|  |
| --- |
| **РАЧУНАРСТВО И ИНФОРМАТИКА** |
| Циљ учења *рачунарства и информатике* је стицање знања, овладавање вештинама и формирање вредносних ставова који доприносе развоју информатичке писмености неопходне за даље школовање, живот и рад у савременом друштву. Усвајањем концепата из рачунарских наука, ученик развија способност апстрактног и критичног мишљења о аутоматизацији послова уз помоћ информационо-комуникационих технологија и развија способност ефективног коришћења технологије на рационалан, етичан и безбедан начин. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ОПШТА ПРЕДМЕТНА КОМПЕТЕНЦИЈА Учењем наставног предмета *рачунарство и информатика* 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 КОМПЕТЕНЦИЈЕСпецифичне предметне компетенцијепредстављају 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; способност писања програма вођених догађајима и разумевање принципа креирања добро структуираних програма. Специфичне компетенције обухватају способност за брзо, ефикасно и рационално проналажење информација коришћењем рачунара, као и њихово критичко анализирање, складиштење и преношење и представљање у графичком облику.

|  |  |
| --- | --- |
| Разред | **Први** |
| Годишњи фонд часова | **74 часа** |

 |
| **ИСХОДИ**По завршетку разреда ученик ће бити у стању да: | **ОБЛАСТ/ТЕМА** | **САДРЖАЈИ** |
| * објасни улогу ИКТ у свакодневном животу
* разуме изазове коришћења савремених тенологија на одговоран и безбедан начин
* кратко опише најважније догађаје у развоју ИКТ
* разликује и користи сервисе Интернета
* приступа Интернету, самостално претражује, проналази информације у дигиталном окружењу и преузима их на свој уређај
* класификује информације са интернета и процењује њихов квалитет и поузданост
* спроводи поступке за заштиту личних података и приватности на Интернету
 | **Информационо - комуникационе технологије у савременом друштву** | ИКТ у свакодневном окружењу (уређаји, облици комуникације, услуге)Развој ИКТ (прикупљања, складиштења, обраде, приказивања и преноса података)Друштвени аспекти ИКТ  (значај и примена дигиталних уређаја, карактеристике информационог друштва, утицај коришћења дигиталних уређаја на здравље и околину, интелектуална својина, безбедност, заштита личних података, правила понашања)Глобална мрежа (интернет)Сервиси интернета (електронска пошта, веб, друштвене мреже, блогови, форуми, учење, мапе, електронска трговина и банкарство, аудио и видео комуникација)Лепо понашање, право и етика на интернету. Безбедност и приватност на интернету |
| * објасни начин дигиталног записа података и бинарног записа природних бројева
* користи јединице за мерење количине података
* кратко опише разлику између хардвера и софтвера
* наводи основне карактеристике компонената дигиталног уређаја и њихову улогу
* разликује системски од апликативног софтвера
* објасни шта је оперативни систем и која је његова улога
* познаје основне типове апликативног софтвера
* разликује појмове и типове лиценци софтвера и садржаја који се деле
 | **Рачунарство** | Дигитални рачунари и дигитални запис података (текста, растерске и векторске слике, звука) Начини приказивања/представљања података и дигиталног записаХардверске и софтверске компоненте рачунарских система |
| * разликује основе елементе графичког корисничког интерфејса
* прилагоди радно окружење кроз основна подешавања
* инсталира и деинсталира корисничке програме
* сачува, модификује и организује податке
* разликује најчешће коришћене типове датотека
 | **Организација података и прилагођавање радног окружења** | Елементи графичко-корисничког интерфејса и интеракција са њима (радна површина, прозори, менији, дугмад, акције мишем или акције на екрану осетљивом на додир, пречице на тастатури, ...).Подешавања оперативног система (подешавање датума и времена, радне површине, регионална подешавања, подешавања језика и тастатуре, коришћење и подешавање корисничких налога).Инсталирање и уклањање програма (апликативних програма, драјвера).Рад са документима и системом датотека.Средства и методе заштите рачунара и информација. |
| * ефикасно и тачно уноси и уређује неформатиран текст
* примењује основне елементе форматирања и структуирања текста
* уређује на елементарном нивоу текст применом нотација за обележавање
* постави напредне текстуалне и нетекстуалне елементе у креирани документ
* познаје основне параметре стилизовања текста на нивоу карактера, параграфа и страница
* користи и креира именоване стилове
* користи елементе у тексту који се аутоматски ажурирају
* припреми документ за штампу и одштампа га
* уређује и приказује слајд презентације
* примењује правила за израду добре презентације.
* користи функционалнoсти намењене сарадничком раду
 | **Креирање и уређевање дигиталних докумената** | Унос текста и његово једноставно уређивање (ефикасно кретање кроз текст, копирање, премештање, претрага, замена текста).Форматирање и обликовање текста (странице, пасуса, карактера).Посебни елементи у тексту (листе, табеле, слике, математичке формуле, ...).Обележавање текста (MarkDown нотација и елементарне LaTeX формула)Коришћење и израда стилова, генерисање садржаја.Презентације и њихова примена (правила добре презентације, етапе у изради презентација).Креирање слајдова (уметање и форматирање текста, графикона, слика, звучних и видео-записа, ...).Штампање докумената. |
| * описује алгоритмом ситуације из реалног живота (говорним језиком, псеудокодом, дијаграмом)
* креира једноставан рачунарски програм у развојном окружењу
* користи изразе за запис математичких формула
* примењује наредбе за контролу гранања и понављања
* анализира програм и предвиђа његово понашање без покретања
* проналази и отклања грешке у програму
* креира програм који реализује једноставне интерактивну 2д графику
* креира програм у текстуалном програмском језику
* разуме и отклања синтаксне грешке у програмском коду
 | **Програмирање** | Појам алгоритма, структура алгоритма и начини описивања алгоритамаУвод у алгоритме и програмирање:* аритметичка израчунавања (изрази, константе, променљиве, операције, основне уграђене функције)
* гранање
* понављање
* дефинисање функција
* коришћење колекција

