

НАСТАВА ПО НОВИМ ПРОГРАМИМА У СРЕДЊОЈ ШКОЛИ

Душа Вуковић, Катарина Вељковић, Милена Марић, Бојана Сатарић

Апстракт: Предавање у оквиру ове теме биће подељено на три целине. Прве две целине посвећене су изменама планова и програма за 1. разред гимназија и обрађују тему Увод у алгоритме и блоковско програмирање, док је трећа целина посвећена новим наставним плановима и програмима за средње стручне школе.

Alice у првом разреду гимназија

У првој целини Душа Вуковић и Катарина Вељковић ће представити алат Alice (доступан на адреси <http://www.alice.org>). Alice је иновативно 3Д програмско окружење у којем је веома једноставно креирати анимације помоћу којих се прича нека прича, креира интерактивна игра или видео који се може поделити на вебу. Alice је слободно доступно наставно средство намењено за први контакт ученика са објектнооријентисаним програмирањем. Оно допушта ученицима да науче фундаменталне програмске концепте у контексту креирања анимираних филмова и једноставних видео-игара. У Alice, 3Д објекти (људи, животиње и возила) се налазе у виртуелном свету и ученици креирају програме којима анимирају те објекте. Помоћу интерактивног интерфејса, ученици превлаче графичке елементе (блокове) којима креирају програм при чему интрукције одговарају стандардним наредбама у правим програмским језицима (нпр. Java, C++ или C#). Alice омогућава ученицима да одмах виде како се њихови анимирани програми извршавају омогућујући има да уоче везе између наредби у програмима и понашања њихових објеката. Баратањем објектима у њиховом виртуелном свету, ученици стичу искуства о свим програмским конструкцијама које се обично уче у уводним курсевима програмирања.

Scratch у првом разреду гимназија

У другој целини Милена Марић ће представити алат Scratch (<https://scratch.mit.edu/>) и могућност његове употребе у почетној настави у првом разреду гимназије. Scratch је слободно доступни визуелни програмски језик развијен у лабораторији MIT Media Lab. Scratch користи велики број студената, наставника и родитеља како би једноставно креирали анимације, игре и слично. Он представља одскочну даску за улазак у напреднији свет рачунарског програмирања. Такође се може користити за креирање едукативних и забавних садржаја, израду математичких и научних пројеката, симулацију и визуализацију експеримената и слично. Могуће је и прегледање велике библиотеке постојећих пројеката на Scratch веб-сајту.

Scratch омогућава корисницима да користе програмирање вођено догађајима са више активних објеката који се називају спрајтови. Спрајтови се приказују као векторске и растерске слике, креиране и обрађене било у графичком едитору који је део система Scratch, било увезене из спољашњих извора (датотека, камера и слично). Scratch такође подржава и рад са звуком који може бити снимљен или увезен. Scratch је доступан било као веб-апликација, било као апликација за оперативне системе Windows, OS X или Linux.

У склопу ове целине, биће дат детаљан приказ реализације часова у оквиру нове наставне теме Увод у алгоритме, кроз велики број примера рађен у Scratch-у, прилагођен узрасту ученика и разрађен по појединачним часовима.

Средње стручне школе

У трећој целини Бојана Сатарић ће представити нове програме предмета Рачунарство и информатика у првим разредима средњих стручних школа. Биће дата упоредна анализа пређашњег и садашњег стања тј. старих и нових програма. Разматраће се питање како унапредити наставу и биће дате препоруке за реализацију нових програма. Такође, биће приказан низ примера добре праксе.

Alat Alice može da se preuzme sa sledeće adrese: **www.alice.org**

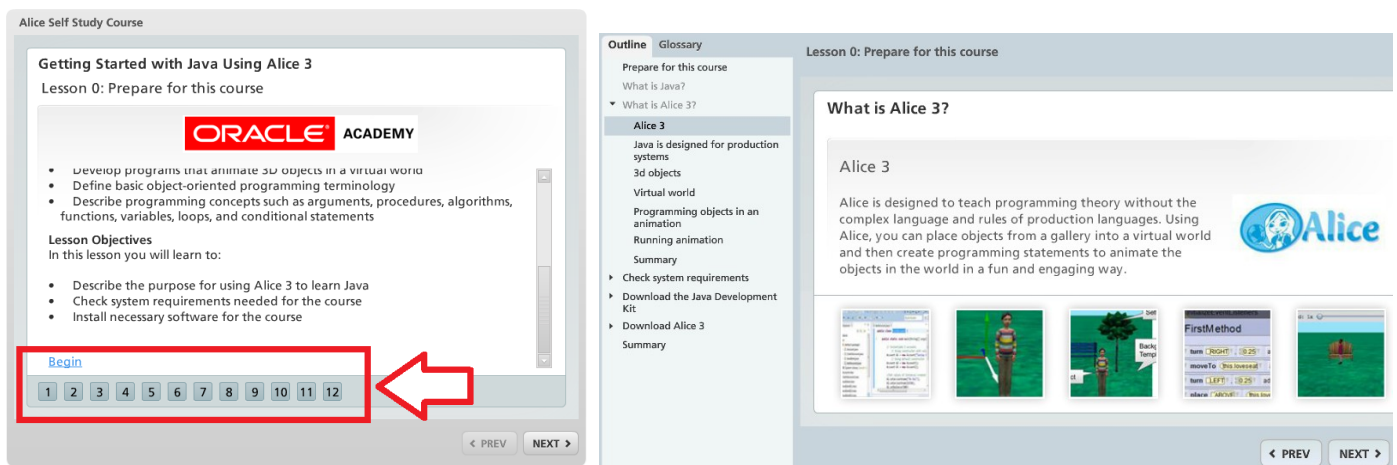
Kratka video uputstva posvećena alatu Alice:

- https://youtu.be/eq120m-_4uA
- <https://youtu.be/tkbUcu71Lfk>

Kurs (engleski) posvećen uvodu u Java programiranje u alatu Alice je dostupan na sajtu Oracle Akademije:

1. sajt Oracle Akademije - **<https://academy.oracle.com/>**
2. Izbor **Training**
3. Izbor **Self-Study**
4. klik na dugme **Getting Started with Java Using Alice Self-Study Course**

Kurs ima više lekcija koje su numerisane. Svakoј možemo direktno da pristupimo klikom na dugme sa rednim brojem lekcije koje se nalazi pri dnu strane (slika levo). Kada se klikne na dugme prikaže se pregled lekcije, a sama lekcija se učitava klikom na *Begin*. Kroz lekciju se prolazi redom pomoću spiska tema na meniju *Outline* sa leve strane, ili uz pomoć navigacije *PREV* i *NEXT* dole desno (slika desno).



Direktan link ka kursu:

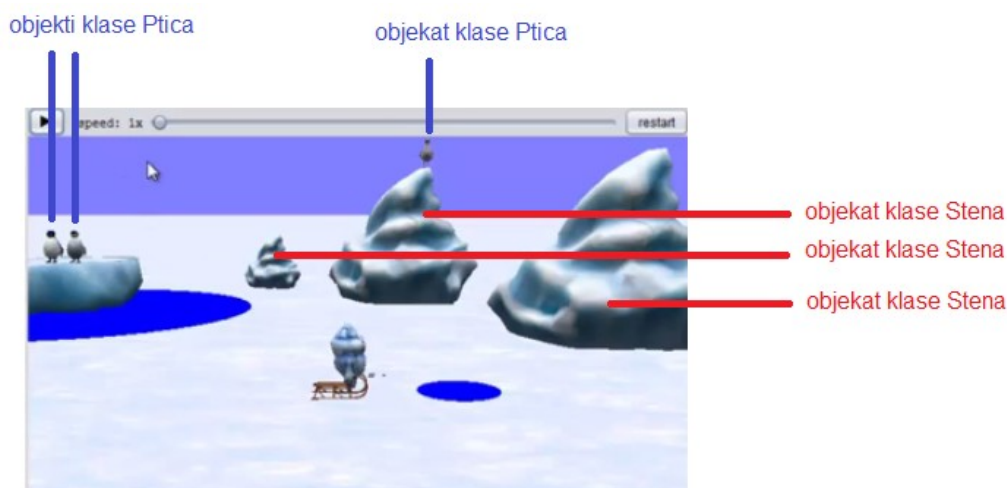
http://ilearningcontent.oracle.com/content/public/oracle_acad/SelfStudy/Articulate/Alice/interaction.html

Uvod u objektno-orijentisano programiranje

Aplikacije koje učenici poznaju kao korisnici računara su sastavljene od objekata. Razvoj aplikacija sa grafičkim korisničkim interfejsom je u vezi sa razvojem objektno-orijentisanog programiranja.

