

MicroBit и сл.) могуће их је искористити у овим пројектима, а у недостатку физичких уређаја може се искористити нека од платформи за симулацију (нпр. Tinkercad или нека слична). Најпре се ученици упознају са основама рада на платформама за симулацију или са самим физичким уређајима: рад са диодама, са rgb диодом, потенциометром (аналогни, дигитални), фоторезистор, УЗ сензор, сервомотор, тастатура, ДЦ мотор, дисплеј, тастери, звучник, ПИР сензор, топлотни сензор. Након тога раде пројекте из одабраних тема, на пример: Програмирање семафора, Прављење зеленог таласа у граду, Аутоматско отварање врата у болницама, алармни систем у стану, алармни систем у банци који заробљава лопова, светлосни ефекти у дискотеци, паркинг сензори, аутоматска хранница за псе и мачке, сеизмограф, стакленик, симулација производе траке у некој фабрици са сортирањем различитих производа и кретањима до магацина и сл.

На наставнику је да изабере обим којим ће се обрадити предузетништво и дигитални стартап. Обавезно је да ученици креирају презентацију у којој представљају свој тим и урађени практични пројекат, и да тимски презентују пред одељењем. Напредније би било да поред тога ученици детаљно раде предложена истраживања тржишта, процењују фактора ризика, анализирају потенцијалне кориснике и тимски креирају детаљан бизнис план или бизнис модел канвас.

Приликом реализације ове теме обавезно се ослонити на материјале за наставу релевантних организација и пројекта из области предузетништва за средњошколце од који су најзначајнији наведени у наставку:

– Организација Достигнућа младих у Србији, део глобалне мреже Junior Achievement Worldwide и њихови програми Ученичка компанија, Пословни изазов и друго;

– Пројекат „Предузми идеју”, који спроводи Иницијатива Дигитална Србија;

– Пројекат „Startit” непрофитне организације SEE ICT која кроз едукацију, информисање, мотивацију и повезивање подстиче технолошки развој.

Према могућностима, укључити и родитеље, локалне стручњаке и предузетнике, наставнике других предмета...

Стартап је врста предузећа, али има и своје специфичности. Дигитални стартап се углавном везује за креирање новог ИТ производа, на пример апликације, који може да доживи изузетно велики раст без једнако брзог раста трошкова. Истражити трендове у глобалној економији и објаснити савремене економске системе. Кроз дискусију са ученицима дефинисати појам ИТ иновације и узрочно-последичне везе између ИТ иновација и предузетништва. Направити осврт на неке успешне светске и домаће стартапе.

Важан документ који ученици треба да израде је бизнис план. Потребно је да бизнис план покрије истраживање тржишта, SWOT анализу (снаге, слабости, прилике и претње), финансијску пројекцију (цена производа, трошкови, буџет, профит за одређени период), планирану производњу, дистрибуцију и елементе маркетинга. Проценити потенцијалне кориснике и величину тржишта. Размислити о кључним партнерима, умрежавању и успостављању сарадње. Истражити конкуренте. За стартапе може да се користи и бизнис модел канвас чији је централни део пропозиција вредности који обухвата цену, исход коришћења производа, задовољство корисника након употребе и слично.

Поставити као захтев ученицима да у своје бизнис планове уграде и елементе друштвено одговорног и еколошког одговорног пословања. Организовати истаживачки рад ученика и дискусију на неку од тема: родна равноправност, запошљавање особа из дискриминисаних група, проблеми особа са инвалидитетом, итд. Навести ученике да се заинтересује на које начине компаније могу да се понашају еколошко одговорно (потрошња енергије, отпад, рециклажа...).

Ученици презентују коначне урађене пројекте и кроз дискусију и критички однос их анализирају и вреднују по задатим

критеријумима. Као обавезан део презентације урадити пич (енг. pitch), кратку презентацију од 5 до 10 минута у којој се јасно износи идеја стартапа на начин да заинтересује потенцијалне инвеститоре. Пич треба да има елементе представљања производа, тржишног потенцијала и способности тима да креирају успешан стартап.