2д цртање уз помоћ графичких примитива:* увод у функионалност одабране графичке библиотеке (координатни систем, платно, оловке, четкице, ...)
* цртање основних облика (дуж, квадрат, круг)
* учитавање и приказ слике из датотеке
* цртање правилних облика са понављајућим елементима

Анимација и интерактивна 2д графика (реаговање на догађаје):* програмирање анимација (понављањем исцртавања облика у правилним временским интервалима)
* концепт догађаја (догађаји миша и тастатуре) и обрада догађаја.
 |

**Кључни појмови садржаја**: информатика и рачунарство, хардвер, софтвер, датотека, фасцикла, обрада текста, слајд презентације, рачунарска мрежа, интернет сервиси, безбедност на интернету, алгоритам, рачунарски програм.

**УПУТСТВО ЗА ДИДАКТИЧКО-МЕТОДИЧКО ОСТВАРИВАЊЕ ПРОГРАМА**

Настава се изводи у двочасу, са половином одељења у рачунарском кабинету,  у групама не већим од 15 ученика.

На почетку наставе урадити проверу нивоа знања и вештина ученика, која треба да послужи као оријентир за организацију и евентуалну индивидуализацију наставе.

При реализацији програма дати предност пројектној, проблемској и активно оријентисаној настави, кооперативном учењу, изградњи знања и развоју критичког мишљења. Уколико услови дозвољавају дати ученицима подршку хибридним моделом наставе (комбинацијом традиционалне наставе и електронски подржаног учења), поготово у случајевима када је због разлика у предзнању потребна већа индивидуализација наставе.

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.Препорука је да наставник, у зависности од могућности ученика и рачунарске опреме, процени и комбинује у току сваког двочаса различите наставне методе и 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 дискусују уводи ученике у нове области.

**Информационо-комуникационе технологије у савременом друштву (8 часова)**

Потребно је нагласити значај ИКТ, али и да коришћење доноси различите ризике и одговорност. Кроз ученицима познате примере навести примере одговорног и безбедног коришћења ИКТ (иако ће се ова тема провлачити током целог школовања, како ученици овладавају алатима и применом ИКТ у свом животу, наглашавати безбедност и одговорност при коришћењу истих).

