

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 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 предметот **програмирање** за III (трета) година математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма
ПРОГРАМИРАЊЕ
модуларно дизајнирана

за III година

Математичко-информатичка гимназија

Скопје, 2022 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	III (трета)
Број на часови неделно/годишно за реализација на наставната програма	3/108
Цели на наставна програма	<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> - да ги продлабочи знаењата по информатика во областа на програмирањето и решавањето на проблеми и да ги применува стекнатите знаења во секојдневни ситуации, во соодветни предизвици како и во други наставни предмети. - да постигне самодоверба во примена 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.

¹Закон за Националната рамка на квалификации.

<p>Модуларни единици на наставна програма</p>	<ul style="list-style-type: none"> • ИСЦРПЕН ПРЕГЛЕД НА ПОДАТОЧНИ СТРУКТУРИ И БИБЛИОТЕКИ (30 часа) • ОСНОВНИ ПАРАДИГМИ ЗА РЕШАВАЊЕ НА ПРОБЛЕМИ (22 ЧАСА) • АЛЧНИ АЛГОРИТМИ (15 ЧАСА) • ДИНАМИЧКО ПРОГРАМИРАЊЕ (24 ЧАСА) • ИЗБРАНИ АЛГОРИТМИ СО ГРАФОВИ (17 ЧАСА)
<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 односно:</p> <ul style="list-style-type: none"> - наставник со завршени студии по информатика/ наставна или друга насока VII/1 или VI A според НРК и 240 ЕКТС; <p>Стручното лице треба да 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; - научен степен доктор на информатички науки; - да е запишан на докторски студии на соодветната област. - да има стекнато научен степен доктор на науки на соодветната област.

Модуларна единица 1: ИСЦРПЕН ПРЕГЛЕД НА ПОДАТОЧНИ СТРУКТУРИ И БИБЛИОТЕКИ (30 часа)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а :</p> <ul style="list-style-type: none"> - да ги наброи различните податочни структури - да ја објасни примената на различни податочни структури - да ги имплементира различните податочни структури доколку за нив не постои вградена имплементација - да ги користи различните податочни структури при решавање на проблемски задачи 	<ul style="list-style-type: none"> • Улога на податочните структури (ПС) при дизајн на алгоритмите • Линеарни ПС со вградени библиотеки: <ul style="list-style-type: none"> - низи со фиксна и променлива големина, - битмаски - Поврзани ПС: поврзана листа, куп (stack), ред (queue), ред со два краја (deque), • Нелинеарни ПС со вградени библиотеки: <ul style="list-style-type: none"> - Бинарен стог (Binaryheap) - Хаш табела (hashtable) - Балансирани бинарни пребарувачки дрва • ПС со имплементација од корисникот <ul style="list-style-type: none"> - Графови 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за повторување за улогата на податочните структури (ПС) при дизајн на алгоритмите; • За секоја од наведените податочни структури <ul style="list-style-type: none"> - Наставникот ја објаснува структурата и нејзината имплементација - Во дискусија и преку бура на идеи се утврдуват можности за примена на дадената структура - Наставникот наведува пример случаи на примена на структурата - Учениците ја имплементираат ПС, доколку се работи за ПС со имплементација од корисникот 	<p>Ученикот/ученичката може да:</p> <ol style="list-style-type: none"> 1.1: идентификува и набројува различни програмски структури 1.2: презентира за примената на различни програмски структури 1.3: имплементира различни програмски структури 1.4: користи различни линеарни податочни структури со вградени библиотеки (низи, битмаски, листа, куп, ред) при решавање на проблемски задачи 1.5: применува различни нелинеарни податочни структури со вградени библиотеки (бинарен стог, хаш табела, бинарни пребарувачки дрва) при решавање на проблемски задачи 1.6: дизајнира и интегрира различни податочни структури (графови, унија на дисјунктни множества, сегментно дрво и бинарно индексирани дрва) при решавање на проблемски задачи

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

		<ul style="list-style-type: none"> - Барач на унија од дисјунктни множества (Union-find disjoint sets) - Сегментно дрво - Бинарно индексирано дрво <p>•Проблемски задачи кои се решаваат со изучените ПС</p> <p>Поими :</p> <p>Битмаска ,поврзана листа, куп (stack), ред (queue), ред со два краја (deque), бинарен стог (Binaryheap), хеш табела (hashtable), балансирано бинарно пребарувачко дрво, Граф, Барач на унија од дисјунктни множества (Union-find disjoint sets), Сегментно дрво, Бинарно индексирано дрво</p>	<ul style="list-style-type: none"> - Учениците решаваат избрани проблемски задачи со примена на изучените податочни структури <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, бура на идеи, практична работа на компјутер - решавање на проблемски задачи</p>	
--	--	--	---	--