## Рачунарство и друштво (6)

Ова тематска целина би требало да представља завршницу предмета „Примена рачунара” и да буде својеврсна систематизација трогодишњег рада. У светлу различитих области примене обрађених у претходној теми урадити анализу међузависности развоја људског друштва и ИКТ-а. Анализирати етичка питања која се појављују као последица велике употребе технологије у савременом друштву.

Посебну пажњу посветити области индустрије софтвера (производња, квалитет, тестирање, лиценцирање и одржавање).

## III. ПРАЋЕЊЕ И ВРЕДНОВАЊЕ НАСТАВЕ И УЧЕЊА

С обзиром да се за овај наставни предмет препоручује реализација хибридног модела наставе – комбинација традиционалне наставе и електронски подржаног учења коришћењем неког од система за управљање учењем, треба одабрати систем за управљање учењем који има развијене алате за праћење и вредновање рада ученика. На тај начин је олакшано редовно и транспарентно праћење рада, давање повратних информација и вредновање рада коришћењем домаћих задатака, тестова и дискусионих форума.

При обради тематске целине *Примена рачунара у математици* редовно задавати домаће задатке и блиц-тестове како би ученици били мотивисани да редовно прате градиво. На крају већих целина реализовати контролне вежбе.

При дискусијама у првом делу тематске целине *Примена рачунара у разним областима* применити вршњачку евалуацију рада и редовно давати ученицима повратну информацију. У другом делу ученици тимски развијају план рада и начин праћења успешности реализације плана и вреднују своју улогу при изради пројектног задатка и активности за које су били задужени.

Тематску целину Рачунарство и друштво реализовати по принципу систематизације градива, кроз разговоре, дискусије и дебате и у складу са тим методама реализовати праћење и вредновање рада.

## ПРОГРАМИРАЊЕ

Циљ учења Програмирања је развој алгоритамског приступа решавању проблема код ученика, овладавање техникама програмирања и стицања знања о савременим програмским језицима.

## ОПШТА ПРЕДМЕТНА КОМПЕТЕНЦИЈА

Учењем наставног предмета Програмирање ученик је развио способност решавања проблема развојем логичког и критичког мишљења и позитивне ставове према рачунарским наукама. Ученик је упознат са основним и неким напреднијим концептима програмирања.

## СПЕЦИФИЧНЕ ПРЕДМЕТНЕ КОМПЕТЕНЦИЈЕ

Специфичне предметне компетенције представљају опис специфичних способности ученика које му омогућавају да развије општу предметну компетенцију. Подразумевају способност за прецизно и концизно дефинисање проблема; разумевање потребе за алгоритамским начином решавања проблема, као и писање модуларних и добро структурираних програма.