Ученике укратко упознати са историјатом развоја ИКТ, рачунских справа и рачунара, не инсистирајући на детаљима (тачним годинама, прецизним карактеристикама уређаја и слично). Ученици треба да буду свесни када се јавиле идеја о рачунарима који се могу програмирати, када су настали први електронски рачунар, када су настали персонални рачунари и како изгледају савремени рачунари у односу на почетке (“од рачунара који заузима целу зграду до уређаја у џепу сакоа”). Са ученицима дискутовати и могуће правце развоја ИКТ у будућности.

Ученицима нагласити да су модерни уређаји данас нераскидиви део Интернета и обрнуто. Потребно је да ученици имају представу о рачунарским мрежама и да јасно разликују локалну мрежу и Интернет. Потребно је направити паралелу између кућне мреже и мреже у школи и скренути пажњу да су за формирање и функционисање мреже потребни и посебни урађаји и програми, без уласка у детаљну анализу њихове улоге и технолошких карактеристикама. Са ученицима дискутовати о сервисима на Интернету и веб-апликацијама које користе и подстакнути их да једни другима укажу на корисне и интересантне сервисе и апликације. Посебну пажњу посветити претраживању информација на интернету и процени њихове поузданости и релевантности. Ова тема треба да буде практично демонстрирана и прожета током читавог наставног процеса.

Такође у овој фази треба да остваре прву комуникацију са наставником користећи мејл (упутити их, уколико не знају, како се пише мејл, елементе поруке, проверити како су активирали налоге на друштвеним мрежама - безбедност и приватност мејла, контаката и садржаја сандучета). Посебну пажњу скренути на прилоге мејлова (шта се може, а шта не може слати мејлом). И у случају мејл-комуникације инсистирати на безбедно и одговорно коришћење уз поштовање правила лепог понашања (нетикеција).

**Рачунарство (10 часова)**

При реализацији ове тематске целине потребно је да ученици стекну знања о томе како се у дигиталним уређајима информације представљају помоћу бројева. Треба да знају да се кодирањем представљају текст, графика и звук. Приказати ученицима како се природни бројеви представљају у бинарном систему (нулама и јединицама). На информативном нивоу показати хексадекадни систем као скраћење записа бинарних бројева и приказати бинарни запис неких података (на пример, текста записаног ASCII кодом).

Ученици треба да познају јединице за мерење количине података (бит, бајт, килобајт, ...) и да умеју да процене колико уобичајени подаци заузимају меморије (нпр. колико отприлике заузима страница текста, фотографија ниске и високе резолуције, филм и слично).

Ученици треба да знају основну структуру рачунара (процесор, меморије и улазно-излазни уређаји, као и комуникацију између њих). Ученици би требало да умеју да објасне чему нека компонента служи и које су њене главне особине, при чему треба да знају: улогу процесора у функционисању рачунарског система (да познају особине процесора, да објасне врсте и улогу различитих меморија у рачунарима (меморије које трајно и привремено памте податке) и да разликују унутрашње меморије (кеш, RAM) од спољашњих, складишних меморија (хард-дискова, флеш-меморија, SSD уређаја, оптичких дискова).

Инсистирати на хијерархијској организацији меморија и објаснити разлику у брзини, капацитету и цени различитих облика меморија (особине меморија); основне врсте улазно-излазних уређаја и начине комуникације са њима; врсте магистрала и њихову улогу у остваривању комуникације између различитих компонената унутар рачунара. Ученик компоненте треба да зна на нивоу препознавања, без улажења у детаље њихове архитектуре и начина функционисања.

Искуствено ученици могу описати улогу оперативних система, и уочити разлику између хардвера и софтвера. Ученици треба да знају разлику између апликативних и системских програма, као и различиту примену апликативних програма у свакодневном животу (на пример, програме за приступ интернету и вебу, рачунарске игре, програме за обраду звука, …..). Посебну пажњу посветити ауторским правима, интелектуалној својини и типовима лиценци софтвера и дељених садржаја. Део тематске целине чији је фокус на зашити аутоских права и коришћењу туђег садржаја треба да се прожима кроз све тематске целине кроз сва четири разреда.

**Организација података и прилагођавање радног окружења (6 часова)**

Извршити систематизацију основних концепата како би се увела заједничка терминологија и како би се обезбедило да ученици мало дубље разумеју основне концепте графичких радних окружења тј. њихових корисничких интерфејса. Истовремено дискутовати графичко окружење стоних и преносних рачунара и мобилних уређаја, набројати  сличности, али и нагласити разлике.