Kao prvi kontakt sa programiranjem je dobro učenicima pokazati kako su kreirane aplikacije koje liče na aplikacije koje oni koriste, na primer, igrice.

Primer 1: Igrica



Svaki objekat klase <i>Ptica</i> ima:	
1. podatke koji ga opisuju	<ul style="list-style-type: none"> • veličina • pozicija na kojoj se nalazi
2. funkcionalnosti (metode, tj. procedure i funkcije)	<ul style="list-style-type: none"> • hoda • leti • priča • razmišlja • okreće se
Klasom <i>Ptica</i> su opisani svi objekti te klase, tj. svi podaci i funkcionalnosti (metode – procedure i funkcije) koje imaju objekti ove klase.	

Programski kod gradimo tako što pozivamo metode objekta, na primer:

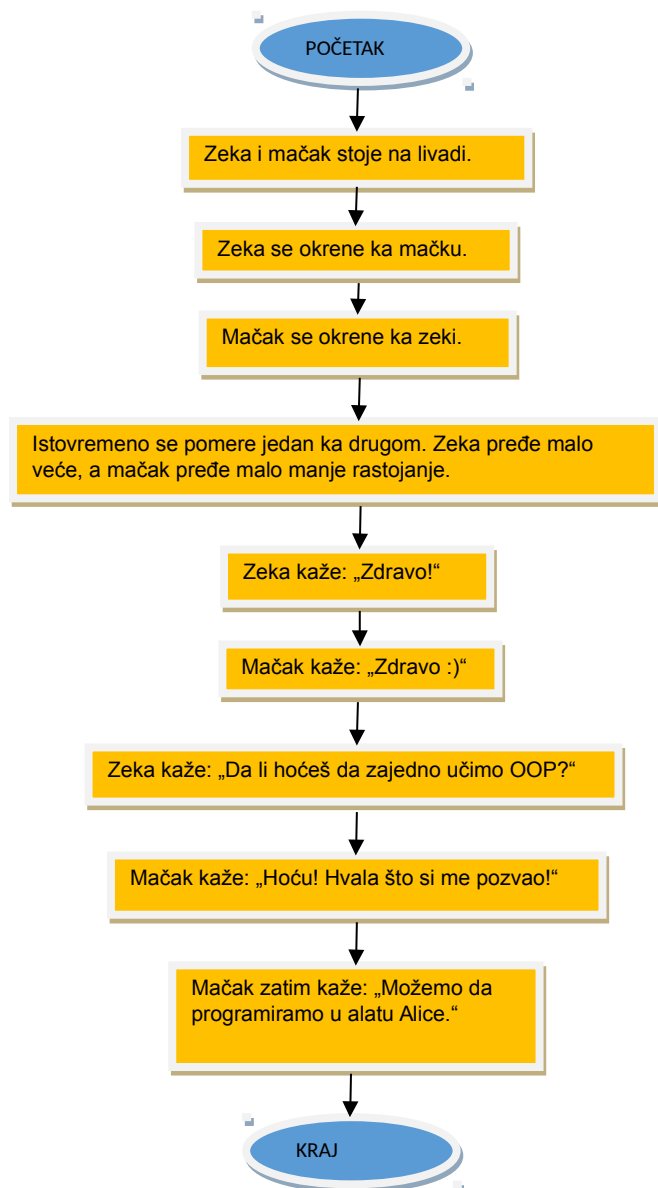
ptica.priča("Kako je lep dan!");

Ovaj programski kod je moguće grupisati u naredbe za kontrolu toka, kao što su grananja i ciklusi, kao i u procedure i funkcije.

Kreiranje animacije u alatu *Alice*

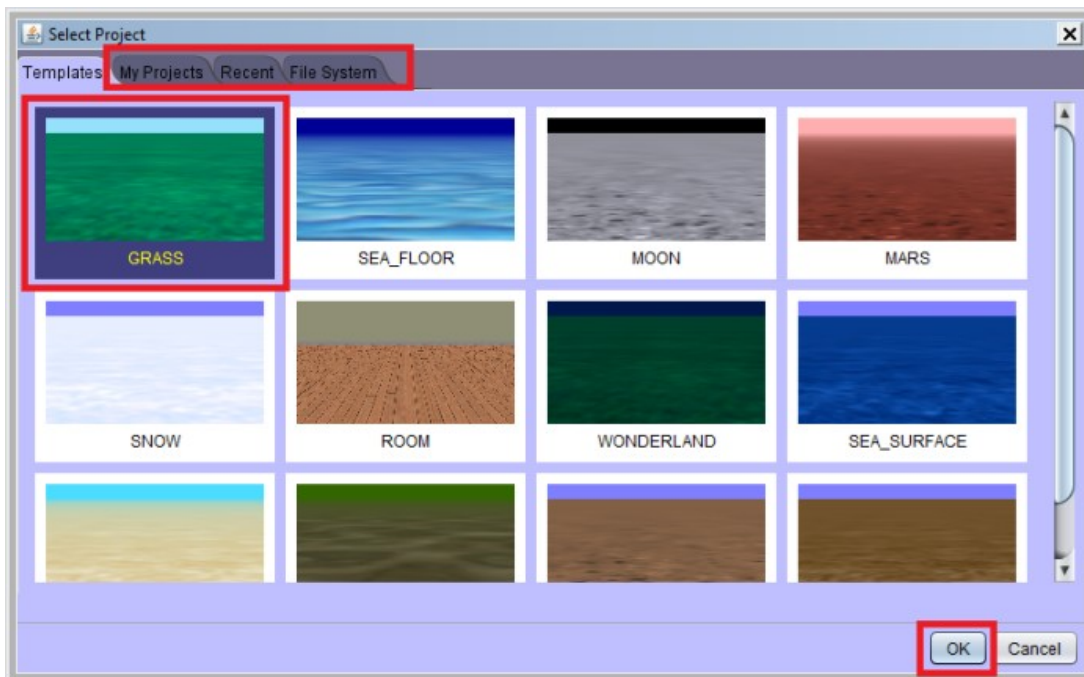
Pre nego što se pristupi kreiranju animacije, potrebno je da se definiše scenario. Scenario može da se opiše rečima ili prikaže dijagramom.

- Zeka i mačak stoje na livadi.
- Zeka se okrene ka mačku.
- Mačak se okrene ka zeki.
- Istovremeno se pomere jedan ka drugom. Zeka pređe malo veće, a mačak pređe malo manje rastojanje.
- Zeka kaže: „Zdravo!“
- Mačak kaže: „Zdravo :)“
- Zeka kaže: „Da li hoćeš da zajedno učimo OOP?“
- Mačak kaže: „Hoću! Hvala što si me pozvao!“
- Mačak zatim kaže: „Možemo da programiramo u alatu Alice.“

