

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, 96/2019 и 110/2019) и член 22 став 1 од Законот за средното образование („Службен весник на Република 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 наставниот предмет **Објектно-ориентирано програмирање** за III (трета) година математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма
ОБЈЕКТНО – ОРИЕНТИРАНО ПРОГРАМИРАЊЕ
модуларно дизајнирана

за III година
Математичко-информатичка гимназија

Скопје, 2022 година

| | |
|--|---|
| Назив на наставната програма | Објектно-ориентирано програмирање |
| Тип на наставна програма | Задолжителна |
| Кредитна вредност на наставната програма | 4 (четири) ЕЦВЕТ ¹ кредити |
| Ниво на квалификација | IV(четврто) ниво |
| Година на изучување | III (трета) |
| Број на часови неделно/годишно за реализација на наставната програма | 2/72 |
| Цели на наставна програма | <p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> - да ги продлабочи знаењата по програмирање во областа на објектно-ориентираното програмирање 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"> • ОБЈЕКТИ И КЛАСИ • ОБРЕМЕНУВАЊЕ ОПЕРАТОРИ • КОНЦЕПТ - НАСЛЕДУВАЊЕ • КОНЦЕПТ - ПОЛИМОРФИЗАМ • НАСТАН - БАЗИРАНО ПРОГРАМИРАЊЕ |
| <p>Материјално-технички и просторни услови</p> | <p>За постигнување на целите на наставата по објектно-ориентираното 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>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; - 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: ОБЈЕКТИ И КЛАСИ (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 дефинира што е објектно ориентираното програмирање; - да ги распознава, користи и имплементира елементите на ООП (класи, објекти, функции-членки, податочни членки) и нивната функција; - да ги дефинира и применува: <ul style="list-style-type: none"> • операторите new и delete; • покажувачот this; • деструктор; • статички членови на класа; • константни објекти и константни функции членки; • композиција на објекти; • пријателски функции на класа; | <ul style="list-style-type: none"> • Класи и објекти • Дефинирање класи и декларирање на објекти • Директен пристап до членови во класа • Конструктори • Дефинирање на функции/членки надвор од класа • Пристап до членовите во класата • Функции за поставување и преземање на податоци од класа • Оператори new и delete • Објект како аргумент на функција • Доделување еден објект на друг објект • Иницијализација на објекти | <p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за поимите: класа, објект, податок – атрибут, функција и пристап до објект • Учениците разгледуваат примери на класи со објекти и пристап до нив • Презентација на функции за поставување и преземање на податоци од класа; • Учениците изработуваат програми со класи и функциите за поставување и преземање на податоци од класа; • Презентација за конструктор и деструктор; | <p>Ученикот/ученичката може да:</p> <p>1.1: објаснува за концептите и елементите на објектно - ориентираното програмирање;</p> <p>1.2: објаснува за конструктор и деструктор и знае да го примени во програми со класи;</p> <p>1.3: дефинира функција и ја применува во програма за поставување и преземање на податоци од класа;</p> <p>1.4: објаснува за операторите new и delete;</p> <p>1.5: објаснува за покажувачот this и да го примени во програми со класи;</p> <p>1.6: објаснува за концептот – композиција на објекти и изработи програма со овој концепт;</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"> - да знае за можноста за издвојување на интерфејсот од имплементацијата и да умее да постави класа во посебна датотека; - да ги применува изучените концепти на ООП при конструкција на класи и изработка на програми; | <ul style="list-style-type: none"> • Враќање објект во функција • Показувач this • Деструктори • Поставување на класа во посебна датотека • Издвојување на интерфејс од имплементација • Концепт – композиција на објекти • Пријателски функции на класа • Статички членови на класа • Константни објекти • Константни функции членки • Принцип - Криење на податоци • Студија на случај за дадена класа: <ul style="list-style-type: none"> - делокруг на класа и пристап до членките од класата - функции за пристап и услужни функции • Изработка на програми со класи и објекти <p>Поими : класа, објект, атрибут, идентитет, објектно-ориентирано програмирање,</p> | <ul style="list-style-type: none"> • Учениците разгледуваат готови пример програми и изработуваат програми со конструктор и деструктор; • Презентација за операторите new и delete • Презентација за показувач this • Учениците разгледуваат готови пример програми и изработуваат програми со показувач this • Презентација за поставување на класа во посебна датотека со пример на програми; • Презентација за концептот – композиција на објекти • Учениците разгледуваат готови пример програми и изработуваат програми со композиција на објекти; • Презентација за техниката издвојување на интерфејс од имплементација; • Презентација за пријателски функции, статички членови на класа и за принципот – Криење на податоци; | <p>1.7: прави разлики меѓу начините на пристап до членовите во класа; 1.8: објаснува за поставување на класа во посебна датотека; 1.9: објаснува за издвојување на интерфејс од имплементација 1.10: изработува програми со класи со примена на изучените концепти на ООП</p> |
|--|---|--|--|--|