Са ученицима систематизовати знање о елементима графичког корисничког окружења: радној површини, прозорима, менијима, дугмадима, пољима за унос текста и слично. Обезбедити да ученици ефикасно баратају основним улазним уређајима тј. да умеју да изведу акције мишем, екраном осетљивим на додир, али и пречицама на тастатури. Обезбедити да ученици разумеју концепте селекције, концепт клиборда и њихову примену на копирање и премештање података. Ученици треба разумеју и да знају да одреагују на разне поруке које добијају од система током рада (на пример, при брисању података, затварању програма, чувању документа…).

Систематизовати са ученицима и основна системска подешавања (датума и времена, радне површине, регионална подешавања, подешавања језика и тастатуре, коришћење и подешавање корисничких налога).

Објаснити, кроз неколико примера инсталацију и уклањање програма (опет направити паралелу стоних и преносивх рачунара са мобилним уређајима).

Паралелно са радом на организацији података на систему датотека оперативног система демонстрирати манипулисање подацима на „облаку“. Дискутовати о предностима и недостацима манипулације података оба начина. Потребно је да ученици знају када податке чувају на диску, на некој преносивој спољној меморији, на телефону, „у облаку“... Потребно је појаснити терминологију (фајл-датотека, фолдер - фасцикла - директоријум - каталог, партиција, диск), и обезбедити да ученици разумеју концепт датотека и фасцикли и њихову примену на хијерархијско организовање података. Ученици треба да познају најпознатије типове датотека, да знају да искључе/укључе приказ типа датотеке и скривених датотека, да знају да су одређени типови датотека повезани са подразумеваним програмима који их отварају, као и да та повезивања подесе. Кроз рад на документима и фасциклама инсистирати на начинима како се дели и приступа фасциклама и датотекама на “облаку” (сарадња, само да прегледају документе....).  Потребно је да ученици разумеју хијерархијску организацију система датотека и путање које одређују позицију (тј. адресу) датотеке у систему. Ученике упознати и са „пречицама“ тј. симболичким линковима ка датотекама. Упознати ученике са неким програмима за архивирање података и потребом за таквим програмима (вежба слање мејла али са архивираним подацима).

Упознати ученике са методама и значајем заштите података, подешавањем антивирусног програма, заштитног зида.

**Креирање и уређевање дигиталних докумената (20 часова)**

При реализацији ове тематске целине инсистирати да ученици науче да вешто и ефикасно врше уношење текста строго придржавајући се дигиталног правописа (у латиничком тексту на српском језику користећи дијакритичке карактере č, ć, ž, š, и сва граматичка правила говорног језика). За почетак вежбати рад са чистим текстом, без уметнутих нетекстуалних елемената и без икаквог програмирања. Осим уношења текста, треба нагласити да су основни кораци у раду са текстом кретање кроз текст, копирање, премештање делова текста, претрага и замена. Ученици треба да користе пречице на тастатури за различите операције са текстом (копирање, премештање, претрага, ….). Ученик приликом уноса текста треба јасно да зна како се текст дели на целине - параграфе и скренути пажњу на разлику између експлицитног уметања ознака за нови ред и прелаза у нови ред које едитор текста аутоматски приказује (а који нису унети у текст). Нагласити ученицима да је овај основни ниво рада са текстом заједнички за веома широку палету програма (од најједноставнијих едитора текста до напредних процесора текста) и демонстрирати рад у неколико различитих програма (на локалном рачунару, али и онлајн), укључујући и програмерске едиторе које ће ученици касније програмирати.

Након рада са чистим текстом, прећи на обраду уметања нетекстуалних елемената и структурирање текста. Ученик треба да зна да организује текст коришћењем нумерисаних и ненумерисаних листа, да у текст уметне нетекстуалне елементе (табеле, слике, једноставне математичке симболе и формуле...). И ову тему приказати кроз неколико различитих програма (нпр. текст процесора, програма за слање електронске поште и креирање онлајн документа или веб стране) и истаћи заједничке карактеристике.

