

Врз основа на член 55 став 1 од Законот за организација и работа на органите на државната управа („Службен весник на Република Македонија“ бр. 58/00, 44/02, 82/08,167/10,51/11 и „Службен весник на Република Северна Македонија“ бр.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 („Службен весник на Република Македонија“ бр. 229/20) и член 3 од Законот за математичко-информатичка гимназија („Службен весник на Република Македонија“ бр. 64/18) министерот за образование и наука ја донесе Наставната програма по предметот **програмски парадигми за IV** (четврта) година математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма
ПРОГРАМСКИ ПАРАДИГМИ
модуларно дизајнирана

за IV година
Математичко-информатичка гимназија

Скопје, 2023 година

Назив на наставната програма	Програмски парадигми
Тип на наставна програма	Изборна
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	IV (четврта)
Број на часови неделно/годишно за реализација на наставната програма	3/99
Цели на наставната програма	<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> - да ги продлабочи знаењата по информатика со програмски парадигми различни од процедуралната и објектно-ориентираната, со фокус на декларативната програмска парадигма; - да биде способен да споредува различни програмски парадигми и да избере соодветна за даден проблем; - да ги продлабочи знаењата и да користи логички програмски јазик за имплементација на ефикасни алгоритми; - да биде способен да користи логички програмски јазик за репрезентација и манипулација со знаење;

¹Закон за Националната рамка на квалификации.

	<p>- да биде способен да користи функционален програмски јазик за пишување на програми без странични ефекти;</p> <p>- да постигне разбирање и да користи функции од прв и повисок ред, функционални затворања, мрзлива евалуација и апстракција на секвенци;</p> <p>- да постигне самодоверба во примената на пишување едноставни конкурентни програми;</p> <p>- да ги продлабочи знаењата 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;</p>
Модуларни единици на наставната програма	<ul style="list-style-type: none"> • ПРЕГЛЕД И СПОРЕДБА НА ПРОГРАМСКИ ПАРАДИГМИ • ВОВЕД ВО ДЕКЛАРАТИВНО-ЛОГИЧКА ПРОГРАМСКА ПАРАДИГМА • ВОВЕД ВО ФУНКЦИОНАЛНА ПРОГРАМСКА ПАРАДИГМА • ПАРАЛЕЛНО И КОНКУРЕНТНО ПРОГРАМИРАЊЕ
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 по програмски парадигми може да ја реализира:</p> <p>-наставник со завршени студии по информатика/наставна или друга насока, VII/1 или VIA според МРК и 240 ЕКТС;</p> <p>Стручно лице кое исполнува најмалку еден од следните услови:</p> <p>-да бил ментор на ученик кој бил награден на престижен меѓународен натпревар од соодветната област;</p> <p>- научен степен доктор на информатички науки;</p> <p>-да е запишан на докторски студии на соодветната област;</p> <p>-да има стекнато научен степен на доктор на науки на соодветната област.</p>

• Модуларна единица 1: ПРЕГЛЕД И СПОРЕДБА НА ПРОГРАМСКИ ПАРАДИГМИ (6 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Критериуми за оценување*
1	<p>Ученикот/ученичката ќе биде способен/способна да:</p> <ul style="list-style-type: none"> - Го објаснува поимот програмска парадигма; - Препознава основни програмски парадигми; - Прави класификација на програмските парадигми и нивните основни карактеристики; - Применува програмски парадигми во компјутерите и секојдневниот живот; - Објаснува различни пристапи на решавање на програмски проблеми и примена на различни програмски парадигми; 	<ul style="list-style-type: none"> • Поим за програмски парадигми; • Општа поделба на императивни и декларативни парадигми; • Разлика помеѓу логички и функционални програмски јазици; • Опишување проблеми во декларативни програмски јазици; • Парадигма на конкурентно процесирање и основи на паралелно програмирање. <p>Поими: програмски парадигми, декларативно</p>	<p>Наставникот започнува дискусија со учениците за програмски парадигми и нивна класификација.</p> <ul style="list-style-type: none"> • Учениците илустрираат разлики помеѓу различни програмски парадигми. • Учениците изработуваат практична вежба за опшување проблемски задачи во декларативни програмски јазици. • Наставникот започнува дискусија со учениците за парадигма за конкурентно процесирање и паралелно програмирање. 	<p>Ученикот/ученичката:</p> <ol style="list-style-type: none"> 1.1. Дава примери за програмска парадигма; 1.2. Разликува програмски парадигми и прави избор на соодветна парадигма за ситуации од секојдневниот живот и од другите наставни предмети; 1.3. Споредува и избира алтернативни пристапи при развој на програмски решенија во компјутерите и секојдневниот живот; 1.4. Категоризира пристапи за решавање на различни програмски парадигми; 1.5. Анализира и менува алтернативен пристап при развој на софтверски решенија.