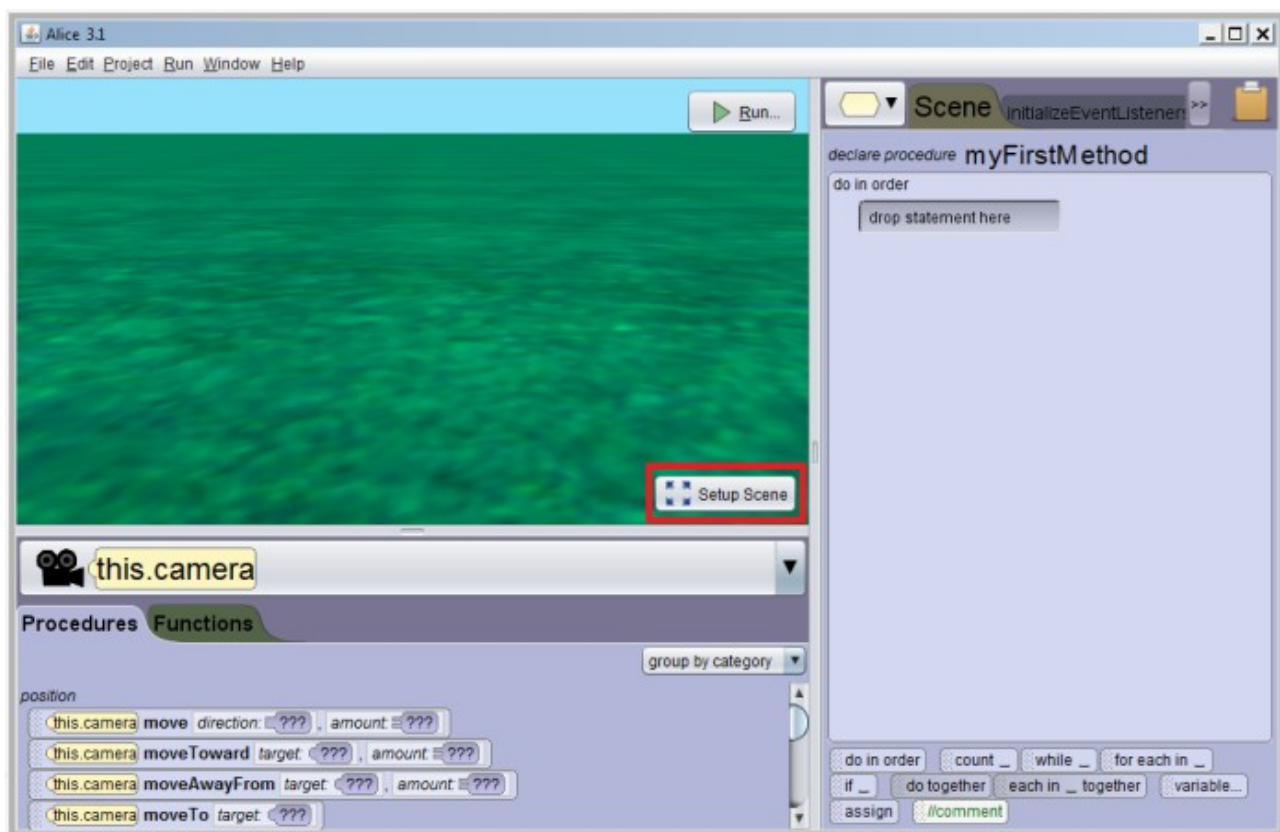


Kada se pokrene alat *Alice*, pojavljuje se dijalog *Select Project*. Pomoću ovog dijaloga je moguće otvoriti već kreirane projekte (neka od opcija My Projects, Recent, File System).

Ukoliko želite da kreirate novi projekat, izaberite prostor (Template) u kojem će se dešavati animacija, a zatim kliknite na dugme *OK*.



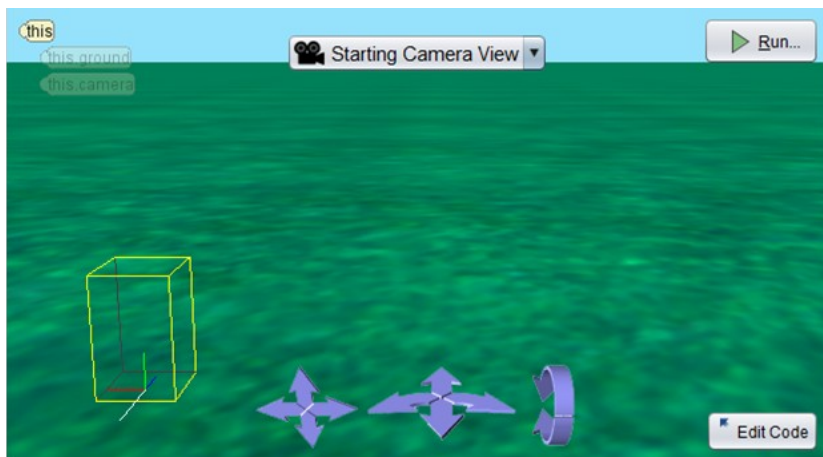
Otvora se novi projekat. Potrebno je da kreirate izgled početne scene. Kliknite na dugme *Setup Scene*.



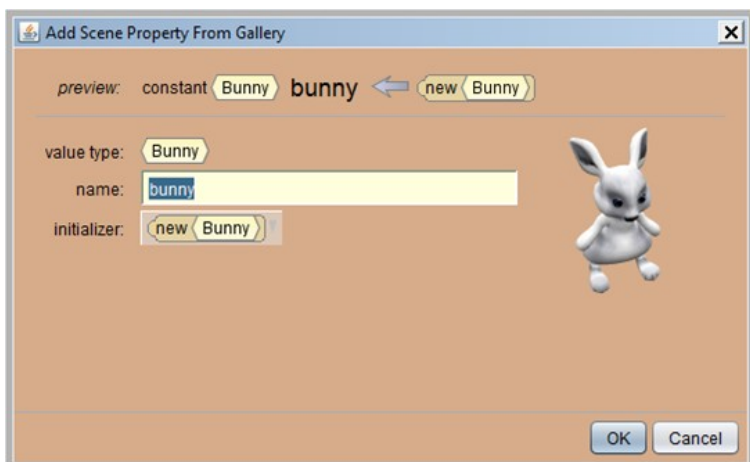
Kada je aktivan prostor za kreiranje scene (izgleda animacije), možete da prevlačite potrebne objekte iz galerije. Galerija sadrži sve klase čije objekte možemo da ubacimo u animaciju. Pronađite zeku i mačka koji se nalaze među dvonošcima.



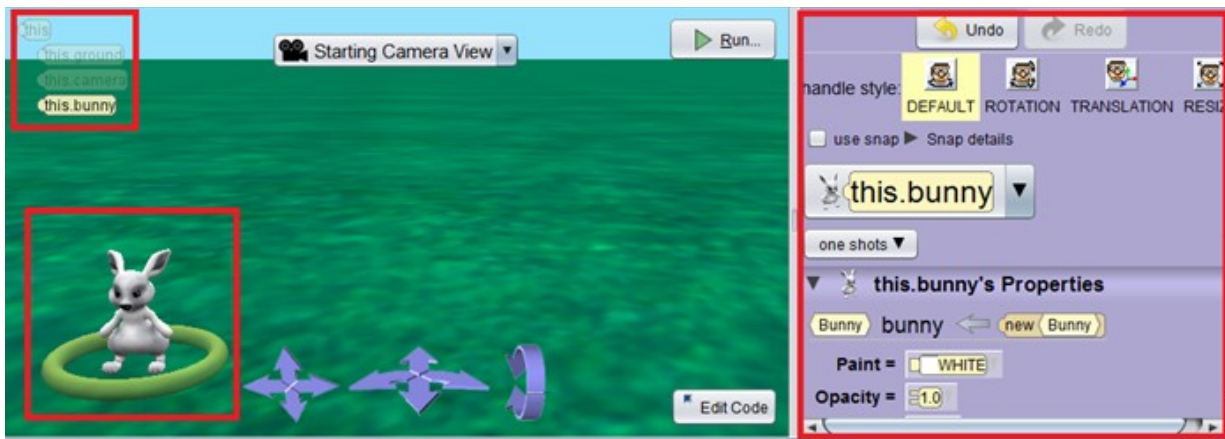
Prvo ubacite zeku tako što ga mišem prevučete na željenu lokaciju.



Kada otpustite taster miša nad željenom lokacijom, pojavljuje se dijalog u kojem možete da izaberete naziv objekta. Podrazumevano se objekat zove kao i klasa, ali sa malim početnim slovom.