Објаснити разлику између приступа уређивања текста коришћењем приступа “што видиш то и добијеш” (*WYSIWYG*) и језика за обележавање. Приступ језика за обележавање могуће је демонстрирати кроз “лаке језике за обележавање” (нпр. *MarkDown*), кроз *Wiki* нотацију и подскуп језика *HTML* (довољно је приказати означавање наслова, секција, пасуса, нумерисаних и ненумерисаних листа и евентуално убацивање слика и линкова). Кроз неколико елементарних примера приказати употребу *LaTeX* синтаксе за запис математичких формула (индекс, експонент, разломак, грчка слова, ...). Улогу језика за обележавање приказати на примерима њихове употребе на вебу. Кроз језике за обележавање вежбати са ученицима прецизно изражавање у формалној нотацији и поштовање прецизне синтаксе, што представља важан увод у тему програмирања у текстуалним програмским језицима.

Потребно је објаснити разлику између логичке структуре документа и његовог визуелног и стилског обликовања и увести стилове као основну технику логичког структурирања документа. Ученик треба да уме да ефикасно користи постојеће, прилагођава именоване стилове и креира сопствене стилове на нивоу карактера, параграфа и странице. Поред тога потребно је да овладају елементима за аутомаско ажурирања документа (садржај, листа свих табела или слика, странице...). На крају рада са документом ученик треба да зна да документ припреми за штампање и одштампа га на папир или извезе у формат *PDF*. Коришћењем програма за креирање слајд презентација ученици треба да примене већ овладане технике форматирања и стилизовања текста и креирају добру и ефективну презентацију. При изради слајд-презентације ученик мора да се приржава правила добре презентације (број информација на слајду, дизајн слајда, естетика, анимације у служби садржаја, ….). Нагласити важност израде сарадничког документа приликом рада на неком тексту или слајд - презентацији.

**Програмирање (30 часова)**

Пре почетка наставе програмирања ученицима треба објаснити важност алгоритамског приступа решавању проблема и вештине програмирања у данашњем свету за све, па и за ученике који се неће професионално бавити програмирањем. Ученике треба упознати са позитивним утицајем наставе програмирања на развој прецизног, формалног изражавања и на развој логичког мишљења. Често се у животу дешава да треба да организујемо неку групу људи и да им објаснимо како да ураде неки посао, а познавање техника описивања алгоритама ће нам помоћи да ту увек урадим на јасан и недвосмислен начин. ИКТ уређаји нас често збуњују чудним понашањем, које се испоставља као сасвим природно и нормално, ако разумемо строге принципе по којима функционишу. Кроз серију једноставних примера приказати могућност употребе техника алгоритамског размишљања и програмирања на аутоматизовање практичних задатака са којима се ученици и иначе срећу током образовања и личног и касније професионалног живота. Навести ученицима колико познавање принципа алгоритама и програмирања може помоћи скоро у свакој професији (на пример, будући психолози и социолози користе напредан статистички софтвер за обраду својих података и да им елементи програмирања могу помоћи да прилагоде тај софтвер потребама своји специфичних анализа, а будући стручњаци за језик могу помоћу релативно једноставних програмерских техника извући потребне делове језичких корпуса и статистички их обрађивати).

У складу са претходним, реализацију теме могуће је започети увођењем појма алгоритма кроз навођење примера из свакодневног живота у којима се одређене радње дешавају по унапред задатим правилима (на пример, кухињски рецепти, организација летовања коришћењем интернета, одређивање победника у игри папир-камен-маказе, слагање Рубикове коцке, слагање Ханојских кула, погађање непознатог броја половљењем интервала). Навести и примере где се као корисници апликације сусрећемо са појмом алгоритма, као што је алгоритам проналажења руте у апликацији за навигацију где можемо и подешавати како алгоритам ради. Приказати и примере алгоритама са којима су се ученици сусретали у настави математике (извођење аритметичких операција потписивањем, конструкције геометријских објеката, НЗД, ...). Анализирати са ученицима више различитих решења истог проблема.