*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"> - Препознава потреба од алтернативен пристап при развој на софтверски решенија. 	програмирање, функционални и логички програмски јазици, конкурентно и паралелно програмирање.	Методи: дискусија, дијалог, демонстрација, учење преку откривање-истражување.	
--	---	---	--	--

Модуларна единица 2: ВОВЕД ВО ДЕКЛАРАТИВНО ЛОГИЧКА ПРОГРАМСКА ПАРАДИГМА (40 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/способна да:</p> <ul style="list-style-type: none"> - Користи искази и исказни формули, како и да ја определува нивната вистинитост; - Употребува конјунктивна нормална форма (KNF) и дисјунктивна нормална форма (DNF); - Користи методи за проверка дали дадено тврдење е логичка последица од други тврдења или не; - Го применува DPLL (Davis-Putnam-Logemann-Loveland) алгоритмот, преку проверка на двете можни вистинитосни 	<ul style="list-style-type: none"> • Вовед во декларативно логичка програмска парадигма • Повторување за исказна логика, изучуван во математичките предмети: • Исказно сметање • Предикатна логика од прв ред • Основи на логички програмски јазик (Prolog) • Претставување на предикатни формули во облик на логичка програма • Пребарување со враќање наназад 	<p>Наставникот презентира вовед во декларативно логичка програмска парадигма</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за исказна логика. • Дискусија за предикатна логика од прв ред. • Наставникот ги запознава учениците со основните карактеристики на логичкиот програмски јазик Prolog, неговата синтакса и семантика. • Учениците составуваат споредбени изрази и логички изрази врз база на дадени текстуални барања. 	<p>Ученикот/ученичката:</p> <ol style="list-style-type: none"> 1.1. Користи искази и исказни формули; 1.2. Определува вистинитост на исказни формули; 1.3. Применува конјунктивна нормална форма и дисјунктивна нормална форма; 1.4. Применува DPLL алгоритам и метод на резолуција; 1.5. Претставува произволни предикатни формули во облик на логички програми низ фазите на сколемизација, супституција и унификација; 1.6. Креира програми за решавање проблемски задачи во програмскиот јазик Prolog;