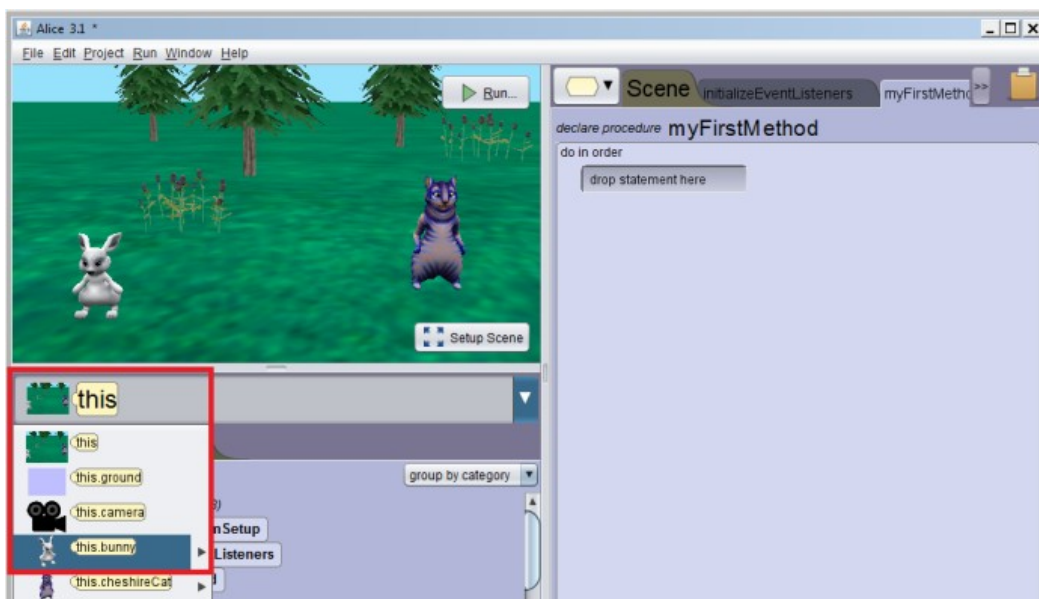
Kada je objekat postavljen, mogu da se menjaju njegova svojstva. U alatu Alice je ovo rešeno isto kao u drugim ozbiljnim razvojnim okruženjima. Potrebno je da objekat bude označen, i onda u prozoru koji se nalazi na desnoj strani mogu da se menjaju njegova svojstva, kao što su na primer veličina i pozicija.



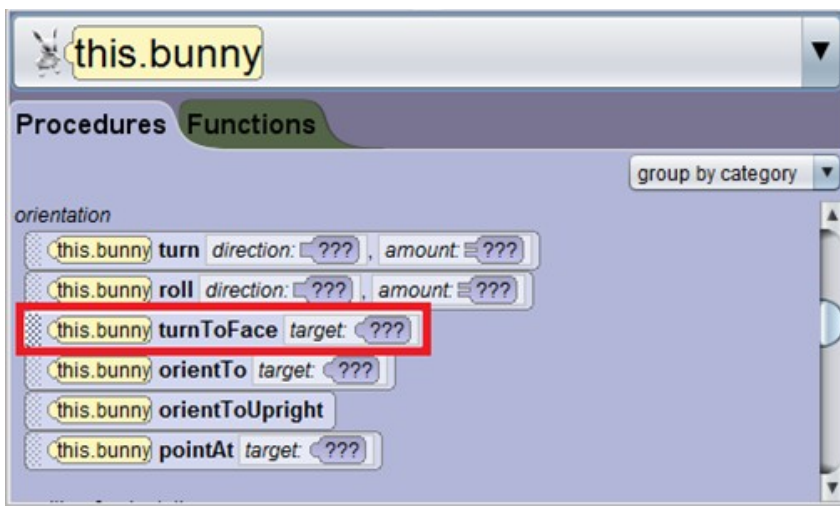
Dodajte i ostale objekte. Neki objekti će biti aktivni tokom animacije. U ovom slučaju će to biti zeka i mačak. Neki objekti mogu da budu samo deo scene. U ovom slučaju to su cveće i drveće. Kada su svi objekti postavljeni, možete da pređete na programiranje tako što ćete da kliknete na dugme *Edit Code*.



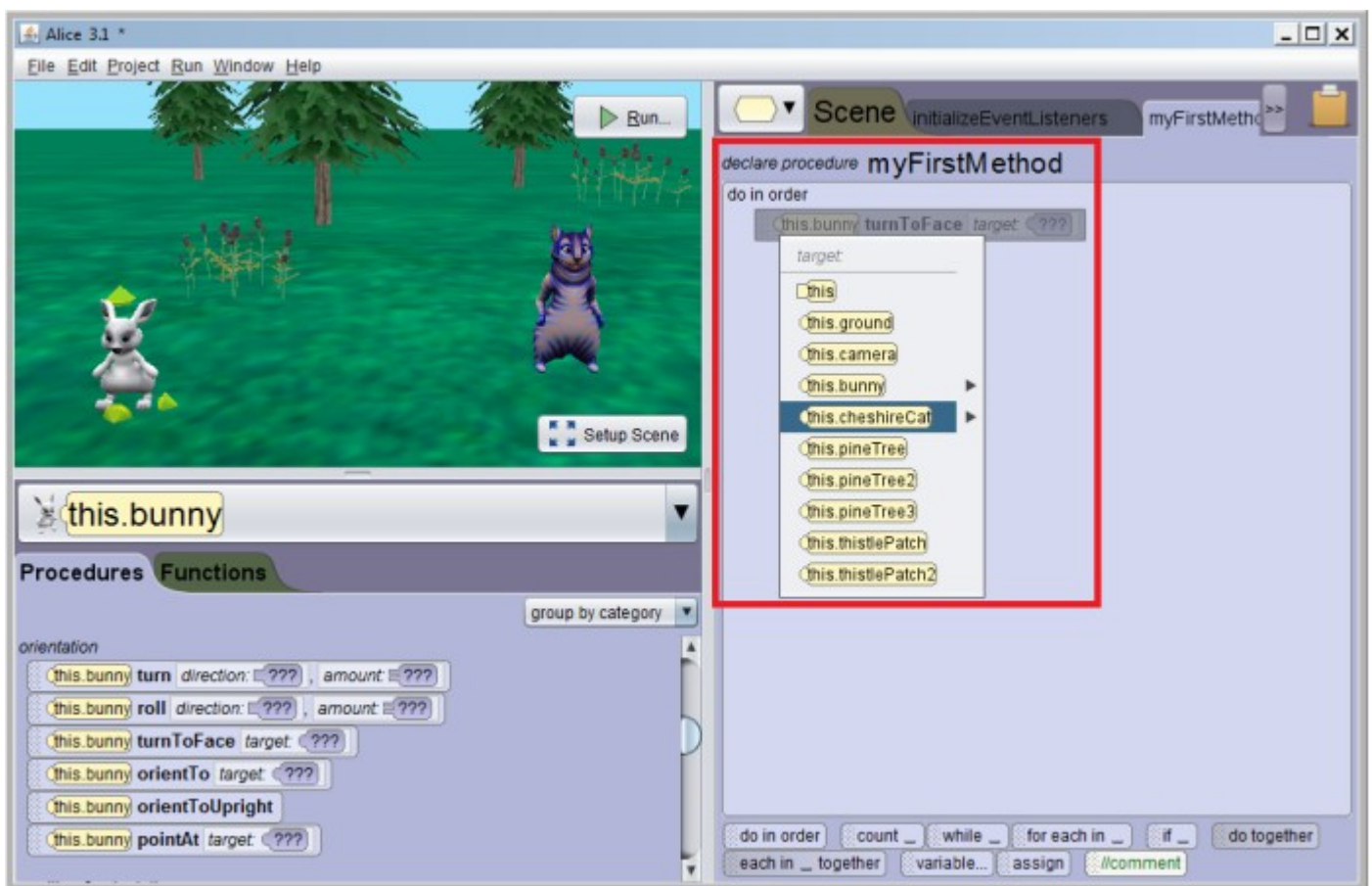
Programirate tako što prevlačite odgovarajuće naredbe u metodu *myFirstMethod*. Ova metoda odgovara metodi *main* u jeziku *Java*. Svaki objekat ima svoje funkcionalnosti. Na primer, zeka može da se okrene ka mačku. Potrebno je da izaberete odgovarajuću proceduru koja opisuje željenu akciju zeke. Prvo izaberite zeku da biste dobili pristup njegovim procedurama i funkcijama.