Разред	Трећи
Недељни фонд часова	3 часа
Годишњи фонд часова	11 часова

ИСХОДИ	ТЕМА
По завршетку разреда ученик ће бити у стању да:	и кључни појмови садржаја програма
<ul style="list-style-type: none"> <li>– имплементира проверу да ли је број прост у времену <math>O(\sqrt{n})</math> и образложи коректност тог алгоритма;</li> <li>– Ератостеновим ситом проналази све просте бројеве мање од броја <math>n</math> у времену <math>O(n \log \log n)</math>;</li> <li>– имплементира факторизацију броја у времену <math>O(\sqrt{n})</math> и образлаже коректност тог алгоритма;</li> <li>– одреди НЗД и НЗС датих бројева <math>m</math> и <math>n</math> у времену <math>O(\log(m+n))</math> применом Еуклидовога алгоритма и образложи коректност тог алгоритма;</li> <li>– примени Ератостеново сито, факторизацију броја и одређивање НЗД и НЗС у ефикасном решавању алгоритамских задатака</li> <li>– примени операције над координатама тачака и вектора на решавање елементарних проблема рачунарске геометрије;</li> <li>– илуструје веома једноставне геометријске алгоритме програмима са ГКИ;</li> <li>– имплементира основне операције над конвексним многоугловима (провера припадности тачке у времену <math>O(\log n)</math>, израчунавање обима, површине, ...);</li> <li>– применом библиотечких или самостално дефинисаних функција имплементира основне операције над нискама (претрага подниске, издававање подниске, ...);</li> <li>– примењује регуларне изразе за проналажење шаблона у текстуалним датотекама;</li> <li>– контекстно-слободним граматикама описује, а техником рекурзивног спуста анализира изразе;</li> <li>– представља граф помоћу матрице повезаности или листа суседа;</li> <li>– имплементира рекурзивни обилазак графа у дубину;</li> <li>– имплементира нерекурзивни обилазак графа у дубину (помоћу стека) и у ширину (помоћу реда);</li> <li>– примењује обилазак графа у решавању алгоритамских задатака;</li> <li>– на примеру прикаже рад одабраног напреднијег алгоритма или структуре података;</li> <li>– уз коришћење литературе имплементира напредније алгоритме и структуре података.</li> </ul>	<p><b>Основни алгебарски алгоритми</b> Прости бројеви. Растављање броја на просте чиниоце и примене. Највећи заједнички делилац и Најмањи заједнички садржалац и примене.</p> <p><b>Геометријски алгоритми</b> Елементарни алгоритми над правима и дужима (припадност тачке правој и дужи, пресеци правих, пресеци дужи, ...). Елементарни алгоритми над троугловима и четвороугловима у равни (израчунавање обима, површине, припадности тачке унутрашњости, одређивање пресека, ...). елементарни алгоритми над конвексним многоугловима (провера припадности тачке, израчунавање обима и површине, ...). Основни елементи рачунарске графике и визуелизације.</p> <p><b>Алгоритми над текстом и текстуалним датотекама</b> Ниске и основне операције над нискама (претрага подниске, издававање подниске, ...). Регуларни изрази. Техника рекурзивног спуста.</p> <p><b>Графовски алгоритми</b> Представљање графа у програму. Алгоритми за обилазак графа. Одређивање компонената повезаности. Тополошко сортирање.</p> <p><b>Одабрани алгоритми и структуре података</b> По избору наставника бира се једна или више тема. Одабрани графовски алгоритми (Примов и Краскелов алгоритам, Дајкстрин алгоритам, Флојд-Варшалов алгоритам, ...). Одабрани алгебарски алгоритми (проширени Еуклидов алгоритам и примене, Кинеска теорема о остацима, ...). Одабрани геометријски алгоритми (алгоритми за одређивање конвексног омотача, ефикасно одређивање свих пресека скупа дужи, пресеци многоуглова, ...). Одабрани алгоритми над нискама (алгоритам КМП, Бојер-Муров алгоритам, хеширање ниски, Рабин-Карпов алгоритам, Маначеров алгоритам, Z-алгоритам, суфиксна дрвета, ...). Имплементација и примена одабраних структура података (повезане листе, бинарна дрвета, хип, сегментна и Фенвикова дрвета, ...).</p>

## УПУТСТВО ЗА ДИДАКТИЧКО-МЕТОДИЧКО ОСТВАРИВАЊЕ ПРОГРАМА

Настава вежби се изводи са половином одељења у рачунарском кабинету, у групама не већим од 12 ученика.

### I. ПЛАНИРАЊЕ НАСТАВЕ И УЧЕЊА

Приликом планирања часа, исходе предвиђене програмом треба разложити на мање и на основу њих планирати активности за конкретан час. Треба имати у виду да се исходи у програму разликују, да се неки могу лакше и брже остварити, док је за одређене исходе потребно више времена, активности и рада на различитим садржајима. Исходе треба посматрати као циљеве којима се тежи током једне школске године.

При обради нових садржаја треба се ослањати на постојеће искуство и знање ученика, и настојати, где год је то могуће, да ученици самостално откривају правилности и изводе закључке. Ученике треба упућивати да користе литературу (уџбенике и друге изворе знања), да би усвојена знања била трајнија и шира, а ученици оспособљени за самостално решавање разноврсних задатака.

