <ul style="list-style-type: none"> - синтаксата на заглавишната линија на потпрограмата. 	<p>Ученикот/ученичката:</p> <p>1.1: може да објаснува за поимот потпрограма, видови потпрограми и нивната структура;</p> <p>1.2: може да препознава проблемски ситуации во кои треба да се примени потпрограма;</p> <p>1.3: може да дефинира и применува основните математички функции од библиотеката <code>cmath</code> (<code>fabs</code>, <code>sqrt</code>, <code>pow</code>, <code>ceil</code>, <code>floor</code>, <code>sin</code>, <code>cos</code>);</p> <p>1.4: може да решава проблеми со примена на потпрограми и да ги повикува од главната програма.</p>

	<p>функции од библиотеката smath (fabs, sqrt, pow, ceil, floor, sin, cos);</p> <ul style="list-style-type: none"> - да ги применува правилата за именување на потпрограма; - да изработи потпрограми и да ги повикува од главната програма; - за правилно пренесување на вредности од главната програма во потпрограмата; - да решава проблеми со примена на потпрограми. 	<p>Поими: модул, потпрограма, функција библиотека на функции, прототип на функција, параметар, повик на функција</p>	<ul style="list-style-type: none"> - за повикот на потпрограмата и предавањето на параметри. • Учениците изработуваат потпрограми (дел од нив се претходно изработени програми) и програми кои искористуваат потпрограми преку повик. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми преку програми-потпрограми</p>	
--	---	---	--	--

НИЗИ (22 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да објаснува и дефинира структура на податоци и 	<ul style="list-style-type: none"> • Структура на податоци – еднодимензионална низа (во C-стил), дефинирање, декларирање и 	Активности	Ученикот/ученичката:

	<p>еднодимензионална низа како структура од податоци;</p> <ul style="list-style-type: none"> - да ја утврди потребата од користење на структура на податоци – еднодимензионална низа; - да декларира, дефинира и користи еднодимензионална низа; - да пристапува до член од еднодимензионална низа; - да користи повторување со броење на циклусите при достапување до сите елементи од низата (поминување низ низа); - да изработува програми со користење на еднодимензионални низи; - да користи алгоритми за сортирање на еднодимензионална низа; - да анализира временска сложеност на алгоритми за сортирање на еднодимензионална низа; - да користи алгоритми за 	<p>иницијализација на еднодимензионална низа (1 час)</p> <ul style="list-style-type: none"> • Пристапување и операции со членовите на еднодимензионална низа (1 час) • Програми со поминување низ еднодимензионална низа (2 час) • Работа со повеќе низи (1 час) • Сортирање низа по растечки и опаѓачки редослед (осврт на временска сложеност) (1 час) • Пребарување во низа (осврт на временска сложеност) (1 час) • Предавање низи на функции (1 час) • Креирање на еднодимензионални низи од неколку објекти (1 час) • Креирање на класи кои содржат еднодимензионални 	<ul style="list-style-type: none"> • Наставникот започнува дискусија и објаснува за: <ul style="list-style-type: none"> - структури на податоци; - еднодимензионална низа; - декларирање и дефинирање на еднодимензионална низа преку примери со програмски код; - индекс на елемент во низа. • Учениците практично изработуваат програми: <ul style="list-style-type: none"> - со еднодимензионални низи (основни операции со низи, пресметување просечна вредност, наоѓање на минималната/максималната вредност во низа, сортирање на низа, пребарување во низа); - со предавање на еднодимензионални низи на функции (на пр. функција која го враќа збирот на елементи на низа, читање/печатење на низа...); - со повеќе низи кои решаваат ситуации од секојдневиот живот; 	<p>1.1: може да објаснува, дефинира и декларира еднодимензионална низа;</p> <p>1.2: може да пристапува до член во еднодимензионална низа со користење на циклус- броење во циклус;</p> <p>1.3: може да изработува програми со еднодимензионални низи;</p> <p>1.4: може да користи алгоритми за сортирање еднодимензионална низа и пребарување во низа;</p> <p>1.5: може да изработува програми со функции чии аргументи се еднодимензионални низи;</p> <p>1.6: може да креира програми со класи кои содржат еднодимензионални низи како податочни членки;</p> <p>1.7: може да креира програми со класниот шаблон <vector> од стандардната библиотека на C++.</p>
--	---	---	--	--