Izaberite proceduru *turnToFace* i prevucite je u *myFirstMethod*.



Prilikom otpuštanja tastera miša, izaberite vrednost argumenta procedure. Kako zeka treba da se okrene ka mačku, treba da izaberete objekat *cheshireCat* koji se prosleđuje proceduri.

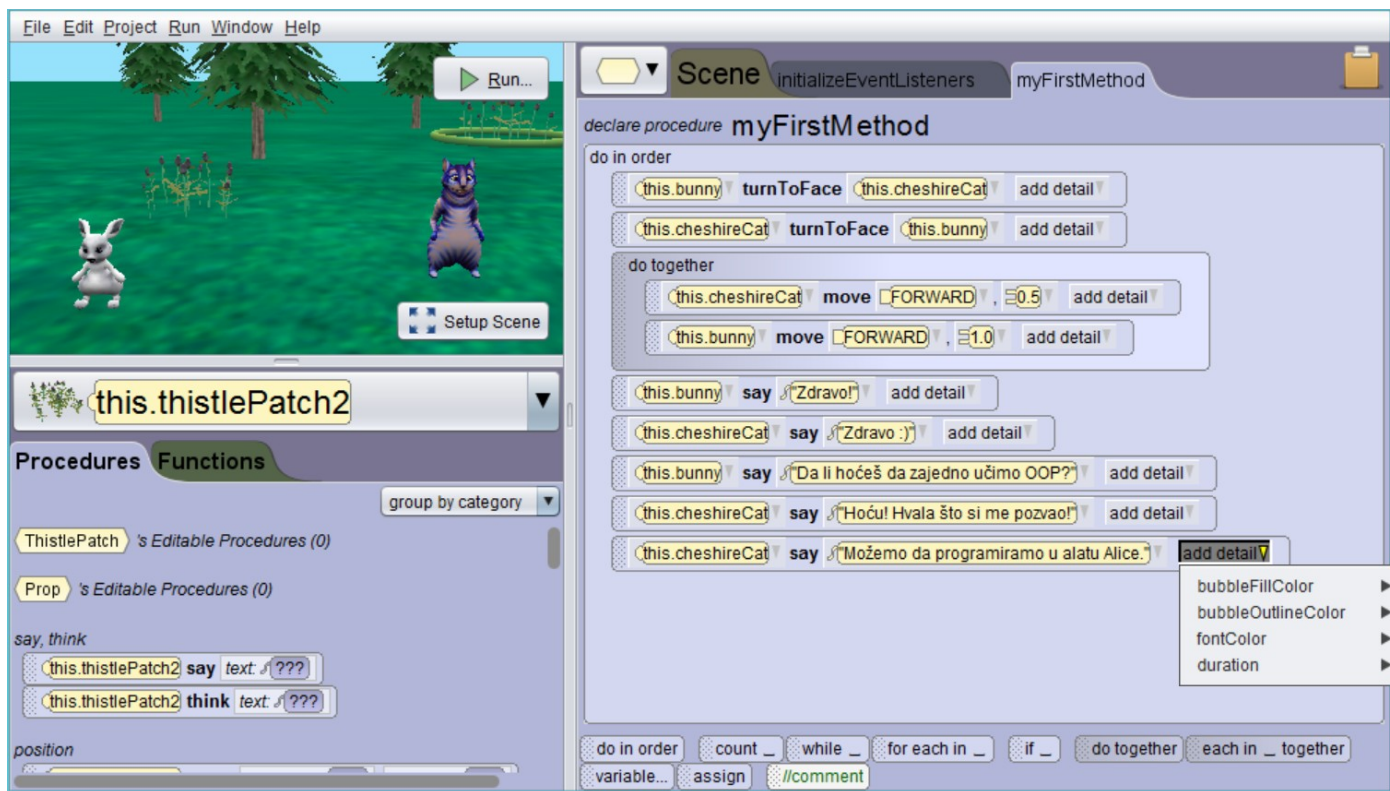


Prva naredba programa će izgledati ovako:



To odgovara načinu poziva metoda u programskom jeziku Java: *objekat.metoda(parametri)*;

Kompletan program možete da vidite na sledećoj slici.



Animaciju pokrenite klikom na dugme *Run*.



Važno je da se svaki program testira. Kada je ovako jednostavna animacija u pitanju, treba proveriti da li u potpunosti ispunjava osmišljeni scenario.



Увод у алгоритме

Милена Марић

milena.maric.f@gmail.com

Девета гимназије „Михаило Петровић - Алас“ – Нови Београд

Апстракт

Измене наставног плана и програма предмета рачунарство и информатика за гимназије донела је поглавље увод у алгоритме у први разред. Улога овог поглавља је да ученике постепено уведе у садржај овог предмета у трећем разреду, где се током целе школске године изучава програмирање. Циљ овог излагања је да представи један од могућих начина како реализовати часове применом неког од окружења за блоковско програмирање. Посебан акценат у оквиру овог излагања биће и на анализи псеудокода као наставне методе којом се подстиче развој алгоритамског размишљања.

Зашто је битан алгоритамски начин размишљања?

Изучавање вештине програмирања не смемо везивати искључиво за оне ученике који ће се касније у животу професионално бавити програмирањем. Програмирање је математичка дисциплина која у потпуности код ученика развија доследност, дисциплину, логичко и апстрактно мишљење. Са тим у вези, основне елементе програмирања требало би да усвоје сви ученици без обзира на њихово будуће професионално опредељење.

Појам и примери алгоритама

У циљу бољег разумевања појма алгоритма било би добро ученике упознати са алгоритмима неких порступака из реалног живота, као и неколико поступака из математике.

Подизање новца са банкомата

1. Унеси картицу у читач како је приказано на слици.
2. Унеси PIN кођ.
3. Ако банкомат пријави да је PIN кођ погрешно унет, иди на корак 2.
4. Ако банкомат нуди могућност избора језика, онда одабери један од понуђених језика.
5. Одабери опцију подизања новца.
6. Ако имаш више рачуна, одабери онај са којег желиш да подижеш новац.
7. Одабери један од понуђених износа (200, 500, 1000, 2000, 5000, 10000, 20000, 50000 динара) или опцију за слободан унос износа.
8. Ако си одабрао опцију за слободан унос, онда унеси износ који желиш да подигнеш.
9. Унеси да ли желиш потврду о трансакцији.
10. Сачекај обраду трансакције
11. Узми картицу
12. Узми новац
13. Ако си одабрао да желиш потврду о трансакцији, онда узми потврду о трансакцији.

Отварање веб-мејл налога

1. У прегледач веба унеси адресу веб-мејл сервиса.

2. Уноси жељено корисничко име све док не унесеш име које није заузето.
3. Уноси своју лозику све док добијаш поруку да је лозинка преслаба.
4. Уноси лозику у поље за потврду лозинке све док се она не поклопи са првобитно унетом лозинком.
5. Унеси своје остале податке (име, презиме, датум рођења, пол и слично).
6. Проучи услове коришћења и ако си сагласан са њима, штриклирај поље којим потврђујеш да се са њима саглашаваш.
7. Ако желиш да примаш промотивне поруке, штриклирај одговарајуће поље.

„Шта је резултат извршавања овог кода?“

Често ученици постављају питање зашто се учи неки програмски језик а не неки други. Да ли је избор језика који се учи на основном курсу програмирања пресудан? Суштински, након основног курса најбитније је да ученици овладају елементарним алгоритмима, да их разумеју, да умеју да их идентификују у конкретном проблему и примене. За ово није пресудан програмски језик. Ученици на почетном нивоу могу и да не изучавају ни један конкретан програмски језик, а да опет имају добра постигнућа. Важно је усвојити основне концепте и добро их разумети. У том циљу ученицима на почетном ниво можемо понудити и псеудокод као једно од могућих решења.