На часовима треба комбиновати различите методе и облике рада, што доприноси већој рационализацији наставног процеса, подстиче интелектуалну активност ученика и наставу чини интересантнијом и ефикаснијом. Препоручује се коришћење интерактивних метода, пројектне, проблемске и истраживачке методе, дискусије, дебате и др., да би ученици били што више ангажованом током наставе. Комбиновати на часовима различите облике рада као што су самостални рад ученика (по принципу један ученик – један рачунар), рад у паровима (два ученика истовремено и заједно решавају конкретне задатке), рад у мањим групама (почетна анализа и идеје за методе решавања), као и рад са целом групом када наставник објашњава, приказује, демонстрира и кроз дискусију уводи ученике у нове области. Избор метода и облика рада,

као и планирање активности ученика ускладити са наставним садржајем који треба реализовати на часу и предвиђеним исхода, али и са специфичностима одељења и индивидуалних карактеристика ученика. Предложени број часова по темама је оквирни, на наставнику је да процени потребан и довољан број часова по темама узимајући у обзир знања и вештине који ученици имају из претходног школовања и животног искуства. Предложени редослед тема није обавезујући за наставнике, већ само представља један од могућих модела, који наставник може прилагодити у складу са изабраним програмским језиком и методолошким опредељењем.

Ради лакшег планирања наставе даје се оријентациони предлог броја часова по темама.

- Основни алгебарски алгоритми (20 часова)
- Геометријски алгоритми (20 часова)
- Алгоритми над текстуалним датотекама (20 часова)
- Графовски алгоритми (30 часова)
- Одабрани алгоритми и структуре података (20 часова)

НАПОМЕНА: Предвиђена су два двочасовна писмена задатка са исправком (6 часова).

### II. ОСТВАРИВАЊЕ НАСТАВЕ И УЧЕЊА

Кроз прве четири теме се ученици обавезно упознају са основним концептима неколико значајних области алгоритмике, а у оквиру последње теме се оставља простор за проширивање и продубљивање знања кроз изучавање додатних, напреднијих, алгоритама и структура података, или по потреби за утврђивање знања које поједини ученици нису довољно утврдили у времену предвиђеном за то.

У оквиру теме **Основни алгебарски алгоритми** ученици треба да буду упознати са основним алгоритмима теорије бројева. Један од основних појмова теорије бројева је појам простих бројева и потребно је обезбедити да ученици науче основне алгоритме везане за деливост и за просте бројеве: одређивање делилаца

броја, испитивање да ли је број прост, одређивање свих простих бројева у неком интервалу (Ератостеново сито), растављање на просте чиниоце и примене. Алгоритми треба да буду имплементирани на ефикасан начин (одређивање делилаца, испитивање да ли је број прост и растављање на просте чиниоце у сложености  $O$  (корен из  $n$ ), а Ератостеново сито у сложености  $(n \log \log n)$ ). Кроз оптимизовање алгоритама теорије бројева подсетити ученике на тему анализе сложености алгоритама и примене општих техника за конструкцију ефикасних алгоритама (на пример, приликом испитивања да ли је број прост користи се одсецање у линеарној претрази, узастопни упити колико има простих бројева у датим интервалима се након Ератостеновог сита могу оптимизовати префиксним сумама и слично). Искористити ову тему и да се са слабијим ученицима поново обраде елементарне технике програмирања (петље и итерација, низови, функције). Приказати што више примена основних алгоритама у решавању проблемских задатака (на пример, растављање на просте чиниоце се може применити да би се одредила допуна броја до потпуног квадрата, да би се ефикасно израчунали број и збир делилаца броја, број узајамно простих бројева са датим бројем - тзв. Ојлерова функција и слично).

Још један значајан и често практично применљив проблем је проблем одређивања НЗД и НЗС два броја. Приказати ученицима Еуклидов алгоритам заснован на одузимању и заснован на дељењу. Приказати рекурзивну и итеративну варијанту и искористити прилику да се ученици још једном подсети теме рекурзије и ослобађања рекурзије.