	<p>пребарување на еднодимензионална низа;</p> <ul style="list-style-type: none"> - да изработува програми со функции чии аргументи се еднодимензионални низи; - да креира програми во кои се користат еднодимензионални низи од неколку објекти; - да креира класи кои содржат еднодимензионални низи како податочни членки - да го применува класниот шаблон <vector> од стандардната библиотека на C++ како замена за еднодимензионална низа во C-стил. 	<p>низи како податочни членки (1 час)</p> <ul style="list-style-type: none"> • Класен шаблон <vector> од стандардната библиотека на C++ (2 часа) <p>Поими: структура на податоци, еднодимензионална низа, член на низа, индекс на член на низа, поминување низ низа, сортирање на низа, пребарување во низа, временска сложеност, низа од објекти</p>	<ul style="list-style-type: none"> - кои вклучуваат еднодимензионални низи од објекти од дадена класа (пр. наоѓање просечна оценка за даден предмет за дадена низа од ученици...); -кои вклучуваат изработка на класа која има еднодимензионални низи за податочна членка (пр. класа ученик, податочна членка низа оценки од различни предмети...); - кои вклучуваат употреба на класен шаблон <vector> наместо стандардни еднодимензионални низи. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми преку програми</p>	
2	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да утврди дека текстуална низа (низа од знаци) е вид на 	<ul style="list-style-type: none"> • Текстуална низа (низа од знаци) како еднодимензионална низа, декларирање / 	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија и објаснува за: -текстуална низа како еден 	<p>Ученикот/ученичката:</p> <p>2.1: Може да објаснува, дефинира и декларира текстуална низа;</p>

	<p>еднодименсионална низа</p> <ul style="list-style-type: none"> - да дефинира и декларира текстуални низи; - да ја користи логичката организација на низа од знаци; - да ги користи основните програмски функции за работа со низи од знаци; - да напише алгоритми и програми за задачи со текстуални низи, со користење на изучените функции за работа со текстуални низи; - да напише алгоритми и програми за задачи со текстуални низи како објекти од класата <code>string</code>. 	<p>дефинирање на текстуална низа (1 час)</p> <ul style="list-style-type: none"> • Функции за работа со текстуални низи (1 часа) • Запознавање со класата <code>string</code> и можноста за примена при работа со низа од знаци (2 часа) • Програми со текстуални низи (2 часа) <p>Поими: текстуална низа, елементи на текстуална низа, индекс на член на текстуална низа, функции за текстуална низа, поминување низ текстуална низа, Класа <code>string</code></p>	<p>вид на еднодименсионална низа;</p> <ul style="list-style-type: none"> - декларирање и дефинирање текстуална низа преку примери со програмски код • Учениците практично изработуваат програми: <ul style="list-style-type: none"> - со текстуални низи (на пр., колку пати се појавува буква во реченица, замена на мали со големи букви, пребројување на зборови во реченица, проверка дали збор е палиндром, составување на реченица од дадени зборови...) - со текстуални низи како објекти од класата <code>string</code> - кои решаваат ситуации од секојдневниот живот <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми преку програми</p>	<p>2.2: Може да ги користи основните програмски функции за работа со низи од знаци при изработка на програми;</p> <p>2.3: Може да напише алгоритам и програма за задачи со примена на текстуални низи;</p> <p>2.4: Може да напише алгоритам и програма за задачи со текстуални низи како објекти од класата <code>string</code>.</p>
--	--	--	---	--