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

	<p>вредности кои се појавуваат во дадена формула;</p> <ul style="list-style-type: none"> - Применува метод на резолуција; - Дефинира предикатни формули и нивна интерпретација; - Претставува произволна предикатна формула во облик на логичка програма низ неколку фази (пренекс нормална форма, сколемизација, супституција, унификација); - Го користи програмскиот јазик Prolog - Генерализира листи во податочни објекти со повеќе компоненти; - Го објасни концептот на странични ефекти и нивно користење во програмирањето; - Именува класи на проблеми погодни за решавање преку декларативно програмирање; - Дефинира правила за решавање на комбинаторни и логички проблемски решенија; 	<ul style="list-style-type: none"> • Клаузална репрезентација на податочни структури и алгоритми • Рекурзија • Основни правила за работи со листи • Прекини • Негација • Претставување и манипулација со знаење со користење на логички програмски јазик • Класи на проблеми погодни за решавање со користење на декларативно логичка програмска парадигма. • Решавање на различни комбинаторни и логички проблеми преку програмирање • Експертни системи • Креирање едноставен експертен систем за препознавање <p>Поими: Декларативна програмска парадигма, исказно сметање, предикатна логика, логички</p>	<ul style="list-style-type: none"> • Наставникот објаснува за DPLL алгоритмот и методот на резолуција. • Презентација на пример програми. Презентација на примери од секојдневието. • Учениците изработуваат програми со методи за проверка на тврдење, дали е логичка последица од други тврдења или не е последица од истите. • Наставникот дискутира за предикатни формули и нивна интерпретација. • Приказ на претставување на произволна предикатна формула во облик на логичка програма во Prolog, низ соодветни фази. Примена на методот на резолуција. • Листи како сложени структури на податоци и операции со нив. • Наставникот објаснува за примената на рекурзија кај програмскиот јазик Prolog. 	<p>1.7. Користи основни правила на припадност на листи, спојување на листи, бришење елементи од листа;</p> <p>1.8. Креира програми за решавање проблемски задачи со пермутации, варијации и комбинации;</p> <p>1.9. Креира програми за решавање задачи со логички проблемски барања;</p> <p>1.10. Имплементира познати алгоритми во Prolog;</p> <p>1.11. Избира компоненти на една Prolog програма – програмски клаузули;</p> <p>1.12. Комбинира елементарни аритметички операции во процес на доаѓање до одговор во Prolog;</p> <p>1.13. Развива сложени структури на податоци – листи, спојување на листи, бришење...);</p> <p>1.14. Планира и модифицира експертен систем за распознавање на различни облици.</p>
--	---	--	--	--

	<p>- креира едноставен експертен систем за распознавање на различни облици, предмети и сл.</p>	<p>програмски јазик, пренекс нормална форма, сколемизација, супституција и унификација, резолуција, пребарување, рекурзија, листи, прекини и негација, експертен систем, Prolog</p>	<ul style="list-style-type: none"> • Учениците изработуваат низа од програми со процес на доаѓање до одговор (со вклучување на сите од досега изучените методи). • Наставникот дискутира за класи на проблеми погодни за решавање со користење на декларативна програмска парадигма. Илустрација на дефинирање на правила за решавање на комбинаторни (пермутации, варијации, комбинации) и логички проблеми (Мисионери и канибали, Ајнштајновиот проблем на куќи). • Учениците пишуваат програми за решавање на различни комбинаторни и логички проблеми. • Дискусија за експертни системи. Примери за експертни системи и нивна примена од секојдневниот живот. • Учениците креираат едноставен експертен систем за препознавање. 	
--	--	---	--	--

			Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку компјутерски програми	
--	--	--	--	--

Модуларна единица 3: ВОВЕД ВО ФУНКЦИОНАЛНА ПРОГРАМСКА ПАРАДИГМА (28 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/способна да:</p> <ul style="list-style-type: none"> - Користи функционална парадигма; - Користи програмски јазик Lisp; - Применува рекурзија преку функционално програмирање; - Користи рекурзивни функции и функции од повисок ред (map, filter, fold) при реализација на програми; - Употребува алгебарски типови на податоци и параметарски полиморфизам; 	<ul style="list-style-type: none"> • Вовед во функционална програмска парадигма • Програмирање без странични ефекти • Структурирани податочни типови и нивно процесирање • Функции како објекти од прв ред • Анонимни функции • Функции од повисок ред • Функционални затворања • Парцијални функции • Мрзлива евалуација 	<p>Наставникот започнува дискусија со учениците за функционална програмска парадигма.</p> <ul style="list-style-type: none"> • Наставникот објаснува за предности и недостатоци на функционалната парадигма преку употреба на примери • Наставникот објаснува за функционалниот јазик Lisp • Учениците изработуваат програми во Lisp и решаваат проблемски задачи 	<p>Ученикот/ученичката:</p> <ol style="list-style-type: none"> 1.1. објаснува за концептот на функционална програмска парадигма; 1.2. креира програми со примена на функции од прв и повисок ред; 1.3. објаснува за предности и недостатоци на функционалната парадигма; 1.4. применува парадигми засновани на правила; 1.5. креира програми кои користат синтаксни изрази и функции;