| | | | | |
|---|---------------------------|--|--|---------------------------------|
| | | <p>функција-членка, податочна членка, конструктор, деструктор, оператори new и delete, покажувач this, интерфејс, имплементација, композиција на објекти, статички членови, константни објекти, константни функции, пријателски функции, принцип - криење на податоци</p> | <ul style="list-style-type: none"> Учениците изработуваат програми со класи со примена на изучените концепти на ООП; <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми со примена на класи и објекти</p> | |
| Модуларна единица 2: ОБРЕМЕНУВАЊЕ ОПЕРАТОРИ (8 часа) | | | | |
| Ред. број | Резултати од учење | Содржини и поими | Активности и методи | Критериуми на оценување* |

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

| | | | | |
|---|---|---|---|--|
| 1 | <p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - да објасни што значи обременување на функции; - да објасни и дефинира операторски функции; - да објасни и дефинира операторска функција за обременување бинарни оператори; - да објасни и дефинира операторска функција за обременување унарни оператори; - да креира и применува операторски функции во програми со класи, со обременување бинарни и унарни оператори; | <ul style="list-style-type: none"> • Поим за обременување функции • Операторски функции и нивно дефинирање • Операторски функции за обременување бинарни оператори • Примена на операторски функции за обременување бинарни оператори во програми со класи • Операторски функции за обременување унарни оператори • Примена на операторски функции за обременување унарни оператори во програми со класи <p>Поими : преклопени функции, операторски функции, operator, бинарни оператори, унарни оператори</p> | <p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за обременување функции; • Наставникот започнува дискусија со учениците за операторски функции, бинарни оператори и унарни оператори • Презентација за начинот на дефинирање на операторски функции за обременување бинарни и унарни оператори; • Учениците изработуваат програми со класи и со примена на операторски функции за обременување бинарни и унарни оператори; <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p> | <p>Ученикот/ученичката може да:</p> <p>1.1: објаснува за операторски функции за обременување бинарни оператори;</p> <p>1.2: објаснува за операторски функции за обременување бинарни оператори;</p> <p>1.3: применува операторски функции за обременување бинарни оператори во програми со класи;</p> <p>1.4: применува операторски функции за обременување унарни оператори во програми со класи;</p> <p>1.5: пишува програми со изучените концепти за обременување.</p> |
|---|---|---|---|--|

| Модуларна единица 3: КОНЦЕПТ – НАСЛЕДУВАЊЕ (10 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 креира надкласа (основна класа) и изведени класи (подкласи) и соодветно да работи со хиерархија од класи; - да објасни за разделување на интерфејсот и имплементацијата при наследување и за разделување на сервисите на наткласите и поткласите при наследување; - да објасни и дефинира како се врши конверзија на типот на објекти (кастирање) - имплицитна и експлицитна конверзија; - да објасни како се одредува типот на објектот во класата: typeid() - да го преименува изучениот концепт на наследување во | <ul style="list-style-type: none"> • Основни класи и изведени класи (наткласи и поткласи) • Редифинирање на функција членка на наткласа • Конструктори при наследување • Разделување на интерфејсот и имплементацијата при наследување • Разделување на сервисите на класите при наследување • Наследување на заштитени членови од наткласа • Повеќенивовско наследување • Наследување од повеќе класи • Конверзија и одредување на тип на објекти • Изработка на програми со класи со примена на концепт на наследување | <p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за основна класа (наткласа) и изведена класа (поткласа); • Наставникот објаснува за начинот на редифинирање на функција членка на основната класа; • Презентација за начинот на повикување на конструктори при концепт на наследување со примери на програми; • Презентација за разделување на сервисите на наткласите и поткласите при наследување со примери на програми; • Презентација за повеќенивовско | <p>Ученикот/ученичката може да:</p> <p>1.1: објаснува за основна класа (надкласа) и изведена класа (подкласа);</p> <p>1.2: креира хиерархија од класи и ја користи во решавање конкретни проблеми;</p> <p>1.3: редифинира функција членка на надкласа и да повикува инструктори при концепт на наследување;</p> <p>1.4: објаснува за повеќенивовско наследување и наследување од повеќе класи;</p> <p>1.5: објаснува и да врши конверзија на типот на објектите (кастирање);</p> <p>1.6: го одредува типот на објектот во класата;</p> <p>1.7: изработува програми со примена на концепт на</p> |