3	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да дефинира и опишува дводимензионална низа – матрица како систем на организација на податоците; - да дефинира и опишува начин на пристапување до секој од елементите на матрицата; - да разликува двоиндексна променлива, индекс, вредност на двоиндексна променлива; - да креира и практично да реализира внесување и печатење на матрици; - да разликува и оперира со ред, колона и елемент од матрица; - да креира и практично да реализира програми за основните операции со матрици; - да креира и практично да реализира програми за едноставни проблеми врзани со матрици. 	<ul style="list-style-type: none"> • Повеќедимензионални низи – дводимензионални низи (1 час) • Дефинирање, декларирање и иницијализација на дводимензионални низи (матрици) (1 час) • Креирање и печатење на матрица, изминување низ матрица (1 час) • Операции со ред, колона и елементи од матрица (1 час) <p>Поими: дводимензионална низа – матрица, член на матрица, индекс на член на матрица, ред, колони, дијагонала</p>	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија за: <ul style="list-style-type: none"> - повеќедимензионални низи и матрици; - декларирање и дефинирање на матрица преку примери со програмски код; - за индексите на член во матрица; - ред, колона и дијагонал во матрица. • Презентирање примери од секојдневието каде што се вклучени матрици. • Учениците изработуваат програми: <ul style="list-style-type: none"> - со едноставни читања/печатења на матрица; - матрици (собирање /множење на елементите, собирање на елементи од дијагоналата, пребројување на позитивни -негативни, парни/непарни... членови во матрица); 	<p>Ученикот/ученичката:</p> <p>3.1: може да објаснува, дефинира и декларира матрица;</p> <p>3.2: може да напише програма за задачи со ред, колона и дијагонала кај матрица;</p> <p>3.3: може да напише програма за едноставни проблеми поврзани со матрица.</p>
---	---	---	---	--

			<p>-кои решаваат ситуации од секојдневниот живот.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање проблеми преку програми</p>	
--	--	--	--	--

ДАТОТЕКИ. ТЕКСТУАЛНИ ДАТОТЕКИ (5 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да дефинира и опишува датотека; - да направи разлика помеѓу датотеки со случаен пристап и текстуални датотеки; - да дефинира и опишува стандардни процедури и функции за работа со текстуални датотеки; - да креира и практично да реализира едноставни 	<ul style="list-style-type: none"> • Датотеки и видови датотеки: (датотеки со случаен пристап, датотеки со секвенцијален пристап - текстуални датотеки, поврзување на поимот за датотека со физичка датотека на хард дискот) (1 час) • Променливи за работа со текстуални датотеки (1 час) • Функции за работа со текстуални датотеки (1 час) 	<p>Активности</p> <ul style="list-style-type: none"> • Презентација на: <ul style="list-style-type: none"> -примери од секојдневието каде се применуваат датотеки; - пример програми кои користат датотеки. • Наставникот започнува дискусија со учениците: <ul style="list-style-type: none"> - каде се сместени датотеките во компјутер и како може текстуалните 	<p>Ученикот/ученичката може да:</p> <p>1.1: дефинира и опишува датотека</p> <p>1.2: прави разлика помеѓу датотеки со случаен пристап и текстуални датотеки</p> <p>1.3: дефинира и користи стандардни процедури и функции за работа со текстуални датотеки</p> <p>1.4: креира едноставни програми користејќи текстуални датотеки и тоа: креирање на датотека, приказ на содржината, префрлање на</p>

	<p>програми користејќи текстуални датотеки;</p> <p>- да креира и практично да реализира програми за едноставни операции со текстуални датотеки како: креирање, приказ на содржината од датотеката на екран, префрлање на податоци од една во друга текстуална датотека.</p>	<ul style="list-style-type: none"> • Креирање текстуална датотека преку програма и директно преку некој текстуален едитор (1 час) • Запишување податоци во текстуална датотека и читање на податоци од текстуална датотека (1 час) <p>Поими:</p> <p>датотеки , случаен пристап, секвенцијален пристап, текстуални датотеки, променливи за работа со текстуални датотеки, функции за работа со текстуални датотеки</p>	<p>датотеки да се креираат со текстуален едитор;</p> <p>- синтаксата на функциите (наредбите) коишто се користат во датотеките.</p> <ul style="list-style-type: none"> • Учениците изработуваат програми: <ul style="list-style-type: none"> - во кои се вклучени за едноставни операции со текстуални датотеки како: креирање, приказ на содржината од датотеката на екран, префрлање податоци од една во друга текстуална датотека; - во кои влезот ќе се чита од претходно постоечка текстуална датотека, а излезот ќе се запишува во ново креирана текстуална датотека. <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми со користење на датотеки</p>	<p>податоци од една во друга датотека</p>
--	---	---	---	---

ПРЕТСТАВУВАЊЕ НА ПОДАТОЦИ ВО КОМПЈУТЕР. ОПШТИ ТИПОВИ НА ПОДАТОЦИ (6 часа)

Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да објасни за начините на претставување броеви, знаци, слики и звук во компјутер; - да објасни што е позиционен броен систем; - да разликува бинарен, октален, декаден и хексадекаден систем; - да прави конверзија од еден во друг броен систем за конкретните бројни системи: бинарен <-> декаден и бинарен <-> октален <-> хексадекаден; 	<ul style="list-style-type: none"> • Општо претставување броеви, знаци, слики, звук (1 час) • Бројни системи - бинарен броен систем, декаден броен систем, октален и хексадекаден броен систем (1 час) • Конверзии помеѓу бројни системи (бинарен <-> декаден), (бинарен <-> октален <-> хексадекаден) (1 час) • Пример за операција со бинарни броеви – собирање (1 час) 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците и објаснува за: <ul style="list-style-type: none"> - начините на претставување броеви, знаци, слика и звук во компјутер; - бројни системи- бинарен броен систем; - примена на бинарен броен систем и начин на претставување на целите броеви во компјутерот; - претставување на некои општи типови на податоци. • Презентација за: 	<p>Ученикот/ученичката:</p> <p>1.1: може да објасни за начините на претставување на броеви, знаци, слики и звук во компјутер;</p> <p>1.2: може да разликува бинарен, октален и хексадекаден броен систем и извршува конверзии помеѓу овие бројни системи;</p> <p>1.3: може да изврши операција собирање на броеви во бинарен броен систем;</p> <p>1.4: може да објасни што е бит, бајт, збор и кое се мерките за количество на податоци;</p> <p>1.5: може да објасни како се претставуваат целите броеви во</p>

	<ul style="list-style-type: none"> - да изврши операција собирање на два броја во бинарен броен систем; - да објасни што е бит, бајт, збор и како се претставуваат целите броеви во компјутерот; - да наведе неколку различни типови на податоци. 	<ul style="list-style-type: none"> • Повторување за бит и бајт, мерки за количество податоци (1 час) • Претставување на целите броеви во компјутер, опсег на броеви кои може да се претстават и општи типови на податоци (1 час) <p>Поими: бинарен броен систем, декаден броен систем, октален броен систем, хексадекаден броен систем, бајт, бит, конверзија, тип на податок</p>	<ul style="list-style-type: none"> -конверзија на броеви од бинарен во декаден броен систем и обратно; -специфична конверзија на броеви и тоа бинарен<->октален <->хексадекаден броен систем; -операцијата собирање на бинарни броеви; -мерките за количество на податоци. • Учениците решаваат задачи за: <ul style="list-style-type: none"> -конверзија на броеви од бинарен во декаден броен систем и обратно; -конверзија на броеви од бинарен<->октален <->хексадекаден броен систем; - претставување на цели броеви во компјутерот; -собирање на броеви во бинарен броен систем. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, решавање на задачи</p>	<p>компјутерот и за опсегот на броеви кои може да се претстават.</p>
--	--	--	--	--

КЛАСИ И ОБЈЕКТИ (6 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да ги применува концептите на објектно ориентираното програмирање; - да ги распознава елементите на ООП (класи, објекти, функции-членки) и нивната функција; - да разликува објект од класа; - да ја примени синтаксата за креирање на класа во програма; - да ја објасни формата и функцијата на функциите за поставување и преземање; - да ги применува изучените концепти во конструкција на 	<ul style="list-style-type: none"> • Вовед во објектно ориентирано програмирање (ООП) (1 час) • Класи, објекти, функции – членки, податочни членки (1 час) • Дефинирање класа со функција членка (1 час) • Податочни членки, функции за поставување и преземање (1 час) • Изработка на програми кои содржат дефиниција на класа и примена на објекти – Инстанци од таа класа (2 часа) 	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците и објаснува за: <ul style="list-style-type: none"> - објектно – ориентираното програмирање преку примери од реалноста; - елементите на ООП (класи, објекти, функции-членки, податочни членки) и нивната функција. • Презентација за: <ul style="list-style-type: none"> - синтаксата за креирање класи; - формата и функцијата на функциите за поставување и преземање. • Учениците изработуваат програми со примена на 	<p>Ученикот/ученичката:</p> <p>1.1: може да објасни за елементите на ООП (класи, објекти, функции-членки, податочни членки) и нивната функција;</p> <p>1.2: може да објасни за формата и функцијата на функциите за поставување (set) и преземање (get);</p> <p>1.3: може да креира програми со примена на класи за примери од реалноста.</p>