Дати примере исказивања алгоритама говорним језиком, дијаграмима тока и псеудокодом. Ученицима разјаснити шта се сматра елементарним кораком у алгоритму, како се кораци секвенцијално могу ређати један иза другог, како се врши условно извршавање одређених делова алгоритма (гранање) и како се одређени делови алгоритма могу понављати (петље тј. циклуси). Увести и могућност скока (тзв. *GOTO*) на обележени корак (на пример, на број корака у алгоритму представљеном у облику нумерисаног низа корака) и укратко на примеру продискутовати мане таквих решења. Као илустрацију изабрати неки једноставан алгоритам који ће се исказати на разне начине,  укључујући, на крају и програмом у програмском језику у коме ће се даље радити. Пре него што ученици самостално крену да формулишу алгоритме, подстицати их да разумевају смисао алгоритама које наставник припрема. Радити на развијању свести код ученика о томе да решење проблема треба да буде образложено јасно и прецизно (најбољи тест је то да је алгоритам описан тако да га други ученик може спровести без икаквих додатних упутстава).

 Након кратке уводне приче о алгоритамском начину размишљања потребно је прећи на програмирање у одабраном програмском језику. Тема програмирања се може у целини реализовати уз коришћење текстуалног програмског језика, а може се започети са визуелним програмским језиком заснованим на слагању блокова, а завршити са текстуалним програмским језиком. Изабран текстуални програмски језик треба да буде једноставан за разумевање и за изражавање кратких програма, као и да има уграђену подршку за рад са колекцијама кроз одговарајуће могућности синтаксе и стандардне библиотеке. Развојно окружење за текстуални програмски језик треба да садржи едитор који је прилагођен синтакси програмског језика и да омогући покретање програма једноставном корисничком акцијом. Пожељно је да развојно окружење омогућава да мањи програми буду садржани у једној јединици изворног кода без потребе креирања пројекта. За реализацију теме се може користити развојно окружење инсталирано на локалном рачунару или онлајн развојно окружење коме се приступа путем прегледача веба (браузера).

Редослед увођења појмова, елемената програмских језика, типова података, библиотечких операција/функција/метода, алгоритама, техника програмирања и функционалности развојног окружења треба прилагодити изабраним програмским језицима, окружењима и методолошким опредељењима.

У склопу обраде ове теме неопходно је покрити следеће елементе одабраног програмског језика:

* аритметичке изразе (константе, променљиве, операторе и коришћење неколико основних библиотечких функција)
* однос између целобројног и реалног дељења
* основне типове и елементарне конверзије између њих, и то: бројевне типове (целобројни и реални), текстуални (ниске тј. стринг) и логички тип
* учитавање и испис вредности основних типова
* услове изражене коришћењем релацијских и логичких оператора и наредбе гранања у непотпуном облику (if) и потпуном облику (if-else)
* наредбе понављања (бројачку петљу, условну петљу)
* дефинисање и коришћење једноставних потпрограма (функција, процедура и сл., у зависности од програмског језика, односно статичких метода, уколико програмски језик не нуди непосреднији облик дефинисања потпрограма)
* елементарно коришћење колекција података (низови, ниске, торке, речници, ...), приступ појединачним елементима колекције, библиотечке функције/методе над колекцијама (број елемената, збир, просек, минимум, максимум, ...), итерација кроз елементе колекције

Не инсистирати на појму прекорачења, прецизности записа реалних бројева и на детаљима интерне репрезентације бројева (стога је, ако је могуће, пожељно одабрати програмски језик који аутоматски подржава рад са неограниченим бројевима). Променљиве у почетку третирати на исти начин као у математици и не мењати вредности једном додељеним променљивима, а императивну доделу, попут i=i+1, која је ученицима често тежак појам, посебно обрадити и разјаснити кроз примере у којима је таква додела заиста потребна.

Алгоритамска структура задатака у првом разреду не би требало да буде комплексна. Кроз разне задатке могуће је приказати следеће типове алгоритама и програма:

* аритметичка израчунавања (нпр. геометријске формуле, формуле из физике, линеарне зависности и пропорције, решавање проблема коришћењем линеарних једначина итд.)
* примена минимума, максимума и апсолутне вредности разлике као мере растојања (нпр. одређивање дужине пресека интервала)
* рад са позиционим записом (нпр. цифре у запису броја, време, углови)
* угнежђено гранање (каскадно и хијерархијско): гранање на основу припадности вредности бројевним интервалима (нпр. одређивање оцене на основу броја поена), гранање на основу дискретног скупа вредности (нпр. одређивање имена месеца на основу редног броја), лексикографско поређење торки (нпр. поређење два датума),  хијерархијско гранање (нпр. одређивање квадранта на основу датих координата тачака).
* статистичка обрада колекција података (израчунавање збира, просека, минимума и максимума), филтрирање (издвајање елемената који задовољавају дати услов) и пресликавање (примена функције на све елементе колекције), линеарна претрага колекција података и комбинација ових алгоритама
* сортирање применом библиотечких функција и решавање проблема помоћу сортирања (не укључује познавање и имплементацију алгоритама сортирања)

Нагласак учења програмирања у првом разреду треба да буде на стицању течности у елементарном кодирању, а не на вештини конструкције и анализе сложености алгоритама. Стога се сугерише да се у што већој мери примењују библиотечке функције и да се ручна имплементација алгоритама примењује само где је то неопходно (на пример, одређивање максимума неколико бројева могуће је у почетку реализовати позивом библиотечке функције, а тек касније увести алгоритам одређивања максималне вредности и применити га у неколико задатака). Алгоритми које ученици самостално имплементирају не би требало да прелазе комплексност основних секвенцијалних алгоритама линеарне сложености за обраду серија елемената (нпр. одређивања збира и просека колекције бројева, или одређивање њеног минималног тј. максималног елемента).

Приликом избора задатака пожељно је трудити се да се текстови задатака формулишу тако да ти задаци сугеришу неку реалну примену било у стварном животу и доменима блиским ученицима (нпр. спорт, филм, музика, мода), било у другим наставним предметима (математика, физика, биологија, историја, географија и слично).

Инсистирати на пажљивом тестирању свих решења (ако је могуће, коришћењем аутоматског система тестова на више тест-примера).

Од ученика тражити да пишу своје једноставне програме, али и да пажљиво анализирају већ написане програме и да предвиде резултате њиховог извршавања и без њиховог покретања. Инсистирати на томе да сви ученици могу да спроведу задати алгоритам корак по корак, експлицитно записујући (на пример у облику табеле) стање, тј. текуће вредности променљивих током извршавања алгоритма Ученицима приказати процес дебаговања (извршавања корак по корак уз анализу међурезултата) и захтевати од њих да пронађу и исправе намерно унете грешке у програме.

Не инсистирати на томе да ученици праве потпуно заокружене апликације (на пример, обрада теме креирања графичког корисничког интерфејса програма није предвиђена за први разред). Дужина програма које ученици пишу не би требало да прелази пар десетина линија кода. Сложеније проблеме разлагати на потпроблеме који се једноставније решавају (на пример, сваки помоћу посебне функције тј. потпрограма). Један од могућих наставних сценарија је да ученици имају задатак да напишу само одређене делове већег програма које наставник унапред припрема (да допуне неколико линија кода, дефинишу неку функцију и слично).

Један интересантан домен за вежбање основних техника програмирања је и 2д цртање и прављење анимација, као и једноставних интерактивних симулација (игара). Стога је пожељно користити језике и библиотеке који ово допуштају. Наставник се може определити да тему уведе и кроз овај домен и тада је могуће је реализовати и следеће теме:

* увод у функционалност одабране графичке библиотеке (координатни систем, платно, оловке, четкице, боје, ...)
* цртање основних облика (дуж, квадрат, круг, ...)
* цртање правилних облика са понављајућим елементима (нпр. екран ишрафиран хоризонталним/вертикалним/дијагоналним линијама, концентрични кругови у центру екрана)
* програмирање анимација (понављањем исцртавања облика у правилним временским интервалима, попут лоптице или неке друге фигуре која се одбија о ивицу прозора или слике лика који се шета лево-десно дуж екрана)
* концепт догађаја (догађаји миша и тастатуре) и обрада догађаја (померање једноставног објекта на екрану стрелицама, исцртавање кругова мишем)
* у оквиру пројектних задатака и додатне наставе могуће је обрадити и неколико примера програмирања веома једноставних игара (нпр. бушење мишем балона који се појављују на насумичним позицијама на екрану уз бројање резултата, померање лика стрелицама кроз препреке које се крећу, попут игре *flappy birds*).