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

| | | | | |
|--|---|--|--|--|
| | <p>ООП при конструкција на класи и изработка на програма за решавање на конкретен проблем</p> | <p>Поими: наткласа-основна класа, поткласа-изведена класа, наследување, релација, редефинирана класа, повеќенивовско (проширено) наследување, кастирање (конверзија) на типот на објекти, имплицитна конверзија, експлицитна конверзија</p> | <p>наследување и наследување од повеќе класи со примери на програми;</p> <ul style="list-style-type: none"> • Презентација за конверзија и одредување на типот на објектот со примери на програми; • Учениците практично изработуваат програми со класи со примена на концепт - наследување <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер, решавање проблеми преку програми</p> | <p>наследување кај класи за решавање на конкретен проблем.</p> |
|--|---|--|--|--|

| Модуларна единица 4: КОНЦЕПТ – ПОЛИМОРФИЗАМ (8 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"> • Поим за полиморфизам • Виртуелни функции и нивно извршување | Активности: | Ученикот/ученичката може да: |

*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"> - да дефинира полиморфизам како концепт на ООП; - да дефинира и објасни како се извршуваат виртуелните функции; - да дефинира апстрактни класи и конкретни класи; - да го преименува изучениот концепт на полиморфизам во ООП при конструкција на класи и изработка на програма за решавање на конкретен проблем | <ul style="list-style-type: none"> • Апстрактни класи • Конкретни класи • Полиморфизам во повеќенивовска хиерархија • Полиморфизам и композиција на објекти • Изработка на програми со класи со примена на концепт на полиморфизам <p>Поими: полиморфизам, виртуелна функција, апстрактни класи, конкретни класи, повеќенивовска хиерархија</p> | <ul style="list-style-type: none"> • Наставникот започнува дискусија за поимот полиморфизам; • Презентација за тоа што се виртуелни функции и како се извршуваат преку пример програми; • Учениците изработуваат програми со класи и виртуелни функции; • Презентација за тоа што се апстрактни класи а што конкретни класи преку пример програми; • Презентација на програми каде се применува полиморфизам во повеќенивовска хиерархија; • Презентација на програми со класи каде се применува концептот полиморфизам и концептот композиција; • Учениците практично изработуваат програми со класи со примена на концепт – полиморфизам; | <p>1.1: го објаснува поимот полиморфизам како концепт во ООП;</p> <p>1.2: опишува и демонстрира виртуелни функции во класа;</p> <p>1.3: дава пример за апстрактни класи, наспроти конкретни класи;</p> <p>1.4: имплементира класни хиерархии со полиморфизам</p> <p>1.5: користи полиморфизам во повеќенивовска хиерахија;</p> <p>1.6: изработува програми со примена на концепт на полиморфизам кај класи за решавање на конкретен проблем.</p> |
|--|---|--|--|--|