	<p>класи и изработка на програми.</p>	<p>Поими: објектно ориентирано програмирање, класа, објект инстанца, функција-членка, податочна членка, функции-членки за поставување вредности (set), функции-членки за преземање вредности (get), интерфејс, имплементација</p>	<p>изучените концепти за конструкција на класи.</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање на програми со примена на класи за примери од реалноста</p>	
--	---------------------------------------	--	--	--

СТРУКТУРИ НА ПОДАТОЦИ И СТАНДАРДНА БИБЛИОТЕКА НА ШАБЛОНИ (15 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да објасни и да даде пример за прости и сложени структури од податоци; - да го објасни поимот динамичка структура на податоци; 	<ul style="list-style-type: none"> • Динамички структури на податоци (1 час) • Куп (анг. stack) и основни операции кај куп (креирање и манипулација со елементите) (2 часа) 	<p>Активности</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците и објаснува за: <ul style="list-style-type: none"> - стандардната библиотека на шаблони на C++; 	<p>Ученикот/ученичката:</p> <p>1.1: може да објасни за динамичката структура на податоци;</p> <p>1.2: може да објасни за поимот куп и изработи програми со основните операции за куп;</p>

<ul style="list-style-type: none"> - да го објасни поимот куп; - да креира подалгоритми за основните операции со куп; - да го објасни поимот ред и умее да направи разлика од куп; - да објасни за множество од структури и алгоритми (STL - Standard Template Library); - да ги објасни операциите кои можат да се вршат врз различни типови податоци; - да ги објасни составните делови на стандардната библиотека на шаблони: контејнерите, итераторите и алгоритмите; - да ги поврзе претходните знаења за класен шаблон <vector> од стандардната библиотека на C++; - да решава проблеми со употреба на класните шаблони и нивните методи; - да објасни пример за асоцијативен контејнер - мапа; - да решава програмски проблеми со употреба на мапа; - да објасни пример за 	<ul style="list-style-type: none"> • Запознавање со ред (анг. queue) и основни операции кај ред наспроти куп (2 часа) • Стандардната библиотека на шаблони: контејнери, итератори и алгоритми (1 час) • Контејнери и нивната поделба (секвенцијални, асоцијативни, адаптери) (1 час) • Креирање празен вектор, додавање елементи и печатење на содржина (1 час) • Примена на класен шаблон <vector> од стандардната библиотека на C++ како замена за еднодимензионална низа (2 часа) • Употреба на итератори (1 час) • Примена на асоцијативен контејнер мапа (map) (2 часа) • Примена на контејнер адаптер: куп (stack) (1 час) • Примена на контејнер адаптер: ред (queue) (1 час) 	<ul style="list-style-type: none"> - врската со претходно изучените типови на податоци со новите поими; - статички и динамички податочни структури; - куп преку графички приказ на куп со неколку елементи; - операции со куп: иницијализација на празен куп, проверка дали купот е полн, ставање податок во куп, проверка дали купот е празен и земање податок од куп; - ред преку графички приказ на ред со неколку елементи; - операции со ред: иницијализација на празен ред, проверка дали редот е полн, ставање податок на крајот од редот, проверка дали празен е празен и земање податок од почетокот на редот; - множеството структури од стандардната библиотека на шаблони на C++; 	<p>1.3: може да објасни за поимот ред и изработи програми со основните операции за ред;</p> <p>1.4: може да објасни за составните делови на стандардната библиотека на шаблони: контејнерите, итераторите и алгоритмите и поделба на контејнери;</p> <p>1.5: може да објасни за вектор и изработи програми со примена на вектор;</p> <p>1.6: може да објасни за мапа и да изработи програми со примена на мапа.</p>
--	---	--	---