У оквиру теме **Геометријски алгоритми** ученици упознати са применама елементарне аналитичке геометрије равни (пре свега рада са векторима) на решавање проблема рачунарске геометрије и рачунарске графике. Као увод у тему могу се урадити задаци у којима се анализирају дужи, квадрати и правоугаоници којима су стране паралелне координатним осама (ови задаци се могу решавати елементарним програмерским техникама и са веома скромним математичким знањем). Након тога могуће је прећи на скуп задатака у којима се троуглови, квадрати и правоугаоници налазе у општем положају (нпр. одређивање четвртог темена паралелограма на основу три позната темена, провера да ли тачка припада троуглу, провера да ли тачка припада паралелограму или произвољном конвексном многоуглу, одређивање пресечне тачке правих тј. дужи, одређивање пресека троуглова, четвороуглова, одређивање обима и површина троуглова, четвороуглова и конвексних многоуглова). Скренути пажњу на проблеме рада са бројевима у покретном зарезу и нумеричких грешака које том приликом настају. Приказати што више практичних задатака у којима се примењују технике које се обрађују у овој области (на пример, израчунавање положаја тела у рачунарској графици).

У циљу илустрације примене геометријских алгоритама у рачунарској графици са ученицима креирати неколико веома једноставних графичких апликација у којима се користе елементарни геометријски алгоритми (на пример, исцртавање и израчунавање обима и површине троугла нацртаног мишем, исцртавање квадрата чија је дијагонала нацртана мишем, бојење круга и троугла у различиту боју у зависности од тога да ли се миш налази унутар круга тј. унутар троугла, исцртавање стрелице тј. вектора ако су му познате координате крајњих тачака). Ученицима који раније нису креирали графичке апликације потребно је приказати само основне технике, док је са ученицима који су раније креирали графичке апликације тему могуће обрадити и мало дубље (на пример, креирати апликацију у којој се конвексни многоугао боји различито у зависности да ли се миш налази у његовој унутрашњости, апликацију у којој се боји пресек два четвороугла и слично).

Кроз тему **Алгоритми над текстом и текстуалним датотекама** приказати основне технике анализе текста и ниски карактера (стрингова). Подсетити ученике на библиотеке функције за рад са нискама (надовезивање ниски, издвајање подниски, претрагу подниски и слично).

Објаснити шта су и чему служе регуларни изрази, а на неком од многобројних сајтова за тестирање регуларних изрази, или у неком од едитора текста који их подржавају, показати како се користе. Увести основну нотацију за запис регуларних изрази и применити

је на проблеме проналажења шаблона унутар текста и провере да ли се унети текст уклапа у жељени шаблон (нпр. проверити да ли је унети текст исправан запис мејл адресе или пронаћи све датуме у тексту, анализирати и модификовати датотеку са титловима неког филма, анализирати „лог“-датотеку веб-сервера и слично). Употребити регуларне изразе и у програмима на програмском језику у коме постоји библиотечка подршка за то (избор језика је прилично широк). Ученицима је могуће приказати и текстуалне датотеке записане у неком од стандардних формата (json, yaml, xml, html, css, markdown, restructuredtext, ...) и њихову анализу коришћењем специјализованих парсера у програмском језику и регуларних изрази.

Показати како неки језички конструкти (нпр. идентификатори, цели и реални бројеви, изрази са изабраним операцијама, са или без заграда, полиноми над једном или више променљивих, наредбе додељивања и сл.) могу да се рашчлане техником рекурзивног спуста. Ради лакшег савладавања технике, пожељно је да се за парсирање сваког под-конструкта, тј. синтаксичке категорије (идентификатор, број, сабирак/терм, чинилац/фактор, израз) користи посебна функција, односно метод. Правила по којима се сложенији конструкти граде од једноставнијих могу да се опишу усмено на основу искуства, или формално, нпр. Бекусовом нотацијом или синтаксичким дијаграмима. Могући примери примене ове технике су провера исправности записа изрази, израчунавање вредности аритметичких изрази или логичких формула и сл.