Модуларна единица 2: ОСНОВНИ ПАРАДИГМИ ЗА РЕШАВАЊЕ НА ПРОБЛЕМИ (22 ЧАСА)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - да направи дистинкција помеѓу четирите парадигми за решавање на проблемски задачи - да препознава и предлага конкретен пристап за решавање на одреден проблем - решава задачи со исцрпно пребарување - решава конкретни задачи со пристап „раздели па владеј“ 	<ul style="list-style-type: none"> • Парадигма на исцрпно пребарување – метод на груба сила (5-6 часа) - Разни техники со итеративно исцрпно пребарување - Разни техники со рекурзивно исцрпно пребарување • Парадигма „раздели па владеј“ (1 час) • Пример примени на „раздели па владеј“ (8 -12 часа): - Брзо множење на полиноми - Распделување на интервали (Weighted Interval Scheduling), - Конвексна обвивка (Convex Hull), 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку презентација ги запознава учениците со карактеристиките на секоја од основните парадигми • Преку дискусија со учениците се утврдуваат сличности и разлики на основните парадигми • Наставникот наведува пример проблеми и бара учениците успешно да препознаат која парадигма е соодветна за конкретниот проблем • Учениците истражуваат за пример проблеми соодветни за конкретна парадигма • Наставникот презентира конкретни техники со исцрпно пребарување и 	<p>Ученикот/ученичката може да:</p> <p>1.1: прави разлика помеѓу четирите парадигми (парадигма на исцрпно пребарување, парадигма „раздели па владеј“, парадигма на алчен пристап и парадигма на динамичко програмирање) при нивна примена за решавање на проблемски задачи</p> <p>1.2: ги објасни пристапите со исцрпно пребарување и „раздели па владеј“</p> <p>1.3: препознае со каков пристап би се решил конкретен проблем</p> <p>1.4: решава задачи со примена на изучените техники за исцрпно пребарување (итеративно и рекурзивно исцрпно пребарување)</p> <p>1.5: решава задачи со примена на пристапот „раздели па владеј“</p>

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

		<ul style="list-style-type: none"> - Проблем на максимална подниза (Maximum Subarray Problem), - Наоѓање на најблиски парови • Парадигма на алчен пристап (1-2 часа) • Парадигма на динамичко програмирање (1-2 часа) <p>Поими: Исцрпно пребарување Груба сила Пристап „раздели па владеј“ Алчни алгоритми Динамичко програмирање</p>	<p>објаснува во кои ситуации има смисол и предност да се користи ваков пристап</p> <ul style="list-style-type: none"> • Учениците решаваат задачи со изучените техники за исцрпно пребарување • Наставникот презентира конкретни проблеми решливи со пристапот „раздели па владеј“ • Учениците имплементираат програмски решенија на проблеми решливи со пристап „раздели па владеј“ <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање на проблемски задачи</p>	
--	--	---	--	--

Модуларна единица 3: АЛЧНИ АЛГОРИТМИ (15 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - Дефинира парадигма на алчен пристап и да препознава проблеми кои се решаваат со алчен пристап - да решава конкретни проблеми со алчен пристап - да препознае дека одреден проблем не може да се реши со алчен пристап 	<ul style="list-style-type: none"> • Алчни алгоритми, како алгоритми за решавање на посебни оптимизациски проблеми - Својство на оптимална подструктура • Примери на алчни алгоритми кои доведуваат до оптимални решенија - Наоѓање на најшироко подмножество од активности кои не се преклопуваат - Проблем на наоѓање на минимално скелетно дрво (алгоритми на Прим и Крушкар) - Алчно бипартитно подударување (впарување) 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку презентација, со наведување на примери ги запознава учениците со карактеристиките на алчната парадигма • Преку дискусија со учениците се укажува на потребата проблемот да има својство на оптимална подструктура за да може да се примени алчен алгоритам • Наставникот наведува и објаснува решенија на пример проблеми со алчен пристап и ја дискутира нивната временска и просторна сложеност • Учениците имплементираат решенија за дадените пример проблеми 	<p>Ученикот/ученичката може да:</p> <p>1.1: презентира дефиниција за парадигма на алчен пристап 1.2: разликува дали даден проблем се решава или не со примена на парадигма на алчен пристап 1.3: реши конкретен проблем со примена на алчен пристап 1.4: применува алчни алгоритми (на пр. алгоритми на Прим и Крушкар, алгоритам на Дикстра, Хуфманов код, алчно бипартитно впарување и сл.) во решавање на задачи за добивање на оптимални решенија 1.5: применува алчни алгоритми кои работат во одредени специјални форми на даден проблем (проблем на ранец, минимален број на парички за дадена сума и сл.)</p>