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

| Модуларна единица 5: НАСТАН - БАЗИРАНО ПРОГРАМИРАЊЕ (16 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 дефинира настан-базирано (визуелно) програмирање; - практично да применува алатки за визуелен развој на апликации, односно визуелно програмирање на апликации со графички кориснички интерфејс; - да ги применува изучените програмски техники при | <ul style="list-style-type: none"> • Компоненти на интегрирана развојна околина (IDE) за настан-базирано (визуелно) програмирање (едитор, компајлер, поврзувач, библиотеки) • Запознавање со основните елементи на интегрирана развојна околина (IDE) за настан-базирано програмирање • Основни визуелни контроли (форми, копчиња, менија) и настани | <p>Активности:</p> <ul style="list-style-type: none"> • Наставникот започнува дискусија со учениците за поимот настан-базирано (визуелно) програмирање и негова практична примена; • Презентација на претходно креирани програмски решенија со визуелни компоненти; • Презентација на основните визуелни контроли (форми, копчиња, менија) и настани | <p>Ученикот/ученичката може да:</p> <p>1.1: го објаснува настан-базираното (визуелно) програмирање;</p> <p>1.2: ги објаснува елементите на развојна околина за настан-базирано програмирање</p> <p>1.3: креира форми и контролите во нив, со поставување и менување на нивните својства во интегрирана развојна околина (IDE) за настан-базирано (визуелно) програмирање;</p> |

| | | | | |
|--|--|---|--|--|
| | <p>креирање на визуелни програмски решенија;</p> | <ul style="list-style-type: none"> • Лента со алатки Toolbox и нејзини можности • Креирање на проекти, поставување и уредување на визуелни компоненти • Креирање на визуелни програми (GUI апликации) кои извршуваат одредена активност (аритметички операции - калкулатор, читање на вредности од тастатура и прикази на резултати во етикета, конвертор на валути и сл.) <p>Поими: настан-базирано (визуелно) програмирање, форми, контроли, својства, лента со алатки – Toolbox, копче (анг. Button), копче со избор етикета (анг. label), поле за текст, поле со листа, ракувач со настани, .NET рамка, GUI апликација, CLR Forms Application</p> | <p>во интегрирана развојна околина (IDE) за визуелно програмирање;</p> <ul style="list-style-type: none"> • Презентација на лентата со алатки (Toolbox) преку конкретен пример; • Учениците изработуваат едноставен проект, поставуваат основни визуелни контроли и ги уредуваат; • Учениците изработуваат едноставна програма во визуелно програмирање каде се користи поле за текст и етикета и испишување на соодветна порака во етикета; • Учениците изработуваат проекти со вклучување на аритметички операции – калкулатор, конвертор на валути, работа со текстуални низи – наоѓање на збор и замена, плоштина и волумен на геометриски фигури и сл.) | <p>1.3: креира проекти со визуелно програмирање кои решаваат проблеми од секојдневниот живот</p> |
|--|--|---|--|--|

| | | | | |
|--|--|--|---|--|
| | | | Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер со креирање на визуелни програми кои решаваат ситуации од секојдневниот живот | |
|--|--|--|---|--|

| | |
|---|---|
| <p>Начини на изведување на наставата</p> | <p>Часовите се изведуваат во компјутерскиот кабинет. На почетокот треба да се изврши проверка на нивото на знаење и вештини на ученикот, која треба да послужи како водич во организација на часовите и евентуално индивидуализација на наставата.</p> <p>Активностите треба да вклучуваат практична работа, примена на ИКТ, поврзување и примена на содржини од различни теми, теми и области кои учениците ги среќаваат надвор од училиштето. Активностите се дизајнирани да ја зголемат мотивацијата за учењето и да го поттикнат формирањето на ставови, верувања и вредносни системи во однос на развојот на јазикот и компјутерската писменост, здравиот начин на живот, развој на креативноста, способноста за оценување и самоевалуација.</p> <p>При спроведувањето на програмата се дава приоритет на проектната, проблемската и активната настава, кооперативно учењето, градење на знаење и развој на критичко размислување. Поттикнување на тимска работа и соработка, особено во областите каде што наставникот оценува дека постојат големи разлики во знаењето на одредени ученици. Ако условите дозволуваат, на учениците треба да се даде поддршка на хибридниот модел на настава (комбинација на традиционално учењето и електронско поддржано учењето), особено кога постои потреба за индивидуализација на наставата поради поголеми разлики во предзнаењето.</p> |
| <p>Оценување на постигањата на учениците</p> | <p>Во текот на наставата по Објектно-ориентирано програмирање се препорачува формативно следење кое вклучува изработка и водење на портфолио на учениците што опфаќа:</p> <ul style="list-style-type: none"> - собирање на 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 Објектно-ориентирано програмирање оценувањето може да се реализира усно, практично, со презентација и слично.</p> |