	<p>контејнер адаптер - куп;</p> <ul style="list-style-type: none"> - да решава програмски проблеми со употреба на контејнер адаптер - куп; - да употреби правилен начин за користење елементи од множеството од структури и алгоритми (STL - Standard Template Library). 	<p>Поими:</p> <p>прости и сложени структури од податоци, статички и динамички структури на податоци, куп, врв на куп, набивање елемент од куп, ред, опашка и глава на редот, стандардната библиотека на шаблони, контејнери, итератори алгоритми, секвенцијални контејнери, асоцијативни контејнери, контејнери адаптери, vector, map, stack, push_back, front, back, size, reverse, sort, resize, clear, итератор, push, pop</p>	<ul style="list-style-type: none"> - множеството компоненти кои имплементираат голем број на познати податочни структури и алгоритми за обработка на тие структури. • Учениците практично изработуваат програми: <ul style="list-style-type: none"> - кои сместуваат податоци од ист и различен тип; - за пристап до елементите во даден вектор и употреба на push_back, front, back, size, reverse resize и clear - кои вклучуваат методи begin, end, insert и erase; - со примена на мапа; - со примена на куп; - со примена на ред. • Разгледување примери и задачи со итератори. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со решавање на програми</p>	
--	--	--	---	--

АЛГОРИТМИ КАЈ НИЗИ И ГРАФОВИ (14 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Стандарди за оценување
1	<p>Ученикот/ученичката ќе биде способен/способна:</p> <ul style="list-style-type: none"> - да распознае проблеми од секојдневниот живот кои вклучуваат пребарување низ податоци; - да го применува алгоритмот за бинарно пребарување; - да ги прецизира условите кои треба да ги задоволува низата за да може да се искористи алгоритмот за бинарно пребарување; - да прави разлика меѓу линеарно и бинарно пребарување; - да ги имплементира алгоритмите за линеарно и бинарно пребарување; - да ги објасни чекорите на алгоритмите за сортирање (сортирање со селекција и кофа – сортирање); - да распознае проблеми во кои се користат алгоритмите за 	<ul style="list-style-type: none"> • Потсетување за еднодимензионални низи и сортирани низи (1 час) • Алгоритми за пребарување-линеарно пребарување и неговата сложеност (1 час) • Бинарно пребарување и негова сложеност (1 час) • Сортирачки алгоритми - сортирање со селекција и негова сложеност (анг. selection sort) (1 час) • Кофа сортирање и негова сложеност (анг. bucket sort) (1 час) • Решавање проблеми кои користат сортирање и пребарување (2 часа) • Потсетување за дефинирање и декларирање, креирање и печатење на матрици, операции со елементите на матрица, работа со повеќе 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците, објаснува и презентира за: <ul style="list-style-type: none"> - проблеми од секојдневниот живот кои вклучуваат пребарување низ големо количество податоци (телефонски именик, речник и сл.). -Времето за извршување на алгоритмот за пребарување, односно бројот на споредби во зависност од големината на векторот. - Алгоритми за пребарување (пр. тестирање дали првиот елемент од низа е еднаков на некој друг елемент, тестирање дали постојат два исти елементи во низата и сл.). 	<p>Ученикот/ученичката:</p> <p>1.1: може да примени алгоритам за бинарно и линеарно пребарување во програми;</p> <p>1.2: може да објасни како работат алгоритмите за сортирање (сортирање со селекција и кофа – сортирање);</p> <p>1.3: може да прецизира услови и препознава проблеми во кои може да се користат алгоритмите за сортирање;</p> <p>1.4: може да изработи програми во кои ќе бидат вклучени операции со повеќе матрици;</p> <p>1.5: може да објасни за граф, видови на графови, основни елементи на графови и претставување на графови со матрица на соседство;</p> <p>1.6: може да решава едноставни програми кои вклучуваат графови;</p> <p>1.7: може да објасни за работата на алгоритмот за пребарување во</p>