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

		<ul style="list-style-type: none"> - Алчно балансирање на товар - Алчно покривање интервали - Хуфманов код и конструкција на хуфманово дрво - Алгоритам на Дикстра за најкраток пат во граф • Алчни алгоритми кои работат само во одредени специјални форми на даден проблем, или може да се користат како хевристика за одреден проблем - Минимален број на парички за дадена сума - Специјален проблем на ранец • Решавање на избрани проблемски задачи со парадигма на алчен пристап <p>Поими: Алчни алгоритми Оптимална подструктура</p>	<ul style="list-style-type: none"> • Учениците решаваат задачи кои се решаваат со алчен пристап • Учениците истражуваат за пример проблеми кај кои алчната стратегија нема да даде оптимално решение, дури може да даде и крајно лошо решение <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање на проблемски задачи</p>	
--	--	---	--	--

		Скелетно дрво Алгоритам на Прим Алгоритам на Крушкал Алгоритам на Дикстра Хуфманов код Проблем на ранец		
--	--	--	--	--

Модуларна единица 4: ДИНАМИЧКО ПРОГРАМИРАЊЕ (24 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	Ученикот/ученичката ќе биде способен/а: <ul style="list-style-type: none"> - да дефинира парадигма на динамичко програмирање и препознава проблеми кои се решаваат со ДП - да решава конкретни класични проблеми со ДП - да решава неklasични проблеми кои се решаваат со ДП 	<ul style="list-style-type: none"> • Техника на динамичко програмирање (ДП) - Проблем на Фибоначиева низа и потреба од меморирање на меѓурезултати • Пример на проблем за кој останатите парадигми не даваат точни и/или доволно брзи решенија • Пристап озгора надолу - Мемоизација • Пристап оздола нагоре 	Активности: <ul style="list-style-type: none"> • Наставникот преку презентација, со наведување примери ги запознава учениците со карактеристиките динамичкото програмирање • Преку дискусија со учениците се укажува на потребата проблемот да има својство на оптимална подструктура за да може да се примени динамичко програмирање и се 	Ученикот/ученичката може да: <p>1.1: презентира дефиниција за парадигма на динамичко програмирање</p> <p>1.2: идентификува дали одреден проблем се решава или не со примена на динамичко програмирање</p> <p>1.3: изработува решение за конкретен класичен проблем со примена на динамичко програмирање (со пристап озгора надолу, со пристап оздола нагоре)</p>

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

		<ul style="list-style-type: none"> - Рационализација на меморија • Компарација на пристапи: озгора надолу наспроти оздола нагоре • Испитување на оптимално решение (реконструкција на решение), наспроти само вредноста на оптималното решение • Стандардни проблеми кои се решаваат со пристап на ДП - Наоѓање консекутивна еднодимензионална подниза со најголем збир на елементите - Наоѓање консекутивна дводимензионална подниза со најголем збир на елементите (и споредба со други пристапи) - Класичен проблем на ранец 	<p>извлекува паралела со алчен алгоритам</p> <ul style="list-style-type: none"> • Наставникот наведува и објаснува решенија на пример проблеми со динамичко програмирање и ја дискутира нивната временска и просторна сложеност • Учениците имплементираат решенија за дадените пример проблеми • Учениците решаваат задачи кои се решаваат со динамичко програмирање • Учениците истражуваат за неklasични проблеми кои се решаваат со динамичко програмирање и прават нивна имплементација <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање на проблемски задачи</p>	<p>1.4: решава конкретен неklasичен проблем со примена на динамичко програмирање</p>
--	--	--	---	--