Вештином програмирања се овладава тако што се у почетку анализирају готова решења. Велики је успех научити ученике да могу да одговоре на питање – „Шта је резултат извршавања овог кода?“ Не треба очекивати од почетника да одмах пишу своје програме, добар почетак је анализа и разумевање већ написаних програма. У ту сврху нам може послужити псеудокод. Пожељно је ученицима давати што више малих примера и тражити анализу кода, односно одговор на питање шта је резултат извршавања неког програма.

Анализирати псеудокод и одговорити на питање шта је резултат извршавања следећег кода:

Сабирање бројева

```
zbir := 0
dok nisi čuo sve brojeve, radi sledeće:
    ako si čuo broj x, onda neka je
        zbir := zbir + x
saopšti zbir
```

Степеновање

```
s := 1,
i := 0
dok je i < n radi sledeće:
    s := s * x
uvećaj i za 1
ispiši s
```

Еуклидов алгоритам за НЗД

```
učitaj a i b
dok je b > 0 radi sledeće:
    tmp := b
```

```
    b := a mod b
    a := tmp
ispiši a
```

Дељење бројева

```
r := a;
q := 0;
dok je r >= b radi sledeće:
    r := r - b;
q := q + 1;
```

Множење бројева

```
p := 0; i := 0;
dok je i < a radi sledeće:
    p := p + b;
i := i + 1;
```

Максимум пет елемената

```
broj := unesi_broj
maksimum := broj
i := 1
dok je i < 5 radi sledeće
    broj := unesi_broj
ako je broj > maksimum onda
    maksimum := broj i := i + 1
saopšti maksimum
```

Када су ученици савладали овај део, тј. када умеју да анализирају већ готове примере алгоритама, можемо да их упознамо са неким од окружења, као што је на пример Scratch и да пред њих ставимо задатак самосталног креирања програма. Следећи сегмент није осмишљен као туторијал окружења Scratch, већ као методичка јединица која даје оправдање зашто би поједине примере требало показати ученицима.

Блоковско програмирање

Једно од окружења која можемо користити у првом разреду гимназије је Scratch. Настао је на престижном Масачусетском технолошком институту (MIT) у Бостону и бесплатно је доступан свима. Његове верзије постоје за све водеће оперативне системе (Microsoft Windows, Linux, Mac OS X), а веома популаран начин коришћења система Scratch данас је из прегледача веба (систему се може приступити на адреси <http://scratch.mit.edu>, одакле се може или користити преко веба, или преузети и инсталирати на локални рачунар).

Scratch је окружење које се може користити за различите намене, као што је креирање анимација, филмова, презентација, у почетним фазама учења програмирања. Scratch је једноставан за коришћење,

ученици самостално могу да истражују окружење и брзо га савладају, а са друге стране, могуће га је користити за имплементацију стандардних алгоритама програмирања на почетном нивоу. Иако ово шарено окружење делује прикладније у настави програмирања у основној школи, у примерима који следе можемо видети да га сасвим оправдано можемо користити и у гимназији у првом разреду.

Пример 1:

Конверзија

Превођење центиметара у инче

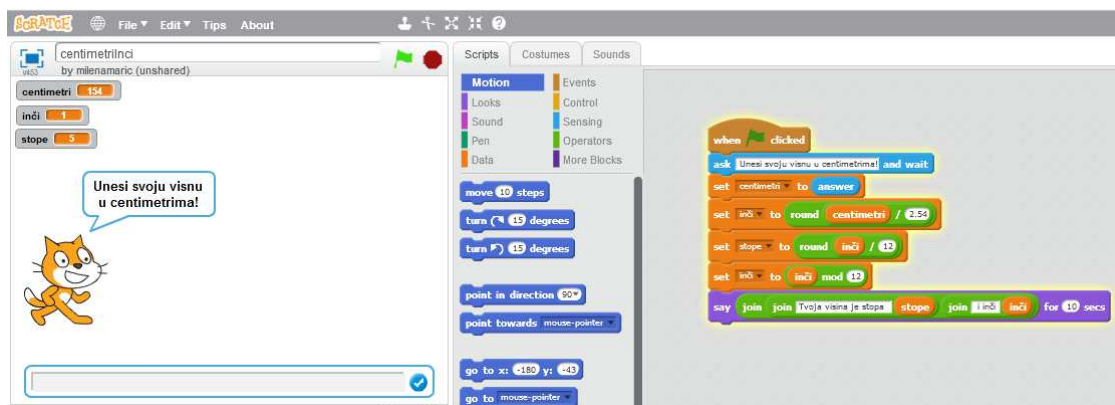
Написати програм који висину дату у центиметрима преводи у стопе и инче.

Анализа задатка:

Ово је пример елементарног задатка у оквиру кога се примењује учитавање података, затим обрада података применом неке математичке формуле и на крају се израчунати резултат исписује.

Даље, радећи овај задатак ученици ће се упознати са начином чувања података у програму, тј. употребом променљивих. У овом примеру постојаће три помоћне бројевне променљиве (променљива за центиметре, стопе и инче).

Што се тиче упознавања са окружењем, ученици ће радећи овај пример увидети да се сваки програм започиње кликом на зелену заставицу, што се у делу за писање скрипта означава nanoшењем блока на коме пише „Када се кликне на (слика зелене заставице)“, који се налази у групи *Догађаји*. Такође, ученици ће усвојити да се подаци чувају у променљивама, а да се променљиве у окружењу Scratch дефинишу у групи *Подаци*, као и да је за комуникацију са корисником потребно користити опције групе *Осећаји*. За опис коришћених математичких формула користе се опције групе *Оператори*.



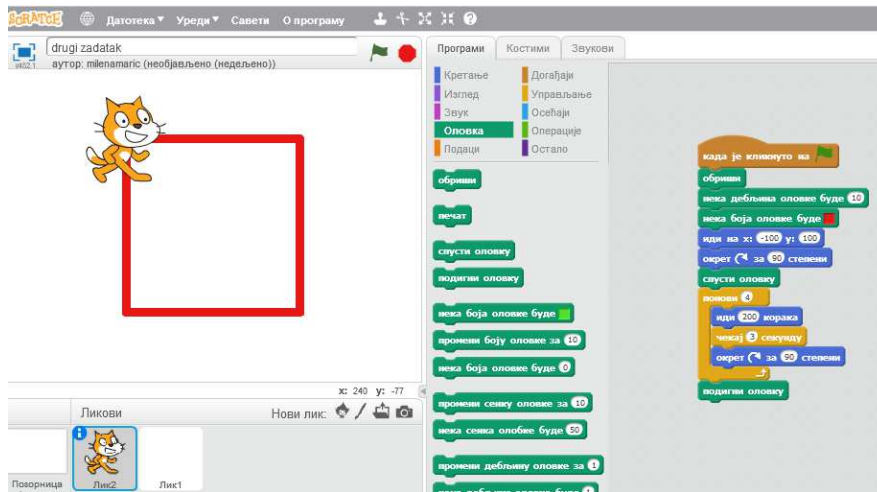
Пример 2:

Нацртај квадрат

Написати програм тако да мачак црта квадрат.

Анализа задатка:

Ово је пример задатка где се ученици упознају како да користе координатни систем. Такође, упознају се како се ликови могу кретати коришћењем опција из групе *Креатње*. Такође, Scratch даје могућност да лик оставља траг приликом кретања, а подршка за цртање се налази у групи *Оловка*.