Пожељно је да се програмерске технике уведене у овој теми примене на скупове текстуалних датотека (фајлова) које могу бити разасуте у некој структури директоријума (фолдера), где је потребно пронаћи захтеване податке и једноставно их обрадити (нпр. ако свака датотека представља оцене једног ученика из свих предмета у неком договореном формату, одредити средњу оцену одељења из неког, или сваког предмета).

Кроз тему **Графовски алгоритми** упознати ученике са појмом графа (који се паралелно обрађује и у оквиру изучавања дискретне математике) и са имплементацијом основних графовских алгоритама. Дефинисати оријентисане, неоријентисане и тежинске графове. Описати и неке специјалне класе графова (дрвета, потпуне графове, ацикличне графове, ...). Ученицима приказати основне начине репрезентације графа у програму (матрица повезаности, листе суседа, ...). Приказати поступке елементарне анализе графова (на пример, одређивање улазних и излазних степена свих чворова). Детаљно проучити алгоритме за обилазак графова и њихове примене. Приказати рекурзивну имплементацију обиласка графа у дубину (алгоритам DFS). Приказати нерекурзивну имплементацију обиласка графа у дубину уз помоћ стека и нерекурзивну имплементацију обиласка графа у ширину уз помоћ реда (алгоритам BFS). Приказати мноштво примена алгоритама за обилазак графова (нпр. одређивање компонентата повезаности, тополошко сортирање, ...) и указати на предности обиласка у ширину у проблемима у којима се тражи најкраћи пут (нпр. најмањи број преседања да се стигне на одредиште). Приказати и Канов (Kahn) алгоритам тополошког сортирања (налажењем чворова који немају улазних грана, уклањањем њихових излазних грана итд.).

Приликом обраде ових тема обезбедити да ученици добро разумеју како функционишу графовски алгоритми, што наставник проверава и тако што ученик приказује кораке који се спроводе током извршавања алгоритма. Ученици могу приказати рад алгоритма било на папиру, било у склопу неког специјализованог едукативног софтвера који пружа ту функционалност.

У оквиру теме **Одабрани алгоритми и структуре података** може се проширити једна или више претходно обрађених тема. Теме се могу проширити одмах након обраде или на крају, када све теме буду обрађене. Алгоритме и структуре података који ће бити обрађени у овој теми бира наставник, при чему наставник има слободу да по својој процени одабере и неку другу тему из области програмирања и алгоритмике (на пример, из званичног програма такмичења). Могуће је индивидуализовати наставу и сваком ученику (или групама ученика) понудити теме које му највише одговарају. Ова тема може да се искористи и за систематизацију градива и утврђивање елементарнијих алгоритамских тема, које ученици нису у довољној мери савладали.

Из области алгебарских алгоритама, ученицима је могуће приказати и напредније алгоритме и њихове примене, као што су проширени Еуклидов алгоритам, одређивање модуларног инверза, алгоритам заснован на кинеској теореми о остацима. Још једна значајна група алгебарских алгоритама која се опционо може обрадити су алгоритми за рад са полиномима и великим бројевима (бројевима репрезентованим помоћу низа цифара). Свим ученицима је могуће приказати елементарну имплементацију основних операција над овим структурама података, а напреднијим ученицима је могуће приказати и ефикаснији Карацубин алгоритам множења заснован на техници подели-па-владај.

Из области геометријских алгоритама, могуће је приказати ученицима и напредније геометријске алгоритме, попут, на пример, одређивања конвексног омотача, ефикасног одређивања свих пресека у скупу дужи (line sweep алгоритам), одређивање пресека многоуглова и слично. Код напреднијих алгоритама детаљно анализирати сложеност.

Из области графовских алгоритама могуће је приказати и друге графовске алгоритме (нпр. Примов и Краскелов алгоритам, Дајкстрин алгоритам, Флојд-Варшалов алгоритам и слично).