		<ul style="list-style-type: none">- Минимален број на парички за дадена сума во општ случај- Проблем на трговски патник- Алгоритам на Флојд-Варшал за најкратки патишта во граф• Решавање на избрани проблемски задачи со парадигма на ДП• Разгледување решенија и решавање на избрани неklasични проблемски задачи со парадигма на ДП <p>Поими: Динамичко програмирање Мемоизација Проблем на трговски патник Алгоритам на Флојд-Варшал</p>		
--	--	---	--	--

Модуларна единица 5: ИЗБРАНИ АЛГОРИТМИ СО ГРАФОВИ (17 ЧАСА)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - да решава задачи со изминување на графови по широчина и длабочина - да имплементира програми кои вклучуваат графови; - да ја објасни работата на алгоритмот Белман - Форд; - да распознава проблемите кои може да се решат со алгоритмите за пребарување и со алгоритми за наоѓање на најкраток пат; - да ги имплементира алгоритмите за наоѓање на најкраток пат на Белман-Форд, Дикстра и Флојд – Варшал 	<ul style="list-style-type: none"> • Потсетување за графови и нивни репрезентации • Изминување на графови по широчина (BFS) и длабочина (DFS) - Наоѓање на сврзани компоненти во ненасочен граф - Боење на графови (flood fill) - Тополошко сортирање кај насочен ацикличен граф - Проверка на бипартитност на граф - Проверка на состојба на ребро, при DFS на граф - Наоѓање на артикулациски точки и мостови во граф • Минимално скелетно дрво во граф и еквивалентен проблем на максимално скелетно дрво 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку дискусија прави навраќање кон претходно стекнатите знаења за графови • Наставникот наведува и објаснува решенија на пример проблеми со изминување на графови • Учениците имплементираат решенија за дадените пример проблеми • Наставникот започнува дискусија со учениците преку навраќање на претходно изучени алгоритми, го објаснува и презентира концептот на наоѓање на минимално скелетно дрво во граф и неговата еквивалентност со 	<p>Ученикот/ученичката може да:</p> <p>1.1: реши задачи со изминување на графови по широчина (BFS) и длабочина (DFS)</p> <p>1.2: ги објасни и применува алгоритмите на Белман-Форд, Дикстра и Флојд-Варшал за наоѓање на најкраток пат</p> <p>1.3: утврди кои проблеми може да се решат со алгоритмите за пребарување и со алгоритми за наоѓање на најкраток пат</p> <p>1.4: ја објасни потребата за наоѓање на минимално (максимално) скелетно дрво во граф</p> <p>1.5: да ги имплементира познатите алгоритми за наоѓање на минимално (максимално) скелетно дрво во граф</p>

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

		<ul style="list-style-type: none"> • Алгоритми за најкраток пат во граф - Алгоритам на Белман - Форд наспроти Алгоритам на Дикстра - Алгоритми со почеток во едно теме, наспроти алгоритми на патишта меѓу сите парови темиња • Решавање на избрани проблемски задачи со изучените алгоритми и концепти <p>Поими: Сврзани компоненти во граф, Боење на граф, Тополошко сортирање, Бипартитност на граф, Артикулациска точка и мост кај граф, алгоритам на Белман - Форд алгоритам на Дикстра скелетно дрво во граф</p>	<p>проблем на максимално скелетно дрво</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците преку навраќање на претходно изучени алгоритми, го објаснува и презентира концептот на наоѓање на најкраток пат во граф (од едно теме наспроти секое со секое теме) и го презентира алгоритмот на Белман-Форд; • Учениците решаваат задачи за најкраток пат во граф <p>Методи: дискусија, дијалог, презентација, илустрација, учење преку откривање-истражување, практична работа на компјутер - решавање на проблемски задачи</p>	<p>1.6: изработи програми кои ги вклучуваат изучените алгоритми и концепти за графови, при решавање на избрани проблемски задачи</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 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>	<ul style="list-style-type: none"> - Учебник по програмирање избран на ниво на училиштето и одобрен од страна на министерот за образование и наука; - Збирки задачи; - 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 2022/2023 година</p>
<p>Институција/ носител на програмата</p>	<p>Биро за развој на образованието(БРО)</p>

<p>Потпис и датум на донесување на наставната програма</p>	<p>бр. 13-7336/6 22.6.2022 година</p> <p style="text-align: right;">МИНИСТЕР, Doc.Dr. Jeton Shaqiri</p> <hr/>
<p>Датум на ревизија</p>	