<p>сортирање;</p> <ul style="list-style-type: none"> - да ги прецизира условите кои треба да ги задоволува низата за да може да се искористи одреден алгоритам за сортирање; - да ги имплементира алгоритмите за сортирање; - да ја препознае и да ја објасни сложеноста на алгоритмите за споредба и сортирање на податоци; - да реализира програми во кои ќе бидат вклучени операции со повеќе матрици; - да креира и практично да реализира програми за едноставни проблеми врзани со матрици; - да дефинира граф и основните поими за графови; - да прави разлика меѓу ориентиран и неориентиран граф; - да решава едноставни програми кои вклучуваат графови; - да ја објасни работата на алгоритмот за пребарување 	<p>матрици, напредно печатење на матрици (1 час)</p> <ul style="list-style-type: none"> • Графови- дефиниција и основни поими, ориентиран и неориентиран графови и сврзаност (1 час) • Претставување на графовите со помош на матрици (1 час) • Пребарување низ графови - пребарување прво по широчина (1 час) • Пребарување прво по длабочина (1 час) • Решавање на проблеми со графови (2 часа) <p>Поими: пребарување, пребарувачки клуч, сортирачки клуч, линеарно пребарување, бинарно пребарување, логаритамско време на извршувањ, сортирање со селекција, кофа сортирање, сортирање со спојување, дводимензионал-на низа (матрица), ред и колона кај</p>	<p>-Алгоритам за бинарно пребарување. - Условите кои треба да ги исполнува низата за да може да се користи бинарно пребарување. - Сложеноста на алгоритмот за бинарно пребарување. - Проблеми од секојдневниот живот кои вклучуваат сортирање на податоци (пр. сортирање по азбучен редослед во дневник и сл.). -Алгоритам за сортирање со селекција и неговата сложеност. -Алгоритам за кофа сортирање, условите кои треба да ги задоволува низата за да може да се примени кофа сортирање и сложеноста на овој алгоритам. -Графови, теме и ребро во граф, неориентиран и ориентиран графови.</p>	<p>граф прво по длабочина и ги препознава проблемите кои може да се решат со овој алгоритми; 1.8: објасни за работата на алгоритмот за пребарување во граф прво по широчина и ги препознава проблемите кои може да се решат со овие алгоритми; 1.9: може да ги имплементира алгоритмите за пребарување прво по широчина и пребарување прво по длабочина во граф.</p>
---	---	--	--

	<p>прво по широчина;</p> <ul style="list-style-type: none"> - да ја објасни работата на алгоритмот за пребарување прво по длабочина; - да ги распознава проблемите кои може да се решат со алгоритмите за пребарување прво по широчина и пребарување прво по длабочина; - ги имплементира алгоритмите да за пребарување прво по широчина и пребарување прво по длабочина. 	<p>матрица, граф, теме, ребро, соседни темиња, степен на теме, ориентиран граф, неориентиран граф, сврзаност, пребарување прво по широчина, пребарување прво по длабочина</p>	<ul style="list-style-type: none"> - Претставување на графовите со помош на матрици на соседството. - Алгоритмот за пребарување прво по широчина. - Проблеми кои се решаваат со алгоритмот за пребарување прво по широчина (најкратка оддалеченост меѓу две темиња и сврзаност на граф. -Алгоритмот за пребарување прво по длабочина. • Учениците практично изработуваат програми: <ul style="list-style-type: none"> -кои користат бинарно пребарување; - од секојдневниот живот кои користат сортирање; -кои применуваат напреден пристап до елементите од матриците и нивно печатење; - со кои се решаваат ситуации од секојдневниот 	
--	--	---	---	--

			<p>живот во кои се користат матрици. (пр. лавиринт); -кои вклучуваат графови (проверка дали две темиња се соседни, наоѓање на степен на теме, наоѓање на соседи на теме и сл.).</p> <ul style="list-style-type: none"> • Илустрација на решенија на проблеми: - кои го користат алгоритмот за пребарување прво по широчина (проблеми кои можат да се претстават со матрици како најкраток пат низ лавиринт и сл.); -кои го користат алгоритмот за пребарување прво по длабочина, Backtracking (преку проблеми кои можат да се претстават со матрици). <p>Методи: дискусија, дијалог, презентација, илустрација, учење преку откривање-истражување, практична</p>	
--	--	--	---	--

			работа на компјутер со решавање на програми	
--	--	--	--	--

<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 посебно; - тековни (формативни) однапред подготвени евалвациони листи за секој ученик посебно кои се изработуваат по конкретната негова активност. <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>	<p>Учебник по програмирање избран на ниво на училиштето и одобрен од страна на министерот за образование и наука;</p> <ul style="list-style-type: none"> - Интернет, образовни софтвери и ресурси; - интегрирана околина за програмирање и демо програми; - аудио-визуелни средства.

Почеток на имплементација на наставната програма	Учебна 2020/2021 година
Институција/ носител на програмата	Биро за развој на образованието (БРО)
Потпис и датум на донесување на наставната програма	бр. 13-12282/7 25.10.2019 година <div style="text-align: right;">МИНИСТЕР, Dr. ArbërAdemi</div> <hr/>
Датум на ревизија	