Пример 3:

Летећи мачак

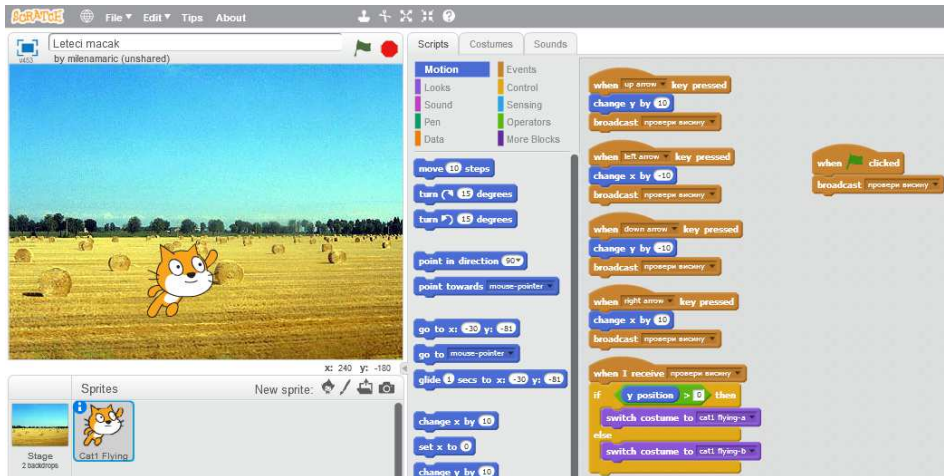
(Гранање)

Написати програм у коме ће бити могуће померати мачка по позорници коришћењем стрелица на тастатури. Притом, позадина треба да буде слика из природе на којој се јасно види хоризонт и мачак треба да мења положај тела у зависности од тога да ли је испод или изнад хоризонта (изнад хоризонта треба да заузме положај у коме лети, а испод треба да заузме положај у коме се спрема да полети). Такође, мачак мења и изглед костима у зависности да ли се налази изнад или испод хоризонта.

Анализа задатка:

Ово је пример задатка где се користи гранање. У зависности од висине мачак ће променити свој костим. Висину на којој се мачак налази треба проверити након сваке промене положаја мачка. Да не бисмо понављали код, употребићемо технику разглашавања догађаја. Сваки пут када се мачак помери разгласићемо нови догађај провери висину.

Такође, у оквиру овог задатка ученици се упознају како да контролишу лик преко стрелице на горе, доле, лево, десно. Подршка за ово налази се у групи *Догађаји*. Промену позадине могуће је подесити у групи *Позадине*. У оквиру овог задатка ученици се сусрећу и са костимима. Костими су заправо слике које најчешће представљају различите положаје главног лика. Постоји велики број ових слика у библиотеци, но корисник увек може и сам да креира нови кости.



Пример 4:

Статистика јабука

Напишимо програм који насумично одређује тежине три јабуке, подешава величину јабука на екрану тако да пропорционално одговара њиховој тежини, а затим израчунава укупну тежину јабука, просечну тежину и најмању и највећу тежину.

Анализа задатка:

Циљ овог задатка је да ученици науче да израчунају суму три броја, њихову аритметичку средину, као и најмањи, односно највећи број од дата три.



Пример 5:

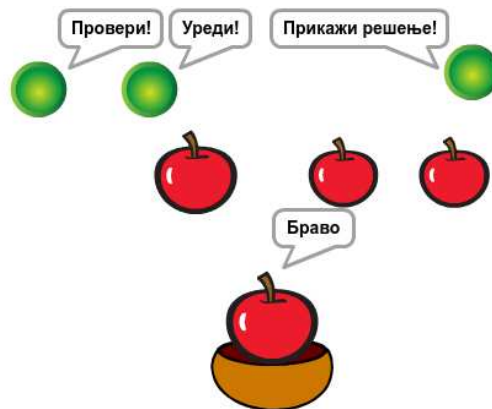
Постави највећу јабуку у чинију!

Написати сада програм у коме се кориснику приказују три јабуке случајно одабране величине и од њега се тражи да највећу од њих убаца у корпу. Постоји дугме којим се проверава да ли је задатак коректно урађен, постоји дугме којим се открива тачно решење и постоји дугме којим се јабуке уређују по величини.

Анализа задатка:

Циљ овог задатка је да ученик одреди највећи број од дата три (тежину највеће јабуке), као и да запамти која од дата три јабуке је била најтежа (уз променљиву у којој се памти вредност максимума, уводимо и променљиву у којој се памти позиција максимума и ажурирамо је када год ажурирамо вредност).

У овом задатку је такође битан детаљ дугме које ређа јабуке по величини, што суштински од ученика захтева примену једне варијанте алгоритма тзв. сортирања селекцијом који прво доводи најмањи елемент на првом месту, затим други по величини на друго и тако даље, све док се сви елементи не уреде. Ово ћемо реализовати тако што ћемо први елемент упоредити са другим и ако је други мањи, разменићемо им места. Након тога, исти поступак ћемо применити на први и трећи елемент. Након та два поређења и након евентуалних размена, на првом месту ће се налазити најмања вредност. Након тога, исти поступак ћемо применити на други и трећи елемент, чиме ћемо постићи да је други елемент мањи од трећег, чиме су сва три елемента уређена неоппадајуће. Такође, овде примењујемо и алгоритам размене две вредности уводећи помоћну променљиву.



Пример 5:

Распореди јабуке у што мање чинија!

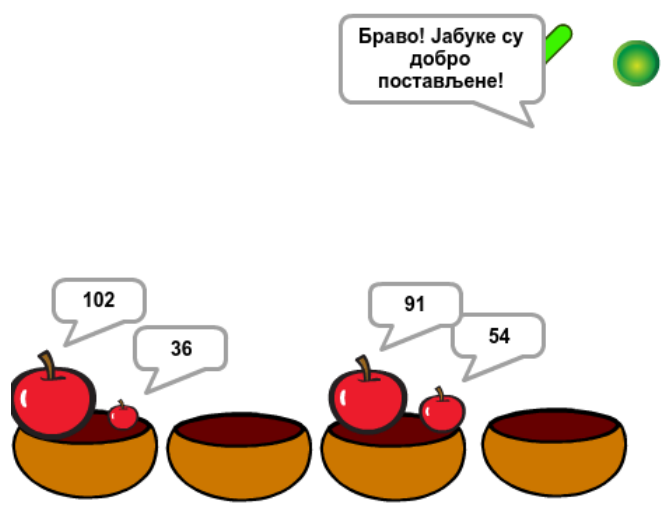
Написати програм у коме су дате 4 јабуке насумично одабраних тежина (између 20 и 120 грама) и четири чиније чија је максимална носивост 150 грама. Задатак корисника је да распореди јабуке у што је мањи број чинија могуће, тако да ни у једној чинији тежина не буде већа од дозвољене.

Игра нуди дугме за проверу коректности решења, али и дугме помоћ помоћу којег се кориснику приказује једно оптимално решење.

Анализа задатка:

За успешно решавање овог задатка потребно је поређати јабуке од најтеже до најлакше, дакле примењујемо алгоритам сортирања. Јабуке не морају бити визуелно распоређене по тежини, али је потребно да ученик интерно запамти која по реду јабука је најтежа, односно која јабука је друга по тежини и тако редом.

Централни део задатка је проналажење оптималног распореда јабука. Опишимо сада централни алгоритам који ће оптимално распоредити јабуке и попунити листу чиније. Уз то, увешћемо и променљиву која ће памтити најмањи број чинија потребан да се све јабуке распореде. Једноставности ради, тежине јабука ћемо на почетку представити помоћу четири променљиве. Ако је збир ове четири вредности мањи или једнак носивости чиније (није већи од њен), тада се све јабуке могу распоредити у једну чинију. Ако се јабуке могу распоредити у две чиније тада се или три најлакше јабуке могу спаковати заједно, а најтежа јабука се пакује сама или се могу спаковати најлакша и најтежа јабука заједно у једну чинију, а средње две јабуке у другу. Заиста, ако би се, на пример, могле спаковати најтеже две у једну чинију и најлакше две у другу, тада се могу разменити друга по величини и најлакша и чиније и даље бити коректно попуњене (чинија у којој је најтежа јабука је само још олакшана, док је чинија у којој су сада друга и трећа јабука коректне тежине јер је она лакша од чиније у којој су биле прва и друга, а то је по претпоставци била коректна чинија). Ако паковање у две чиније није могуће, а јесте у три чиније, онда је могуће спаковати најлакше две јабуке заједно. На крају, ако је збир најлакше две јабуке већи од капацитета чиније, онда је неопходно употребити све четири чиније (по бројевим датим на почетку задатка, гарантовано је да свака јабука може стати сама у сваку чинију).