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

	<p>- Креира програми со влез/излез со употреба на стандардни библиотеки.</p>	<ul style="list-style-type: none"> • Бесконечни секвенци и нивна апстракција • Конкурентност и паралелизам • Карактеристики на функционален програмски јазик (Lisp) <p>Поими: функционална и програмска парадигма, структурирани податочни типови, функции од прв и повисок ред, рекурзија и рекурзивни функции, конкурентност и паралелизам, програмски јазик Lisp</p>	<ul style="list-style-type: none"> • Учениците работат со синтаксни изрази и функции • Решаваат задачи со употреба на рекурзивни функции • Наставникот дискутира заедно со учениците за конкурентност и паралелизам. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање проблеми преку компјутерски програми.</p>	<p>1.6. креира програми кои користат рекурзивни функции; 1.7. комбинира типови податоци и параметарски полиморфизам; 1.8. Правилно интегрира стандардни библиотеки во програми.</p>
--	--	---	---	---

Модуларна единица 4: ПАРАЛЕЛНО И КОНКУРЕНТНО ПРОГРАМИРАЊЕ (25 часа)				
Ред. број	Резултати од учењето	Содржини и поими	Активности и методи	Критериуми за оценување*
1	<p>Ученикот/ученичката ќе биде способен/способна да:</p> <ul style="list-style-type: none"> - Објаснува за концептите на конкурентно и паралелно програмирање; 	<ul style="list-style-type: none"> • Концепти на конкурентно процесирање и паралелно програмирање 	<p>Наставникот започнува дискусија за:</p> <ul style="list-style-type: none"> - концептите за конкурентно и паралелно програмирање; 	<p>Ученикот/ученичката:</p> <p>1.1. ги објаснува поимите конкурентно процесирање и паралелно програмирање;</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"> - Разликува различни нивоа на апстракција во конкурентно и паралелно програмирање; - Имплементира програми со конкурентно и/или паралелно програмирање за стандардни проблеми; - Самостојно да решава проблеми со синхронизација; - Изработува елементарни програми во избран програмски јазик. 	<ul style="list-style-type: none"> • Различни начини на имплементација на препреки • Семафори • Дистрибуирани бинарни семафори • Алокација на ресурси • Препраќање пораки • Асинхроно и синхроно испраќање на пораки • Синхрона, асинхрона и условна комуникација • Комуникација со сигурни врски <p>Поими: конкурентно процесирање и паралелно програмирање, семафори, алокација на ресурси, испраќање пораки, повици, препреки, синхронизација</p>	<ul style="list-style-type: none"> - потребата за примена на овие концепти; - нивоа на апстракција. • Наставникот презентира пример програми со конкурентно и/или паралелно програмирање за најчести проблеми на различни јазици за програмирање. • Наставникот започнува дискусија за: <ul style="list-style-type: none"> - семафори. - бинарни семафори. • Наставникот ги презентира концептите за останатите наведени теми со наведување пример програмски кодови • Учениците изработуваат програми со реализација на семафори, условни региони, монитори и пренасочување на пораки. <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер -</p>	<p>1.2. препознава различни нивоа на апстракција во конкурентно и паралелно програмирање;</p> <p>1.3. дефинира и применува концепти, алгоритми, принципи, проблеми и решенија поврзани со конкурентно и паралелно програмирање;</p> <p>1.4. може да решава проблеми со примена на семафори, монитори и пренасочување на пораки;</p> <p>1.5. користи програмски јазици кои поддржуваат паралелно и/или конкурентно програмирање при решавањето на задачи;</p> <p>1.6. креира елементарни програми во избран програмски јазик.</p>
--	--	---	---	--

			решавање проблеми преку компјутерски програми	
--	--	--	---	--

<p>Оценување на постигањата на учениците</p>	<p>Во текот на наставата по програмски парадигми 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 програмски парадигми 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 програмски парадигми 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.
<p>Почеток на имплементација на наставната програма</p>	<p>Учебна 2023/2024 година</p>

Институција/ носител на програмата	Биро за развој на образованието (БРО)
Потпис и датум на донесување на наставната програма	бр. 13-6691/17 4.7.2023 година <p style="text-align: right;">МИНИСТЕР, Doc. Dr. Jeton Shaqiri</p> <hr/>
Датум на ревизија	