Из области алгоритама за обраду текста ученицима је могуће приказати и ефикасне алгоритме претраге подниски (нпр. Кнут-Морис-Прагов, КМП алгоритам, Бојер-Муров алгоритам, Рабин-Карпов алгоритам и слично).

Израда пројектних задатака није обавезна у склопу овог предмета, али наставник може заинтересованим ученицима понудити теме пројектних задатака и оценити и ту њихову активност. Теме које се обрађују у склопу овог предмета се могу користити у склопу израде пројектних задатака из предмета „Објектно-оријентисано програмирање”.

### III ПРАЋЕЊЕ И ВРЕДНОВАЊЕ НАСТАВЕ И УЧЕЊА

У процесу вредновања потребно је континуирано пратити рад ученика. У настави оријентисаној на достизање исхода вреднују се и процес и продукти учења. Прикупљање информација из различитих извора (свакодневна посматрања, активност на часу, учествовање у разговору и дискусији, самосталан рад, рад у групи, тестови) помаже наставнику да сагледа постигнућа (развој и напредовање) ученика и степен остварености исхода. Свака активност је добра прилика за процену напредовања и давање повратне информације. Важно је и ученике оспособљавати и охрабривати да процењују сопствени напредак у учењу.

У процесу праћења и вредновања значајну улогу имају домаћи задаци. Редовно задавање домаћих задатака (уз обавезну повремену проверу од стране наставника), помаже наставнику да стекне бољи увид у степен остварености исхода кроз анализу задатака које ученици нису умели да реше. Важно је и мотивисати ученике који редовно раде домаће задатке тако што ће њихов рад бити оцењен.

Вредновање активности у оквиру тимског рада се може обавити са групом тако да се од сваког члана тражи објашњење елемената урађеног рада и мишљење о сопственом раду унутар тима.

Препоручује се да наставник са ученицима договори показатеље на основу којих сви могу да прате напредак у учењу, ученици се уче да размишљају о квалитету свог рада и о томе шта треба да предузму да би свој рад унапредили. Оцењивање тако постаје инструмент за напредовање у учењу. На основу резултата праћења и вредновања, заједно са ученицима треба планирати процес учења и бирати погодне стратегије учења.

Препоручено је да коначна оцена за сваког ученика буде добијена комбиновањем различитих начина оцењивања:

- активност на часу, учествовање у разговору и дискусији;
- редовна израда домаћих задатака;
- тестови - провера знања;
- пројектни рад, и појединачни и тимски.

Комбиновање различитих начина оцењивања помаже да се сагледају слабе и јаке стране сваког ученика. Приликом сваког вредновања постигнућа потребно је ученику дати повратну информацију која помаже да разуме грешке и побољша свој резултат и учење. Потребно је да наставник резултате вредновања постигнућа својих ученика континуирано анализира и користи тако да примени део своје наставне праксе.

### ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ

Циљ учења Објектно оријентисаног програмирања је стицање основних знања о објектно оријентисаној парадигми и њеној примени у решавању практичних проблема, развијање апстрактног и критичког мишљења и оспособљавање за примену стечених знања и вештина у даљем школовању и будућем раду.

### ОПШТА ПРЕДМЕТНА КОМПЕТЕНЦИЈА

Учењем наставног предмета Објектно оријентисано програмирање ученик је оспособљен да примени стечена знања и вештине из области информационо-комуникационих технологија ради испуњавања постављених циљева и задатака у свакодневном животу, даљем школовању и будућем раду. Развио је способност апстрактног и критичног мишљења уз помоћ информационо-комуникационих технологија. Развио је дигиталну писменост и позитивне ставове према рачунарским наукама.

### СПЕЦИФИЧНЕ ПРЕДМЕТНЕ КОМПЕТЕНЦИЈЕ

Специфичне предметне компетенције представљају опис специфичних способности ученика које му омогућавају да развије општу предметну компетенцију. Подразумевају способност за одговорно коришћење информационо-комуникационих технологија уз препознавање потенцијалних ризика и опасности. Специфичне компетенције обухватају способност да се разуме и примени начин решавања практичних проблема применом објектно оријентисане парадигме.