Новине у програмима информатике за средње стручне школе

Знања, вештине и вредносни ставови са становишта развоја појединца, кључни су за развијање иновативности, продуктивности, а тиме и конкурентности сваког друштва. Динамика промена које доносе нове технологије захтевају компетенције за прихватање и прилагођавање променама.

Информационо-комуникационе технологије треба да буду заједнички именитељ за све предмете. Рачунар треба ученику да буде на корист. Наставници који примењују ИКТ кроз свој предмет су „агенти” промене.

Очекивани исходи зависе од подручја рада и образовног профила. Мора се изнаћи прикладан начин за праћење индивидуалног напретка ученика и усмеравање њиховог даљег развоја.

Пређашње стање:

У Просветном гласнику број 11/13 од 28.8. 2013. објављен је наставни програм за рачунарство и информатику. По том програму рада циљ наставног предмета рачунарство и информатика у средњој стручној школи је стицање знања, овладавање вештинама и формирање вредносних ставова који доприносе развоју информатичке писмености неопходне за живот у савременом друштву, даље стручно усавршавање и практичну примену у процесу рада; као и оспособљавање ученика да ефикасно и рационално користе рачунаре на начин који не угрожава њихово физичко и ментално здравље.

Садашње стање:

У Просветном гласнику број 11/16 од 26.8. 2016. објављен је нови наставни програм за рачунарство и информатику.

Циљеви предмета:

1. Упознавање са значајем информатичке компетентности и дигиталне писмености за живот у савременом друштву.
2. Упознавање ученика са основним постулатима информатике.
3. Оспособљавање ученика за коришћење основних могућности оперативних система и система датотека.
4. Упознавање основних принципа функционисања локалних мрежа и оспособљавање за коришћење мрежних ресурса.
5. Упознавање основних принципа функционисања интернета и оспособљавање за коришћење његових најпопуларнијих сервиса.

6. Оспособљавање ученика да користе различите изворе информација и имају критички однос према њима, да баратају информацијама, да одвајају битно од небитног, да процењују исправност извора и употребе их на ефикасан начин.
7. Оспособљавање ученика за тумачење и разликовање података и информација кроз табеларно, графичко, текстуално приказивање, проналажење примене, повезивање са претходним знањем из других предмета.
8. Оспособљавање ученика да одаберу и примене најприкладнију технологију сходно задатку, области у којој је примењују или проблему који решавају.
9. Оспособљавање за одговорну примену препорука заштите здравља, безбедности, личне и приватности других у свакодневном раду у дигиталном окружењу.
10. Развијање свести о важности етичког коришћења информација и технологије.
11. Оспособљавање за коришћење рачунара у другим предметним областима.
12. Развијање интересовања за стицање потребних знања и развој вештина које би ученици употребили за даље образовање и напредовање.

Пређашње стање:

Наставне теме:

1. Основе рачунарске технике (6-8)
2. Основе рада у оперативном систему са графичким интерфејсом (8-10)
3. Текст-процесор (14-16)
4. Слајд-презентације (10-12)
5. Рад са табелама (14-16)
6. Интернет и електронска комуникација (12-14)

Садашње стање:

Наставне теме:

1. Основе рачунарске технике (10)
2. Основе рада у рачунарском систему (8)
3. Примена ИКТ-а (текст процесор, израда презентација, табеларне калкулације) (40)
4. Рачунарске мреже, интернет и електронска комуникација (16)

Наставни програм је иновиран иако има исте теме/модуле. Дат је исходно. По завршетку тема ученик је у стању да: објасни, наведе, разликује, претвара, препозна,наброји,изврши, организује, одабере, креира, користи, припреми, изведе, приступа, пријави, води рачуна....

Најобимнија тема је ПРИМЕНА ИКТ. Ученике треба упознати са постојањем текст процесора који омогућавају дељење докумената и заједнички рад на њима преко интернета (рад у облацима). Ученике треба упознати са логичком структуром типичних докумената (молби, обавештења, итд.), школских реферата, семинарских и матурских радова. Практичну реализацију модула израда презентација, извести у неком од програма за израду презентација (слајд, WEB или презентације у „облацима“) · Конкретне примере за вежбање прилагодити образовном профилу кроз корелацију са стручним предметима. Резултат рада сваког модула у оквиру теме Примена ИКТ треба да буде један комплетан пројекат везан за образовни профил и примену међупредметних компетенција.

Веома значајна тема посвећена интернету код ученика развија навике коришћења интернета за прикупљање информација за потребе наставе информатике, али и осталих предмета. На крају ове теме ученик треба да процени квалитет информација са којима се сусреће -препозна прихватљиво / неприхватљиво понашање у оквиру дигиталне комуникације, пријави непримерене дигиталне садржаје или нежељене контакте и потражи помоћ, препозна знаке зависности од технологије и правовремено реагује као и да води рачуна о онлајн идентитету и приватности.

Како унапредити наставу и препоруке за реализацију?

При реализацији програма дати предност пројектној, проблемској и активној настави, кооперативном учењу, изградњи знања и развоју критичког мишљења. Подстицати тимски рад и сарадњу нарочито у областима где наставник процени да су присутне велике разлике у предзнању код појединих ученика.

- Конкретне примере за вежбање прилагодити образовном профилу.
- При реализацији модула инсистирати на вештинама.
- У процесу оцењивања користити електронске тестове и тестове практичних вештина.
- У току реализације наставног програма инсистирати на правопису и употреби одговарајућег језика тастатуре.
- Подстицати креативност код ученика.
- Код упознавања са развојем информационих технологија не упуштати се у перформансе рачунара појединих генерација већ сагледати место информационих технологија у развоју науке и технике у датом историјском периоду и утицај технологије на развој писмености, привреде и људског друштва уопште (реализовати кроз дискусију).
- Објаснити ученицима значај и направити хијерархију утицаја појединих компоненти рачунарског система на перформансе рачунара.

- Ради постизања важног педагошког циља: развоја код ученика навике за самостално коришћење помоћне литературе, ученици могу да, коришћењем рачунарских часописа, Интернета или обиласком продавница, опишу конфигурацију рачунарског система која у том моменту има најбоље перформансе.
- Ученицима развити правну и етичку свест о ауторским правима над информацијама које се дистрибуирају путем мреже. Посебну пажњу посветити потреби коришћења лиценцираних програма, заштити програма и података, вирусима и заштити од њих.
- Повезивати градиво са искуствима ученика.
- Вршњачко учење примењивати што чешће у пракси.
- Кроз тему Примена ИКТ-а ученик треба да ради на једном пројекту у оквиру сваког модула чија је тематика везана за конкретан образовни профил и да на пројекту примени сва усвојена знања и вештине. Потенцирати на корелацији предмета (рачунарство и информатика, матерњи језик, стручни предмети...). Примена табеларних калкулација је модул за примену алгоритамских шема код линијских и раздранатих структура. Занимљиво би било дати ученицима да направе и обраде неку анкету сами. За израду презентација инсистирати на појашњењу принципа презентовања, важности идеје, следа, композиције, увежбаности, ...

Примери добре праксе:

- Посетити сервис рачунара где би ученици имали могућност да виде склапање/расклапање и сервисирање рачунара.
- Посетити провајдерке куће и показати ученицима како раде велике мреже.
- Поставити час као: покажи ми како умеш да користиш рачунар, са пуно вршњачког учења.
- За проверу знања и вештина користити електронске тестове, квизове и проверу практичних вештина.
- Рад у текст-процесору може бити израда дужег новинског чланка – репортаже.
- Код табеларних калкулација може се радити спецификација трошкова за организовање екскурзије у трајању од неколико дана.
- За израду презентације користити сараднички рад на дељеној презентацији „у облаку”.